Ministry of Education and Science of Ukraine
National Technical University of Ukraine
"Kyiv Polytechnic Institute"

National Academy of Sciences of Ukraine
Institute for Information Registration Problems

A.A. Snarskii, D.V. Lande

COMPLEX NETWORKS MODELING

Tutorial

Kyiv – 2015

UDC 004.5
BBC 22.18, 32.81, 60.54
C 95

Recommended for printing by the Academic Council of the Educational and Scientific Complex "Institute of Applied System Analysis" NTUU "KPI" (protocol No. 8 dated September 2014.), Scientific Council of the Institute for Information Registration Problems of the National Academy of Sciences of Ukraine (Protocol No. 2, dated January 2015.)

Reviewers :
Sci. Dr., Prof. A.G. Dodonov;
corresponding member NAS of Ukraine, Sci. Dr., Prof. B.I. Lev

C 95 **Snarskii A. A., Lande D. V. Complex Networks Modeling: A Tutorial**. – Kyiv: Engineering, 2015. – 206 p. (Translation into English using Google Translate)

The textbook discusses the basic issues of the theory of complex networks: characteristics, algorithms, models, search problems, ranking, and also provides the information necessary for mathematical and computer modeling and analysis of complex networks.

The theory of complex networks is a complex scientific direction, located at the intersection of such sciences as discrete mathematics, graph theory, algorithm theory, nonlinear dynamics, the theory of phase transitions, percolation, etc. Therefore, for successful modeling of complex networks, basic information from all these areas is needed, which are presented in this tutorial.

The publication is intended for students and graduate students of higher educational institutions, engineers and researchers working in the fields of system analysis and applied mathematics.

UDC 004.5
BBC 22.18, 32.81, 60.54
ISBN 978-966-2344-44-8
Circulation 300 copies.

Table of contents

Introduction

Now one can hardly argue against the fact that a new paradigm has been born, or in a more official language, a new scientific direction – the theory of complex networks. Let's look back at 30 years ago. Then, everyone who read, flipped through or at least looked at the pictures and captions to them in the book of B. Mandelbrot "The Fractal Geometry of Nature" discovered that everywhere you look – only fractals. And trees, and bushes, and profiles of mountains and clouds, and the trajectory of a Brownian particle familiar from school, etc. and so on. The term fractal has become fashionable, entered (sometimes "out of business") into the terminology of articles and books on various fields of science – physics, chemistry, biology, economics, medicine,... And this despite the fact that the first fractal objects were studied already in the XIX century (Peano curve, Cantor set, etc.). Now interest in fractals has cooled down, the frequency of using the term fractal in books is falling. A new area that is actively developing is the so-called "complex networks". Now, everywhere you look, you see networks – social networks, networks of friendship, co-authorship in scientific publications, sexual relations, business connections, publications in the media, sharing words in texts, metabolisms (metabolic processes), blood vessels, transport, finally, (see figure) Internet and WWW. Huge field of activity, many results, new sections in scientific journals and new journals.

Benoit Mandelbrot (1924 – 2010)

Mandelbrot b. Fractal Geometry of Nature = The Fractal Geometry of Nature. – M.: Institute of Computer Science. research, 2002. – S. 656.

"Where do you work?" "I am a null physicist." An astonished look. "Listen, tell me, please, what is null physics? I can't understand at all." – "Me too". *A. and B. Strugatsky "Distant Rainbow".*

At the first acquaintance with books and reviews on the theory of complex networks, a question arises. Isn't this theory of complex networks just a fancy name? The economic

component of the fashionable term is well known, he wrote "nanofilter" and received a mega-grant. What is the difference between a network, albeit a complex one, from a graph? And the theory of complex networks is derived from the theory of graphs, a theory that was initiated by Euler with his famous problem of the Königsberg bridges.

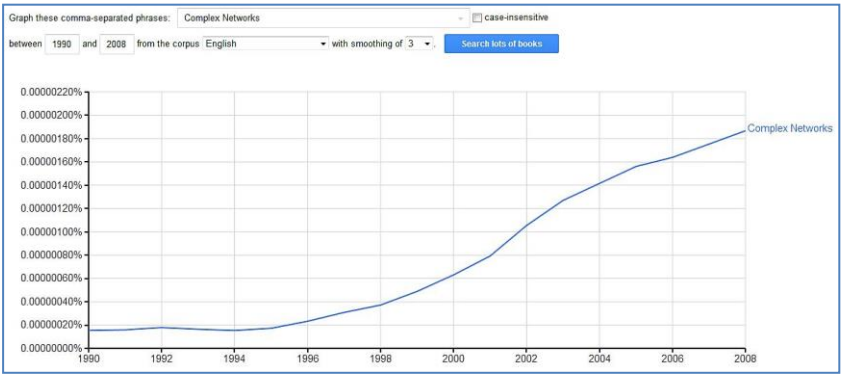


Figure 1. The dynamics of the use of the phrase Complex Networks, obtained using the Google service Ngram Viewer (https://books.google.com/ngrams)

Formally, any network is a graph. And, again, formally, the theory that studies the properties of complex networks should be called graph theory. We give the following counterexample. Classical mechanics studies, among other things, the motion of material points. An ideal gas is just moving material points. However, the properties of gas are studied by another branch of physics – statistical mechanics.

She has her own methods, terms, techniques. And "ordinary" classical mechanics cannot cope with the task of describing gas, there are too many particles even in a small "piece" of gas. And Maxwell, Boltzmann, Gibbs and many others had to create a new science and introduce new concepts, for example, temperature and entropy, which are not needed and are not introduced in principle in classical mechanics.

The relationship between graph theories and complex networks is similar. As the basis of classical statistical physics is classical mechanics, so the basis of the theory of complex networks is the theory of graphs. At the same time, graph theory can receive meaningful statements, as a rule, for graphs (networks) of small size or special structure. Whereas the theory of complex networks deals with a large number of, as a rule, randomly connected nodes. In this regard, on the one hand, many common questions of graph theory are not of interest to the theory of complex networks. For example, such a question – is this graph planar? For a random graph with a large number of nodes and connections, the answer is obvious (and of no interest) – no. On the other hand, many very important concepts in the theory of complex networks for a graph with a small number of nodes and connections are either of no interest, or they are simply difficult to formulate meaningfully. For example, such an important characteristic in the theory of complex networks as the distribution function according to the degree of nodes can be calculated for any, incl. and a small number of nodes. However, due to its probabilistic content, this concept is useful only in the case of a large number of nodes and links.

Dorogovtsev S.N., Mendes JFF Evolution of Networks: From Biological Networks to the Internet and WWW. – Oxford, USA: Oxford UniversityPress, 2003. – P. 280.

Mark Newman, Albert-Laszlo Barabasi, Duncan J. Watts. The Structure and Dynamics of Networks: (Princeton Studies in Complexity). – Princeton, USA: Princeton UniversityPress, 2006. – P. 624.

What is the theory of complex networks? Let's list some problems and tasks.

First, by studying the standard characteristics of graphs for complex networks of various nature – random graphs, scale-free networks, small-world networks, etc.

Secondly, the definition and study of new characteristics of complex networks, such as, for example, the average minimum path, mediation, clustering coefficient.

Thirdly, the study of various "physical" processes on complex networks – diffusion, epidemic processes, various flows (information, electric current, etc.) After all, the famous PageRank algorithm considers wandering through links (hyperlinks) in the complex web of the WWW.

Fourthly, there is a direction that is very important from an applied point of view – methods of restoring, protecting and destroying networks. Such a question – how many nodes (connections) need to be "killed" in order to, for example, collapse a "giant cluster" or to significantly increase the minimum average path? That is, as the system administrators say, so that the "network is down." This is also related to the issues of network optimization.

Fifth, the search for implicit connections, those that are artificially hidden. An important application of this task is the search for terrorist connections. And, of course, business intelligence.

Python-networkx package in Ubuntu (https://launchpad.net/ubuntu/+source/python-networkx)

The methods that are used to solve these and many other problems can be conditionally divided into three types. Graph theory methods are largely combinatorial. Numerical modeling is now well developed, tested and adapted for the purposes of the researcher of complex networks. For example, a special package (python-networkx) has been developed for the Phyton programming language – a toolkit for creating, manipulating and studying complex networks, which makes it possible to numerically find almost all possible characteristics of complex networks. The third type of methods, which made it possible to establish the main patterns in complex networks, are the methods of theoretical physics. From mean field theory to renormalization group and diagram technique. No wonder a significant number of leading researchers of complex networks of physics are theorists.

What is our book about and who is it intended for? It must be said right away that for those who are familiar with the main reviews or books on the theory of complex networks, our book is of no interest. It is intended, first of all, for those who have only

heard the term "complex network" or met with it in an article on biology, economics, etc. It is quite difficult for an interested reader to immediately master a large overview, terminology, methods, and main tasks. This is where this book should help. The first part describes, using simple examples, the main characteristics and properties of several of the most common networks. The second part of the book is intended for those readers who would like, as an example, to see "how" the theory of complex networks works. Here are a few tasks (problems) selected by the taste of the authors.

The material of the book is the basis of a two-semester special course "Modeling of complex networks", which is read to students of the Institute of Applied System Analysis (IPSA) of the National Technical University of Ukraine "KPI". We express our gratitude to the rector of NTUU "KPI", the head of the Department of Mathematical Methods of System Analysis (MMSA), Academician M.Z. Zgurovsky, Dean of the Faculty of System Research of the IPSA, Professor V.D. Romanenko and Associate Professor of the Department of MMSA Yu.A. Timoshenko for the idea and support in creating this special course.

And, finally, we express our sincere gratitude to our colleagues who supported the idea of writing this tutorial and are directly related to many of the results presented here: M.I. Zhenirovsky, I.V. Bezsudnov, A.G. Dodonov and S.M. Braichevsky.

Part I. Complex networks

1. Basic concepts

1.0. The direction of "complex networks"

For a long time, such a riddle has been spread among the inhabitants of Koenigsberg: how to pass through all seven bridges (across the Pregolya River, Fig. 1.0.1) without passing through any of them twice. No one could prove or disprove the possibility of the existence of such a route.

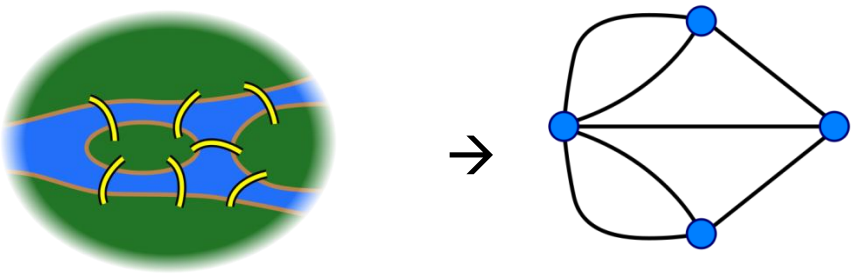


Figure. 1.0.1 – Scheme of the bridges of Koenigsberg

Leonard Euler (1707-1783)

Leonhard Euler 13 March 1736 in a letter to an Italian mathematician and engineer, Marioni wrote that he had found a rule, using which it is easy to determine whether it is possible to pass over all bridges without passing twice over any of them:

- The number of odd vertices must be even (there is no graph that has an odd number of even vertices)
- If all the vertices are even, then you can draw a graph without lifting your pencil from the paper. In this case, you can start at any vertex and end at the same vertex.
- A graph with more than two odd vertices cannot be drawn in one stroke.

The Count of Königsberg Bridges has five odd vertices, that is, bridges cannot be bypassed without passing twice over some.

PS: In 1905, on the orders of Kaiser Wilhelm, the Imperial Bridge was built, which was subsequently destroyed during the bombing during the Second World War. There are currently seven bridges in Kaliningrad, and the graph still has no Euler path.

Despite the fact that the theory of complex networks includes various networks – electrical, transport, information, the greatest contribution to the development of this theory was made by the study of social networks.

The term "social network" denotes the concentration of social objects, which can be considered as a network (or graph), the nodes of which are objects, and the links are social relations. The term was coined in 1954 by Manchester School sociologist J. Barnes in "Classes and Assemblies in the Norwegian Island Parish". In the second half of the 20th century, the concept of "social network" became popular among Western researchers. In the theory of social networks, such a direction as the analysis of social networks (Social network Analysis, SNA). Today, the term "social network" denotes a concept that turned out to be wider than its social aspect, it includes, for example, many information networks, including WWW.

Barnes, JA " Class and Committees in a Norwegian Island Parish," Human Relations 7:39-58.

In the theory of complex networks, there are three main areas: study of statistical properties that characterize the behavior of networks; creating a network model; predicting the behavior of networks when changing structural properties. Applied research usually uses such typical network analysis characteristics as network size, network density, degree of centrality, etc.

In the analysis of complex networks, as in graph theory, parameters of individual nodes; network parameters in general; network substructures.

The new paradigm – "complex networks" covers networks that have the following properties:

1) big sizes;
2) elements of randomness in the formation;
3) growth (change) in time;
4) some nodes can form compact groups – ensembles.

Network density – the ratio of existing and possible connections: $\Delta = \frac{2L}{n(n-1)}$, *Where* L *is the number of observed links,* n *is the number of nodes in the network.*

1.1. Characteristics of complex networks

1.1.1. Host settings

For individual nodes, the following parameters are distinguished:

- node input power is the number of graph edges that enter the node;
- node output power is the number of graph edges that leave the node;
- distance from one node to others;
- eccentricity – the largest of the minimum distances from this node to others;
- mediation (betweenness), showing how many shortest paths pass through a given node;
- centrality – the total number of connections of a given node in relation to others.

1.1.2. Node degree distribution

Networks in general are characterized by such parameters as the number of nodes, the number of links, the distance between nodes, the average distance from one node to others, network density, the number of symmetric, transitive and cyclic triads, the diameter of the network – the largest distance between nodes in the network, etc..

There are several topical problems in the study of complex networks, among which the following main ones can be distinguished:

- detection of clicks in the network. clicks – these are subgroups or clusters in which nodes are more strongly interconnected than with members of other cliques;
- selection of components (parts of the network) that are not interconnected, the nodes of which are connected within these components;
- finding blocks and jumpers. A node is called a jumper if, when it is removed, the network breaks up into unconnected parts;
- selection of groupings – groups of equivalent nodes (which have the most similar link profiles).

Classification of networks by type can be done in various ways. Networks can be distinguished by what nodes and links are.

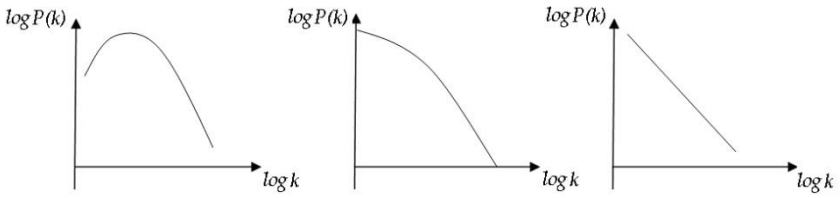
ER – Erdős-Rényi network, the so-called random graph

SF – network *scale free,* scale-free network

For large (N >> 1) networks co random structure one one of the most important characteristics is the distribution function $P(k)$ over the powers of the nodes. Most real CNs are similar (close) to the following three :

1. Random network or Erdős-Rényi (*ER*) network $P(k) \sim e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}$.
   Thus, in the case of an ER network, the distribution function is a Poisson function.
2. Network with exponential distribution $P(k) \sim e^{-k/\langle k \rangle}$.
3. Network with power distribution (Scale – Free) $P(k) = k^{-\gamma} / \varsigma(\gamma) \sim k^{-\gamma}$, where $\varsigma(\gamma)$ is the Riemann zeta function.

In double logarithmic scale these distributions have the form shown in Fig. 1.1.2.



A b V

Figure. 1.1.2 – Distribution densities $P(k)$ in double logarithmic scale: a – Poisson distribution (network *ER)*; b – exponential distribution; c – power distribution

An Erdős-Rényi network can be constructed by randomly distributing M links between N nodes. Then, on the one hand $\langle k \rangle = 2M / N,$, and on the other hand $\langle k \rangle = mN$, where m is the probability of connecting nodes. At $N \to \infty$ and $m \to 0$ the knot degree distribution is Poisson.

BA networks, which are built using a special procedure, which consists in the fact that N_0 new nodes are gradually added to the initially small number of nodes, the links from which are more likely to be connected to those nodes that have links more.

There are procedures for constructing networks of a different type, when random links are added to an ordered structure. The best-known example of such a network is the so-called Small World Network (SWN).

Albert – Laszlo Barabashi

Barabási AL, Albert R. Emergence of scaling in random networks. *Science, 1999.* – Vol. 286 (5439): 509–512.

1.1.3. Shortest path between nodes

The distance between nodes is defined as the number of steps that must be taken to get from one node to another along

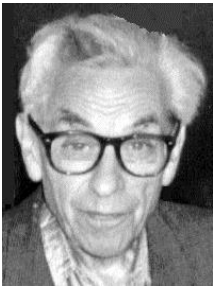
existing edges. Naturally, nodes can be connected directly or indirectly through other nodes.

The shortest path (SP, shortest path) between knots let's call minimum distance between them. For all networks Can enter the concept of the average shortest path, as the average over all pairs of nodes, the minimum distance between them:

$$l = \frac{2}{n(n+1)} \sum_{i \geq j} l_{ij}, \qquad (1.1.1)$$

where n is the number of nodes, l_{ij} is the shortest distance between nodes, i and j.

P. Erdős and A. Rényi showed that the average shortest path in a random graph grows slowly – as a logarithm of the number of nodes.



Paul Erdős (1913-1996)

Erdős, P., Renyi A. On Random Graphs. Publicationes Mathematicae, 1959. – No. 6. – pp. 290-297.

Erdős P., Rényi A. On the evolution of random graphs. Publ. Math. Inst. hungar. Acad. Sci., 1960. – No. 5. – pp. 17-61. – 1960.

The name of P. Erdős is associated not only with studies of complex networks, but also with the popular Erdős number, which is used as one of the criteria for determining the level of mathematicians in the corresponding society, based on the so-called network of co-authorship. It is known that Erdösz wrote about one and a half thousand articles, and also that the number of his co-authors exceeded 500. Such a large number of co-authors gave rise to such a thing as the Erdös number, which is defined as follows: Erdős himself has this number equal to zero; for Erdős' co-authors, this number is equal to one; co-authors of people with an Erdős number equal to one have an Erdős number of two; etc.

Thus, the Erdős number is the length of the path from some author to Erdős himself on joint works. It is a known fact that 90% of mathematicians have an Erdős number no higher than 8, which corresponds to the networks of "small worlds", which will be discussed below.

Some networks may be disconnected, i.e. there are nodes, the distance between which is infinite. Accordingly, the average path may also be equal to infinity. To take into account such cases, we introduce the concept of the average inverse path E between nodes, calculated by the formula:

$$E = \frac{2}{n(n-1)} \sum_{i>j} \frac{1}{l_{ij}}. \qquad (1.1.2)$$

Networks are also characterized by such a parameter as the diameter or the maximum shortest path, which is equal to the maximum value of all l_{ij}.

1.1.4. Clustering coefficient

D. Watts and S. Strogatz in 1998 defined such a network parameter as the clustering coefficient. This coefficient characterizes the tendency to form groups of interconnected nodes, the so-called cliques (Clique). For a particular node, the clustering coefficient indicates how many nearest neighbors of that node are also nearest neighbors to each other.

k *links* coming out of the node that connect it to k other nodes, nearest neighbors. If we assume that all nearest neighbors are connected directly to each other, then the number of connections between them would be $k(k-1)/2$. That is, this is a number that corresponds to the maximum possible number of links that could connect the nearest neighbors of the selected node.

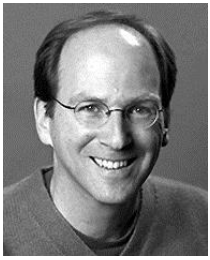
The ratio of the actual number of links that connect the nearest neighbors of a given node i to the maximum possible (the one at which all the nearest neighbors of a given node would be connected directly to each other) is called the clustering coefficient of the node C_i. Naturally, this value does not exceed one.

The clustering coefficient can be determined both for each node and for the entire network:

$$C = \frac{1}{n} \sum_{i=1}^{n} C_i . \qquad (1.1.3)$$

In social networks, one can speak of a "community structure" when there are groups of nodes that have a high density of connections between themselves, while the density of edges between individual groups is low. For large social networks, the presence of a community structure turned out to be an integral property. The traditional method for identifying the structure of communities is cluster analysis. There are dozens of acceptable methods for this, which are based on different measures of distances between nodes, weighted path indices between nodes, and so on.

D. Watts



S. Strogatz

Watts DJ, Strogatz SH Collective dynamics of “small-world” networks. Nature, 1998. – Vol. 393.-pp. 440-442.

1.1.5. Betweenness

Betweenness is a parameter indicating how many shortest paths pass through the node. This characteristic reflects the role of this node in establishing links in the network. The nodes with the most mediation play a major role in establishing links between other nodes in the network. Node m mediation b_m is determined by the formula:

$$b_m = \sum_{i \neq j} \frac{B(i,m,j)}{B(i,j)}, \qquad (1.1.4)$$

where $B(i,j)$ is the total number of shortest paths between nodes i and j, $B(i,m,j)$ is the number of shortest paths between nodes i and j passing through node m.

1.1.6. Network elasticity

The elasticity property of networks refers to the distribution of distances between nodes when removing individual nodes (tolerance to attacks).

The elasticity of a network depends on its connectivity, i.e. the existence of paths between pairs of nodes. If a node is removed from the network, the typical length of these paths will increase.

Albert R., Jeong H., Barabasi A. Error and a ttack tolerance of complex networks. Nature, 2000. – Vol. 406. – pp. 378-382.

In the study of attacks on web servers, the effect of removing the nodes of the network, which is a subset of the web space of 326,000 pages and about 1.5 billion hyperlinks, was studied. For this network, the parameters of the input and output distribution of degrees were determined: $P(k) \sim k^{-\gamma}$, Where $\gamma_{in} = 2{,}1$ and $\gamma_{out} = 2{,}45$.

The average distance between two nodes, as a function of the number of removed nodes, almost did not change when nodes were randomly removed (high elasticity when attacking power-law networks). At the same time, purposeful removal of nodes with the largest number of connections leads to the destruction of the network. Thus, the web space is a highly elastic network in relation to the accidental removal of a node in the network, but highly sensitive to deliberate attack on nodes with high degrees of connections with other nodes.

1.1.7. Examples of calculating the characteristics of networks

The main characteristics of networks will be listed below and at the same time they will be demonstrated using six simple examples. In order to "feel" what exactly these or other characteristics describe, the reader is strongly recommended to get a few "by hand" numbers in the examples.

The six examples of networks considered below will be denoted by numbers No. 1 – No. 6 (Fig. 1.1.3).

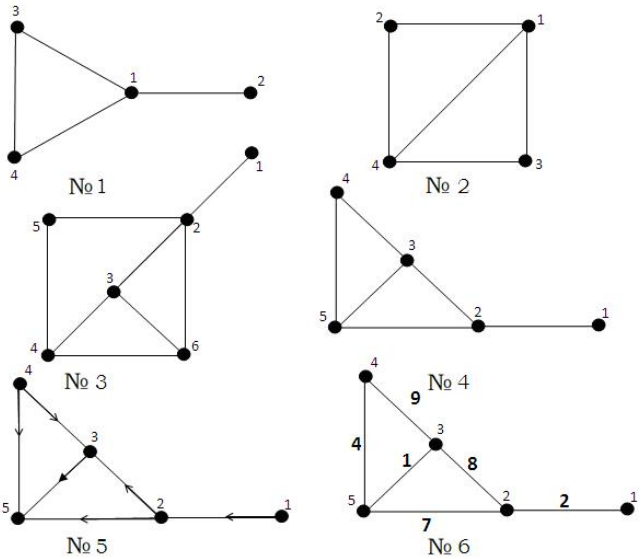


Figure. 1.1.3 – Examples of the simplest networks: network No. 5 – the so-called digraph (directed graph), network No. 6 – a network with link weights (numbers are indicated on the links)

The adjacency matrices of the networks depicted in fig. 4 are below:

$$\boldsymbol{A}_1 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \quad \boldsymbol{A}_2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

$$\boldsymbol{A}_3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{A}_4 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix},$$

For network #5, the adjacency matrix A_5 is not symmetrical.

The adjacency matrix elements A_6 are the link weights.

$$\boldsymbol{A}_5 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{A}_6 = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 8 & 0 & 7 \\ 0 & 8 & 0 & 9 & 1 \\ 0 & 0 & 9 & 0 & 4 \\ 0 & 7 & 1 & 4 & 0 \end{pmatrix}.$$

Distribution of nodes by their degrees $P(k)$ shown in fig. 1.1.4, which also shows such a characteristic of networks as the average number of links per node:

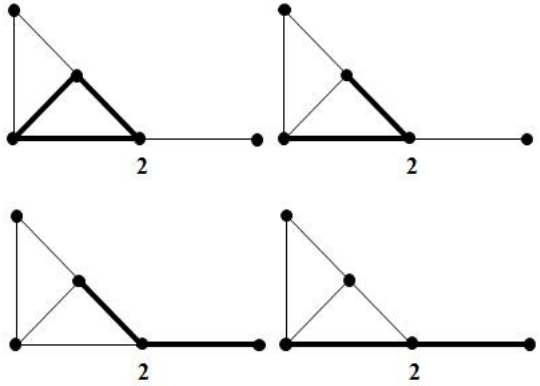
$$\langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i = \sum_{i=1}^{N} k_i P(k_i), \qquad (1.1.5)$$

which is calculated as the ratio of all links in the network to the number of nodes and, at the same time, as the average of a discrete variable k_i with a distribution function $P(k_i)$.

An important characteristic of a node is its clustering coefficient – C_i, which characterizes the connectivity between the neighbors of this node i. The clustering coefficient C_i can be written as the ratio of the number of triangles with a vertex i to the number of forks (two bonds coming out of a node) with a base at that node:

$$C_i = \frac{\text{Number of triangles with vertex } i}{\text{Number of forks with vertex } i}. \qquad (1.1.6)$$

Let us consider, for example, nodes 2 and 3 networks No. 4. For node No. 2, thick lines denote forks (there are three of them) and triangles (it is one).



As you can see, there are three forks and one triangle for the second node, so $C_2 = 1/3$.

For the third node, there are three forks and two triangles – $C_2 = 2/3$.

Another network is shown in Fig. 1.1.4.

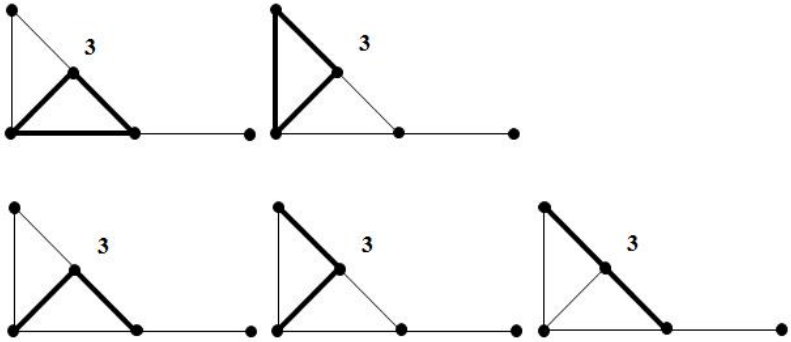


Figure. 1.1.4 – Network options

The clustering coefficient of the entire network is calculated by the formula:

$$C = \langle C_i \rangle = \frac{1}{N}\sum_{i=1}^{N} C_i. \qquad (1.1.7)$$

$$C_i = \frac{\sum \Delta_i}{\sum \vee_i}.$$

C is the network clustering coefficient.

In table. 1.1.1 shows the clustering coefficients for all nodes of networks No. 1, No. 2, No. 3 and No. 4, as well as the clustering coefficient of the entire network C. On fig. 1.1.5. The distribution of nodes by degrees for the networks shown in fig. 1.1.3.

As you can see, network No. 2 has the highest clustering coefficient; on average, it has the largest number of neighbors of each node interconnected. Network No. 3 has the smallest clustering coefficient.

Table 1.1.1 Clustering coefficients

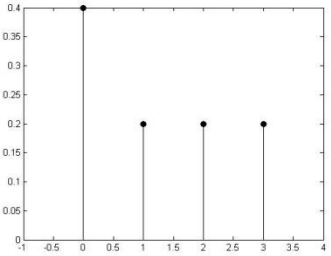
Net	C_1	C_2	C_3	C_4	C_5		C_6	C
No. 1	1/3	0	1	1	1			7/12
No. 2	2/3	1	1	1	2/3			10/12
No. 3	0	1/6	2/3	2/3	1/3	0	2/3	11/36
No. 4	0	1/3	2/3	2/3	1	2/3		8/15

The clustering coefficient of a node can be calculated without resorting to listing triangles and forks, directly from the adjacency matrix:

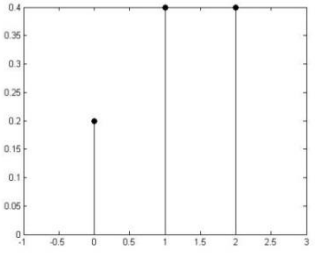
$$C_i = \frac{\sum_{j,m} A_{ij}A_{jm}A_{mi}}{k_i\left(k_i - 1\right)}, \qquad k_i = \sum_j A_{ij}, \qquad (1.1.8)$$

where the summation is over all nodes.

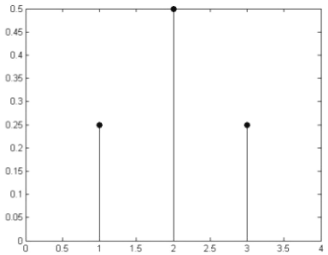
Using examples of simple networks, it is easy to verify directly that definitions (1.7) and (1.8) give the same values of C_i .



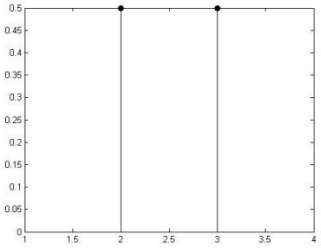
Network No. 1, $\langle k \rangle = 2$



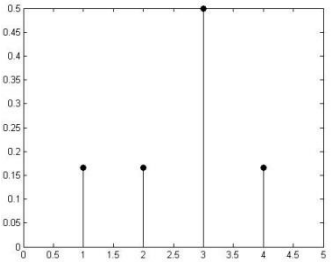
Network number 2, $\langle k \rangle = 2{,}5$



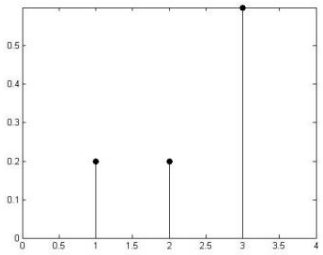
Network No. 3, $\langle k \rangle = 8/3 \approx 2.67$



Network No. 4, $\langle k \rangle = 12/5 = 2.4$



Network No. 5, incoming links $\langle k \rangle = 1.2$



Network No. 6, outgoing links $\langle k \rangle = 1.2$

Figure. 1.1.5. Distribution of nodes by degrees for the considered networks

In addition to the definition of the clustering coefficient of the entire network (1.1.7) and (1.1.8), there is another definition in the literature that is close, but not identical, sometimes called transitivity – T :

T – transitivity, a characteristic close to the clustering coefficient.

$$T = \frac{\sum_{i=1}^{N}\left(\boldsymbol{A}^3\right)_{ii}}{\sum_{i=1}^{N} k_i\left(k_i - 1\right)} \equiv \frac{Tr\boldsymbol{A}^3}{\sum_{i=1}^{N} k_i\left(k_i - 1\right)}, \qquad (1.1.9)$$

which is expressed in terms of the number of triangles and forks in the entire network as follows:

$$T = 3\frac{\text{Number of triangles in the network}}{\text{Number of plugs in the network}} \qquad (1.1.10)$$

For comparison, below are the values of the clustering and transitivity coefficient for networks No. 1, No. 2, No. 3 and No. 4 – Table. 1.1.2.

Table 1.1.2. Clustering coefficients

Net	No. 1	No. 2	No. 3	No. 4
C	$\frac{7}{12} \approx 0.58$	$\frac{10}{12} \approx 0.83$	$\frac{11}{36} \approx 0.31$	$\frac{8}{15} \approx 0.53$
T	$\frac{3}{5} = 0.6$	$\frac{3}{4} = 0.75$	$\frac{3}{35} \approx 0.086$	$\frac{3}{5} = 0.6$

Another important characteristic of networks is the average minimum distance between their nodes:

$$l = \left\langle l_{ij} \right\rangle = \frac{1}{N(N-1)} \sum_{i \neq j} l_{ij}, \qquad (1.1.11)$$

where l_{ij} is the least number of steps from node i to node j. In this case, each step corresponds to a unit distance.

A similar definition can be used in loaded networks, while each step can correspond to both a distance proportional and inversely proportional to the weight:

$$\tilde{l} = \frac{1}{\left\langle 1/l_{ij} \right\rangle} = \left(\frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{l_{ij}} \right)^{-1}. \qquad (1.1.12)$$

In the case of the arithmetic mean (1.1.11), the largest contribution will come from the longest paths (out of the shortest ones), while in the case of the harmonic mean, the shortest ones. If there are isolated nodes in the network that cannot be reached and for which it is natural to consider $l_{ij} = \infty$, the definition of the minimum average distance as an arithmetic average ceases to be informative, since even if there is one such isolated node $l = \left\langle l_{ij} \right\rangle = \infty$. In such a situation, it is convenient to use the definition of the average minimum distance as the harmonic mean – $\tilde{l}$. Sometimes the reciprocal $\tilde{l}$ is denoted $E = 1/\tilde{l}$ and called efficiency (efficiency).

The smallest number of steps l_{ij} from node i to node j can be written as a matrix ***SP*** (short path). The numerical values of this matrix for networks No. 1 – No. 6 are equal to:

$$\mathbf{SP}(1) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 2 & 2 \\ 1 & 2 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{pmatrix}, \qquad \mathbf{SP}(2) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

$$\mathbf{SP}(3)=\begin{pmatrix}0&1&2&3&2&2\\1&0&1&2&1&1\\2&1&0&1&2&1\\3&2&1&0&1&1\\2&1&2&1&0&2\\2&1&1&1&2&0\end{pmatrix},\qquad \mathbf{SP}(4)=\begin{pmatrix}0&1&2&3&2\\1&0&1&2&1\\2&1&0&1&1\\3&2&1&0&1\\2&1&1&1&0\end{pmatrix},$$

$$\mathbf{SP}(5)=\begin{pmatrix}0&1&2&\infty&2\\\infty&0&1&\infty&1\\\infty&\infty&0&\infty&1\\\infty&\infty&1&0&1\\\infty&\infty&\infty&\infty&0\end{pmatrix},\qquad \mathbf{SP}(6)=\begin{pmatrix}0&2&10&13&9\\2&0&8&11&7\\10&8&0&5&1\\13&11&5&0&4\\9&7&1&4&0\end{pmatrix}.$$

Note that the matrix $\mathbf{SP}(5)$ (for a directed network) contains infinite elements, for example $l_{21}(5)=\infty$ (compare with $l_{21}(4)=1$). This, of course, means (see Figure 5 of the network) that it is impossible to get from node 2 to node 1.

In a network with weights, l_{ij} each step along a link is multiplied by its weight, so that not just the number of steps (links) is summed up, but their weights. This is precisely why the shortest path in network 6 between nodes 3 and 4 includes two steps with weights 4 and 1, and not one step with weight 1.

The average values of the shortest paths (1.1.11) for the networks under consideration are:

$$\langle l(1)\rangle=\frac{4}{3}\approx 1.3,\qquad \langle l(2)\rangle=\frac{7}{6}\approx 1.16,\qquad \langle l(3)\rangle=\frac{23}{15}\approx 1.53,$$

$$\langle l(4)\rangle=\frac{3}{2}=1.5,\qquad \langle l(5)\rangle=\infty,\qquad \langle l(6)\rangle=7.$$

harmonic mean:

$$\left\langle \tilde{l}(1) \right\rangle = 1.2, \qquad \left\langle \tilde{l}(2) \right\rangle \approx 1.09, \qquad \left\langle \tilde{l}(3) \right\rangle \approx 1.32,$$

$$\left\langle \tilde{l}(4) \right\rangle \approx 1.28, \qquad \left\langle \tilde{l}(5) \right\rangle \approx 2.86, \qquad \left\langle \tilde{l}(6) \right\rangle \approx 3.85.$$

Another important characteristic of the network is the so-called mediation (betweenness). Table 1.1.3 shows the numerical values of mediation for all network nodes No. 1 – No. 6.

Table. 1.1.3. Mediation values for networks #1 – #6.

Knot / network number	B_1	B_2	B_3	B_4	B_5	B_6
1	4	0	0	0		
2	1	0	0	1		
3	0	10	4/3	2	4/3	4/3
4	0	6	2	0	2	
5	0	2	0	0	0	
6	0	6	0	0	8	

Recall that when calculating mediation in a network with weight, the shortest path is calculated taking into account the weight of the links.

In addition to the main characteristic of the network – the distribution of the degree of nodes over links – $P(k)$, see fig. 1.1.2 also introduces the distribution of bond ends – $P_{nn}(k)$:

$P_{nn}(k)$ - distribution of bond ends.

$$P_{nn}(k) = \frac{kP(k)}{\langle k \rangle}. \qquad (1.1.13)$$

This distribution is also called the degree distribution of nearest neighbors (degree distribution of the nearest neighbor), which is where the notation comes from $P_{nn}(k)$. Using a simple example of network No. 1, it is easy to verify that (1.1.13) does indeed give the distribution of the ends of the connections. For network No. 1, we have eight ends, i.e. the probability that one end of the link hits the node is $1/8$. The node has 1one link, so $P_{nn}(узел_1)=1/8$. Each node has 3 and 4 – $2/8$, i.e. $2\cdot 2/8=1/2$, thus $P_{nn}(узел_3)=P_{nn}(узел_4)=1/2$, $P_{nn}(узел_2)=1/2$ and for node 1 – $3\cdot 1/8$, i.e. $P_{nn}(узел_3)=3/8$.

On the other hand, the same values are immediately obtained from (1.1.13), for example, for the first node

$$P_{nn}(узел_1)=\frac{1P(1)}{\langle k\rangle}=\frac{1\cdot 1/4}{2}=\frac{1}{8},$$

and similarly for others.

Qualitatively, the expression for $P_{nn}(k)$(1.1.13) can be explained as follows. The probability that a randomly selected link hits nodes with k links is proportional to the number of all links of such nodes $P_{nn}(k)\sim NkP(k)$, where N is the total number of nodes in the network. Taking into account the fact that $\sum P_{nn}(k)=1$, the normalization constant is equal to $1/\langle k\rangle$, whence (1.1.13) follows.

The distribution $P_{nn}(k)$for simple networks coincides with $P(k)$only in exceptional cases (when the equality holds for each node $k=\langle k\rangle$), for example, for an infinite square lattice.

Having the distribution of the ends of the links, $P_{nn}(k)$we can introduce the degree of nodes averaged over this distribution :

$$\langle k\rangle_{nn}=\sum_{k=1}^{N}kP_{nn}(k)=\sum_{k=1}^{N}k\frac{kP(k)}{\langle k\rangle}=\frac{\langle k^2\rangle}{\langle k\rangle}. \qquad (1.1.14)$$

For an uncorrelated network, knowing the distribution of ends $P_{nn}(k)$ allows you to find the probability of connecting nodes with powers k and k' - $P(k,k')$. Because the probability that the link "sticks" into a node with a degree k is equal to $P_{nn}(k)=kP(k)/\langle k\rangle$, the probability that it simultaneously "sticks" and into a link with a degree k' is equal to the product

$$P(k,k')=P_{nn}(k)P_{nn}(k')=\frac{kP(k)k'P(k')}{\langle k\rangle^2}. \qquad (1.1.15)$$

We now introduce the concept of the average number of first, second, and next neighbors. If a node has degree k, then it has k neighbors, so the average number of nearest (first) neighbors z_1 is, as it should be:

$$z_1=\sum k\cdot p_k=\langle k\rangle. \qquad (1.1.16)$$

The first neighbors of the node, in turn, have their own neighbors, which it is logical to call the second neighbors closest to the original node. If the first neighbor has degree k, then it has $k-1$ neighbors (one of the links "left" to the original node).

The number of first neighbors with degree k for all nodes is $N\cdot k\cdot p_k$, so the number of all second neighbors is:

$$z_2=\frac{1}{N}\sum(k-1)k\cdot p_k=\sum\left(k^2-k\right)p_k=\langle k^2\rangle-\langle k\rangle. \qquad (1.1.17)$$

The ratio of the average number of second neighbors to the first:

$$B = \frac{z_2}{z_1} = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \qquad (1.1.18)$$

is called the branching factor (branching coefficient), it characterizes the degree of "multiplication" of bonds as you move away from the node.

The branching factor can also be obtained from such considerations. Probability $P_{nn}(k)$ can be interpreted as the probability $Q(k-1)$ that a link from some selected node A will enter a node B with $k-1$ links, so that the node B will have in the sum k of links:

B – branching factor

$$Q(k-1) = P_{nn}(k) = \frac{kP(k)}{\langle k \rangle}. \qquad (1.1.19)$$

Then the average number of links outgoing from the node B (without taking into account the connection with the node A) is equal to:

$$\sum_{k=0}^{N} kQ(k) = \sum_{k=0}^{N} \frac{k(k+1)P(k+1)}{\langle k \rangle} = \\ = \sum_{k=1}^{N} \frac{(k-1)kP(k)}{\langle k \rangle} = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}. \qquad (1.1.20)$$

The last equality is valid for $N \gg 1$.

Knowing the branching factor B (1.1.18), it is easy to calculate the average number of the following (third, fourth, etc.) neighbors:

$$z_m = Bz_{m-1} = \frac{z_2}{z_1} z_{m-1} = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} z_{m-1}, \quad (1.1.21)$$

$$z_m = \left(\frac{z_2}{z_1} \right)^{m-1} z_1. \quad (1.1.22)$$

The expression for the average number $m-$ of neighbors makes it possible to answer a very important question: does the network have a so-called giant cluster (Giant connected Cluster or Giant connected component).

Those. whether there is a network cluster (an interconnected part of the nodes), which includes the number of nodes of the order of the total number of nodes in the network. When calculating more and more distant neighbors ($m >> 1$) according to (1.1.22), two scenarios are possible: when $z_2 > z_1, \quad z_m >> 1$ and at $z_2 < z_1, \quad z_m << 1$.

GC – Giant Cluster – a giant cluster

In the second case (it is assumed that $N >> 1$) from (1.1.21) it follows:

$$\sum_m z_m = z_1 \sum_m \left(\frac{z_2}{z_1} \right)^{m-1} \approx \frac{z_1^2}{z_1 - z_2}, \quad (1.1.23)$$

and strict equality, of course, takes place in the limit $N \to \infty$.

From (1. 1.23) it immediately follows that at $z_2 \geq z_1$, i.e. at

$$\langle k^2 \rangle - \langle k \rangle \geq \langle k \rangle, \quad (1.1.24)$$

there is an infinite cluster in the network.

Inequality (1.1.24) is called the Molloy-Read criterion, it is often written in this form

Molloy-Read criterion

$$\langle k^2 \rangle - 2\langle k \rangle \geq 0. \quad (1.1.25)$$

Thus, according to (1. 1.25), a giant cluster exists if the number of second neighbors is greater than the average number of links leaving the node.

The existence of a giant cluster also means that by choosing arbitrarily distant nodes from each other, we can pass through the network from one node to another with a non-zero probability. That is why the values of the parameters at which $\langle k^2 \rangle$ it reaches the value $2\langle k \rangle$ are referred to as the percolation threshold. The value of the percolation threshold p_c in this case is equal to:

$$p_c = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle}. \qquad (1.1.26)$$

1.2. Artifact network models

1.2.1. Erdős-Rényi networks

A network (graph) of Erde sha Renyi (ER -network) is such a network when each pair of nodes is connected with probability p. In the limit of a large number of nodes, N the distribution function of degrees of nodes has the form:

$$P_k = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}. \qquad (1.2.1)$$

In the limit $N \to \infty$, the value in $\langle k \rangle$ the ER network constructed in this way is uniquely determined. In a real case, for a finite value of the number of nodes, two models of the ER network should be distinguished – the Hilbert model (G_{np} -model) and the Erdős-Renyi model itself (G_{nm}). The probability is fixed p in the model G_{nm}. For a network with a finite number of nodes, N this means that $\langle k \rangle = pN$, while the number of connections M is determined only on average

$\langle M \rangle = pN(N-1)/2$. In the Hilbert model, G_{np} not the probability is given p, but the number of connections M, now the probability p is defined as $\langle M \rangle = pN(N-1)/2$, and the average degree of the node is $\langle k \rangle = 2M/N$. The distinction between these models is analogous to the distinction between canonical and microcanonical ensembles in statistical physics. In the microcanonical ensemble, the energy of the system is given, and in the canonical ensemble, the temperature, while the energy fluctuates around the mean value.

In the limit $N \to \infty$, $p \to 0$, pN is a finite number not equal to zero, both definitions of the Erdős-Rényi and Hilbert networks coincide.

The Erdős-Rényi network is "well connected" – the average minimum distance between nodes of order $\ln N << N$ ($N >> 1$). The average minimum distance between nodes can be easily estimated from the following considerations. Each neighbor also has, on average, $\langle k \rangle$ neighbors, each of which can be reached in two steps. For l steps, you can reach $\langle k \rangle^l$ the nodes on average.

Then for the average minimum distance l between the nodes of the network from N the nodes we get:

For *ER network*:
$l \sim \ln N$

$$l = \frac{\ln N}{\ln \langle k \rangle}. \qquad (1.2.2)$$

A very small (logarithmically small compared to the number of nodes) value of the minimum distance makes a random ER network a so-called "small world". For each node of a network consisting, for example, of $N = 10^9$ nodes (the order of the number of people on Earth) with an average number of connections $\langle k \rangle = 100$ (approximately how many people we

personally know), the minimum average distance is $l = \ln 10^9 / \ln 10^2 \approx 4.5$, i.e. no more than five steps.

Note that for a regular network, for example, for a square lattice, this distance is much larger $l \sim N^{1/2} >> \ln N$, for the given example $l \sim \sqrt{10^9} \approx 30000 >> 4,5.$

The critical value p_c at which a giant cluster is born in the ER network is immediately found from the Molloy-Read criterion $p_c = \langle k \rangle / \left(\langle k^2 \rangle - \langle k \rangle \right)$. Since $\langle k^2 \rangle = \langle k \rangle \left(\langle k \rangle + 1 \right)$, then for p_c we get:

$$p_c = \frac{1}{\langle k \rangle}. \qquad (1.2.3)$$

1.2.2. Scale invariant networks

The node degree distribution function P_k for a scale-invariant network (Scale – Free – SF) has the form:

$$P_k = \frac{1}{k^\gamma}, \quad k \geq 0. \qquad (1.2.4)$$

Such a distribution is well known in probability theory as the Pareto distribution. Typically, for real networks, the indicator γ ranges from 2 to 3.

In the case when k it can be considered a continuous variable, we denote it – x ($x > 0$) it is necessary to take into account that in all real cases there is a minimum value of this variable – $x_{\min}$. Normalization constant C for the continuous case is determined, in this case, from the condition $\int_{x_{\min}}^{\infty} p(x) dx = 1$ and is equal to

$$p(x) = Cx^{-\gamma}, \quad C = (\gamma - 1) / x_{\min}^{\gamma - 1}, \qquad (1.2.5)$$

whence it is clear that the distribution (1.31) makes sense only for $\gamma > 1$.

An additional restriction $\gamma > 2$ on the indicator is imposed by the requirement to have a finite average value - $\langle x \rangle$:

$$\langle x \rangle = \int_{x_{\min}}^{\infty} xp(x)dx = \frac{\gamma - 1}{\gamma - 2} x_{\min}, \quad \gamma > 2, \qquad (1.2.6)$$

and for the existence m of the -th moment

$$\langle x^m \rangle = \frac{\gamma - 1}{\gamma - 1 - m} x_{\min}^m, , \qquad (1.2.7)$$

condition is required $\gamma > 1 + m$.

Associated with the distribution (1.2.5) is the so-called rule $80/20$, in a joking version – "20% of people drink 80% of beer", and it is assumed that this kind of ratio takes place for many other human activities.

Indeed, the proportion of nodes $S(x)$ with a degree value greater than x, is:

$$S(x) = \int_{x}^{\infty} p(x)dx = \left(\frac{x}{x_{\min}} \right)^{-\gamma+1}. \qquad (1.2.8)$$

These nodes in total contain a share $W(x)$ of all links

$$W(x) = \frac{\int_{x}^{\infty} xp(x)dx}{\int_{x_{\min}}^{\infty} xp(x)dx} = \left(\frac{x}{x_{\min}} \right)^{-\gamma+2}. \qquad (1.2.9)$$

From (1.2.8) and (1.2.9) it immediately follows

$$W = S^{\frac{\gamma-2}{\gamma-1}} = S^{1-\frac{1}{\gamma-1}}. \qquad (1.2.10)$$

On fig. 1.2.1 shows the dependence W on S for various values of γ. Obviously, the greater the value of the index γ, the $W = W(S)$ closer the dependence is to a linear one.

With an indicator $\gamma = 2.161$ for, $S = 0,2$ the value $W = 0.8$ corresponds to the Pareto rule (80/20). The greater the value of the index γ, the $W = W(S)$ closer the dependence is to a linear one.

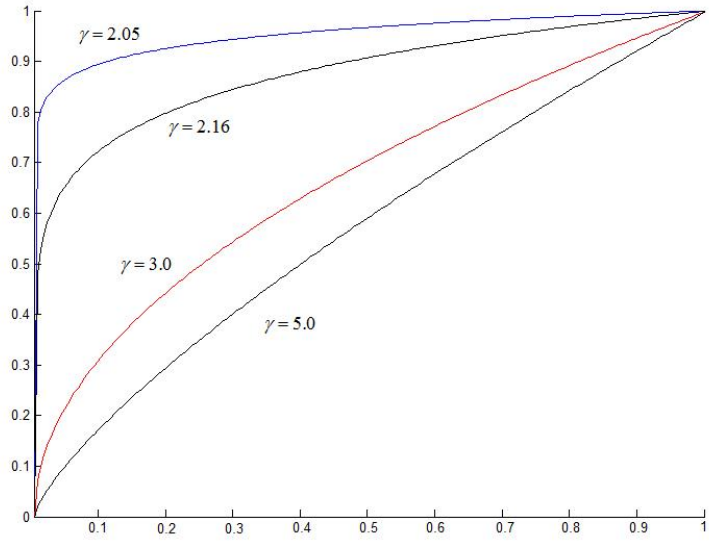


Figure. 1.2.1 – Dependency $W = W(S)$

Let us now explain what the term “scale-free” means in relation to the SF network. For clarity, we will assume that the degree of the node (the value of the quantity x) is the wealth that a person (this node) possesses. Then you can calculate the relative proportion $S(x_M)$ of people with great wealth x_M, who own half of all money.

With an indicator $\gamma = 2.161$ for, $S = 0,2$ the value $W = 0.8$ corresponds to the Pareto rule (80/20). The greater the value of the index γ, the $W = W(S)$ closer the dependence is to a linear one.

Let us now explain what the term “scale-free” means in relation to the SF network. For clarity, we will assume that the degree of the node (the value of the quantity x) is the wealth that a person (this node) possesses. Then you can calculate the relative proportion $S(x_M)$ of people with great wealth x_M, who own half of all money.

On the one side:

$$S(x_M) = \int_{x_M}^{\infty} p(x)dx = \left(\frac{x_M}{x_{\min}}\right)^{-\gamma+1}, \qquad (1.2.11)$$

on the other hand, a share of their wealth $W(x_M) = 1/2$, i.e.

$$W(x_M) = \frac{\int_{x_M}^{\infty} xp(x)dx}{\int_{x_{\min}}^{\infty} xp(x)d} = \left(\frac{x_M}{x_{\min}}\right)^{-\gamma+2} = \frac{1}{2}. \qquad (1.2.12)$$

Substituting values x_M from (1.2.11) to (1.2.12) we find

$$S(x_M) = 2^{-\frac{\gamma-1}{\gamma-2}}, \qquad S(x_M)\big|_{\gamma=2.161} = 6.7 \cdot 10^{-3} \qquad (1.2.13)$$

,

those. for the value of the indicator, $\gamma = 2.161$ less than 1% (0.67%) of people own half of all money.

Let's call these people "rich". Scale-free means that among the "rich" the distribution between "rich" and "poor" is exactly the

same. That is, how easy it is to calculate that 0.67% of the "rich" own half of the "wealth" of the "rich" (i.e. 1/4 of all wealth).

For any value b for scale-free distribution

$$P(bx) = g(b)P(x), \qquad (1.2.14)$$

those. rescale ($x \to bx$) only results in multiplying the distribution by a constant.

Another often cited example for the SF distribution is related to "computer life". If the PC has files of size 2 Since B is a quarter of 1 KB files, then the number of 2 MB files will be a quarter of the number of 4 MB files.

In the discrete version of SF, the distribution is normalized a little more complicated:

$$1 = \sum_{k=1}^{\infty} P_k = C\sum_{k=1}^{\infty} \frac{1}{k^{\gamma}} = C\zeta(\gamma), \qquad (1.2.15)$$

where $\zeta(\gamma) = \sum_{k=1}^{\infty} k^{-\gamma}$ is ζ the Riemann function.

Thus, according to (1.2.15):

$$P_k = \frac{k^{-\gamma}}{\zeta(\gamma)}$$

Dorogovtsev, AV Goltsev, JFF Mendes. Critical phenomena in networks complex, arXiv:0705.0010.

In the case when the "poor tail" is cut off, i.e. nodes with degrees less than $k_{\min}$ are not considered, the distribution p_k is written as follows:

$$p_k = \frac{k^{-\gamma}}{\zeta(\gamma, k_{\min})}, \quad k \geq k_{\min}, \qquad (1.2.16)$$

a is $\zeta(\gamma, k_{\min})$ equal to

$$\zeta(\gamma, k_{\min}) = \sum_{k=k_{\min}}^{\infty} k^{-\gamma}, \qquad (1.2.17)$$

A network (graph) with an SF node degree distribution can be either random or deterministic. There are many "toy" examples of deterministic networks with SF distribution, for example, the so-called (u,v)-flowers and (u,v)-trees. The construction (u,v) of a -flower begins with a chain $w = u + v$ of links, after which, at each next step, each link is replaced by a chain of two parts u and v, as shown in Fig. 5.

On fig. 1.2.2 shows several steps for constructing (1.2), (1.3), and (2.2) colors.

At n the th construction step, the number of connections in such a network is $M_n = (u+v)^n$.

At the same time, the number of nodes at n the -th step is N_n obeys the following iterative relation

$$N_n = wN_{n-1} - w. \quad w = u + v. \qquad (1.2.18)$$

where

$$N_n = \left(\frac{w-2}{w-1}\right) w^n + \frac{w}{w-1}. \qquad (1.2.19)$$

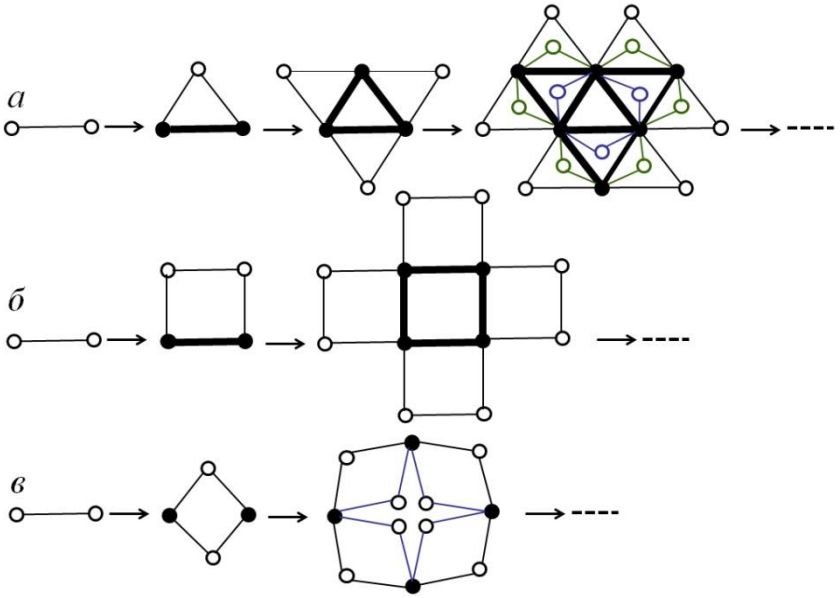


Figure. 1.2.2 – An example of s construction (u,v) of –flowers: a – (1,2)-flower; b – (1,3)-flower; c – (2,2)-flower:

○ – nodes appearing at the given construction step;

● - "old nodes"; bold lines are links that appear at a given construction step.

According to the rule for constructing (u,v)-colors, at n the -th step there are nodes only with degrees

$$k = 2^m, m = 1,2,...,n, \qquad (1.2.20)$$

denoting N_n - the number of nodes at a step n with a degree 2^m, we can write the following iterative relation :

$$N_n(m) = N_{n-1}(m-1) + (w-2)w^{n-1}\delta_{m,1} \qquad (1.2.\ 21)$$

From :

$$N_n(m) = \begin{cases} (w-2)w^{n-m}, & m < n, \\ w, & m = n. \end{cases} \qquad (1.2.22)$$

T ak as with $n \gg 1$ $N_n(m) \sim p(k)$ from $N_n(m) \sim w^{-m}$ and $k = 2^m$ it follows that

$$p(k) \sim k^{-\gamma}, \qquad (1.2.\ 23)$$

Where

$$\gamma = 1 + \frac{\ln w}{\ln 2} = 1 + \frac{\ln(u+v)}{\ln 2} \qquad (1.2.24)$$

And thus (u,v)-flowers are indeed SF-networks.

For a network with $p_k \sim k^{-\gamma}$ there are nodes with the maximum degree value – $k_{\max}$. The value of $k_{\max}$, of course, depends on the total number of nodes – N. This dependence is power-law and depends on the exponent as follows γ:

$$k_{\max} \sim N^{-\frac{1}{\gamma-1}}. \qquad (1.2.25)$$

The clustering coefficient for SF for a random graph is determined from

$$C = \frac{\langle k \rangle}{N} \left(\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle^2} \right)^2, \qquad (1.2.26)$$

where for $\gamma < 3$ we have $\langle k^2 \rangle \sim k_{\max}^{3-\gamma}$ and, thus:

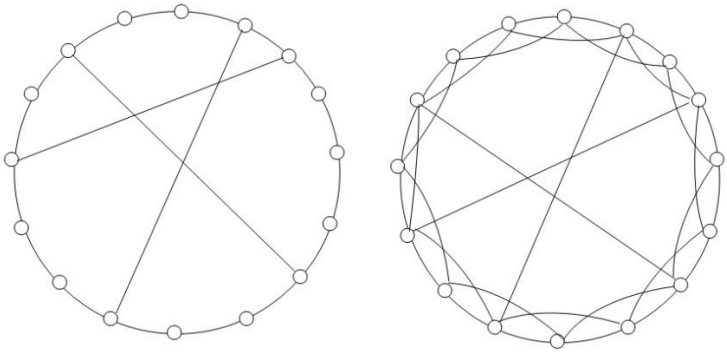
$$C \sim N^{-\beta}, \quad \beta = \frac{3\gamma - 7}{\gamma - 1}. \qquad (1.2.27)$$

1.2.3. Watts–Strogatz Small World Networks

In the 70s of the last century, the American psychologist Milgram (Milgram) conducted an interesting study. He wondered what the "distance" was between two randomly selected people. Distance refers to the number of acquaintances required to establish a connection between these people. Milgram acted as

follows – since he lived in Boston, a city far from Boston was chosen – Nebraska, and envelopes were handed out to randomly selected people, which had to be transferred to Boston. Envelopes could be transferred only through their acquaintances and relatives. Milgram got a very unexpected result: on average, each envelope passed through six people. And so the theory of "six handshakes" was born. Those. each person is connected with any other by a chain of no more than six personal acquaintances. In this sense, our world is spoken of as a small world – " small world ".

The model of transition from a large (regular) world to a small one was proposed by Watts (Watts) and Strogatz (Strogatz). This model is a one-dimensional regular lattice consisting of N nodes, where each node is connected only to its k nearest neighbors and periodic boundary conditions are imposed, i.e. the lattice was rolled into a ring, see fig. 1.2.3. After that, each connection $\phi \ll 1$was transferred with probability to another randomly selected node. True, with such a procedure, there is a possibility of the appearance of isolated nodes.



A b

Figure. 1.2.3 – An example of a small world with three hops ($N = 16$): a – each node is connected to its nearest neighbors ($k = 2$), b – each node is connected to four neighbors ($k = 4$)

Formally, the distance to them from any node will be infinite. To avoid this, Newman and Watts proposed not to transfer connections, but simply to add them. Let's take a closer look at this version of the model.

The average distance between the ends of the added links is: $N/(2\cdot\phi\cdot N)=1/(2\cdot\phi)$. For convenience, we omit the two in the denominator and define it ξ as:

$$\xi=\frac{1}{\phi}. \qquad (1.2.28)$$

For $k\geq 2$ a natural generalization gives:

$$\xi=\frac{1}{k\cdot\phi}. \qquad (1.2.29)$$

Since there is only one characteristic size of the system ξ, the dimensionless ratio of the average distance between graph nodes to the number of all graph nodes l/N can depend only on the dimensionless quantity N/ξ. Those. you can write:

$$l=N\cdot f\left(\frac{N}{\xi}\right), \qquad (1.2.30)$$

where $f(x)$ is a scaling function with the following asymptotics:

$$f(x)\sim\begin{cases} cons, & x\ll 1 \\ \dfrac{\log(x)}{x} & x\gg 1 \end{cases}. \qquad (1.2.31)$$

As mentioned above, there are many ways to determine the correlation radius. Let's assume that $\xi\sim\phi^{-\tau}$. Let us show with the help of the renormalization group transformation (for $k=2$) that $\tau=1$. So let's have:

$$l=N\cdot f\left(N\cdot\phi^{\tau}\right). \qquad (1.2.32)$$

Let us perform the renormalization group transformation shown in Fig. 1.2.4, namely: we combine neighboring nodes in the

graph into pairs, while in the new graph the nodes are connected by an added link, if at least one of the pairs had such a link in the original graph.

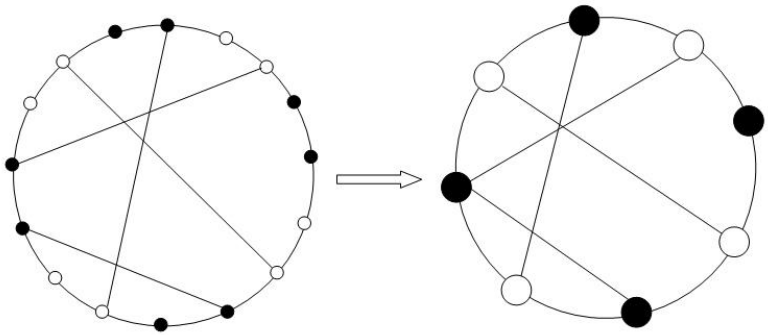


Figure. 1.2.4 – Renormalization group transformation. Two adjacent black nodes are merged into one big black node in the new graph, and similarly for white nodes

With such a transformation, it can be clearly written that:

$$N' = \frac{1}{2} N \ . \ \phi' = 2 \cdot \phi \ . \qquad (1.2.33)$$

where the dashed values refer to the right graph in Fig. 1.2.4.

It is also clear that the average minimum distance in the new graph l' will be two times different.

$$l' = \frac{1}{2} l \ . \qquad (1.2.34)$$

Substituting (1.2.33) and (1.2.34) into (1.2.32) we get:

$$\tau = \frac{\log\left(N/N'\right)}{\log\left(\phi'/\phi\right)} = 1 . \qquad (1.2.35)$$

For, $k > 2$ a similar transformation can be carried out, only now it is necessary to group not by two nodes, but by $k/2$ nodes. The result naturally remains the same – $\tau = 1$.

The small world model described above can be generalized to large dimensions. So, for example, in the two-dimensional case it can be a regular square lattice with additional constraints, as shown in Fig. 1.2.5. Here below, by k is understood the degree of the node, i.e., number of nearest neighbors. But one must keep in mind that in the literature, when describing a small world, the term is k often understood as the number of neighbors in one direction.

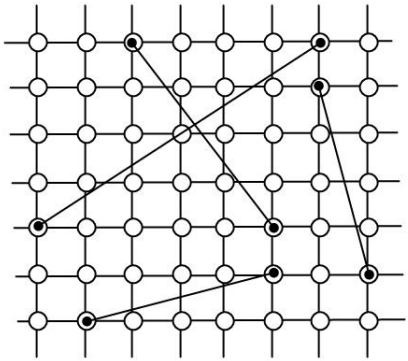


Figure. 1.2.5 – An example of a two-dimensional ($d=2$) small world, $k=4$.

Then instead of (1.2.29) we will have:

$$\xi = \frac{1}{\left(k \cdot \phi \cdot d\right)^{1/d}}, \qquad (1.2.36)$$

where is d the dimension of the small world. And the expression (1.2.30) will take the form:

$$l = \frac{N}{k} \cdot f\left(\left(\phi \cdot k\right)^{1/d} \cdot N\right). \qquad (1.2.37)$$

For a one-dimensional small world, one can explicitly find the clustering C and the average minimum distance l for a small world; here are the final expressions:

$$c(\phi) = \frac{3 \cdot (k-2)}{4 \cdot (k-1)} \cdot (1-\phi)^3 \qquad (1.2.38)$$

And

$$l(\phi) = \frac{1}{\phi \cdot k \cdot \sqrt{1 + \frac{2}{N \cdot \phi}}} \cdot \operatorname{arth}\left(\frac{1}{\sqrt{1 + \frac{2}{N \cdot \phi}}} \right). \qquad (1.2.39)$$

As it should be when writing the expression (1.2.39) in the form (1.2.38), the function $f(x)$ has the following asymptotics:

$$f(x) \sim \begin{cases} \frac{1}{4}, & x \ll 1 \\ \frac{\log(2 \cdot x)}{4 \cdot x} & x \gg 1 \end{cases}. \qquad (1.2.41)$$

On Fig. 1.2.6. plots of normalized dependences of clustering $\bar{C}$ and average distance $\bar{l}$ on the concentration of umklapps are given p.

You can read more about scaling in the small world here:

M. EJ Newman and DJ Watts. Scaling and percolation in the small-world network model. Phys. Rev. E 60, 7332–7342 (1999) (arXiv: 9904419v2).

About renormalization group transformation Can read here : *Newman M.E.J., Watts DJ* Renormalization group analysis of the small-world network mode. Phys. Lett. A 263, 341–346 (1999).

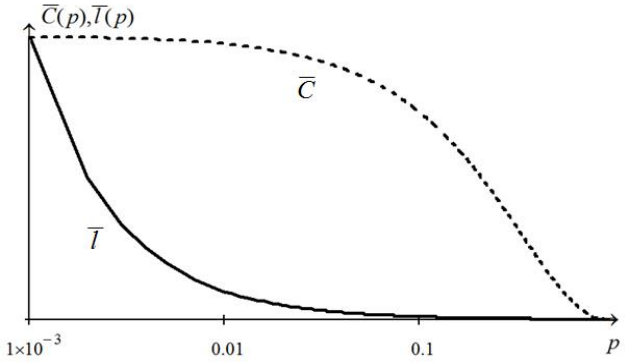


Figure 1.2.6 – Dotted line – normalized clustering. Solid line – normalized medium minimum distance. Normalization takes place on a regular graph (without transfers) – $\overline{C}(0)=1,\ \overline{l}(0)=1$

An ordinary regular graph (for example, a grid) is characterized by a large average minimum distance and a large (close to unity) clustering. And for a completely random graph, both of these quantities fall. Therefore, usually large l is associated with large C and vice versa small l with small C. Here we see (see Fig. 1.2.6) that there is a large range of values, in which relatively large C and small l. This is the characteristic feature of the small world.

A detailed derivation of formula (1.2.41) based on the mean field theory approximation is described here: *"Mean – field solution of the small – world network model" – M. E. J. Newman, C. Moore and D. J. Watts, Phys. Rev. Lett. 84, 3201–3204 (2000) (arXiv : 9909165 v 2).*

1.2.4. Percolation networks

Let us briefly dwell on one more type of networks – percolation networks. In the simplest version, a percolation network is built from a regular, for example, square lattice by pulling out (destroying) randomly selected links. On fig. 1.2.7 the remaining bonds are indicated by a thick line torn out by a thin one. Let connections (nodes) break out when creating a percolation network with a probability of 1 – *p,* then it will

consist of $p \cdot N$ (N is an integer number of links (nodes)) of the so-called black links (nodes).

For each such a network (square, triangular, hexagonal, cubic,...) exists such meaning p_c (percolation threshold) that when $p > p_c$ it is possible to pass through the black links through the entire network, but $p < p_c$ not when.

On fig. 1.2.8 shows a large percolation network for two cases *а* - below the percolation threshold $(p < p_c)$ and *б* - above $(p > p_c)$.

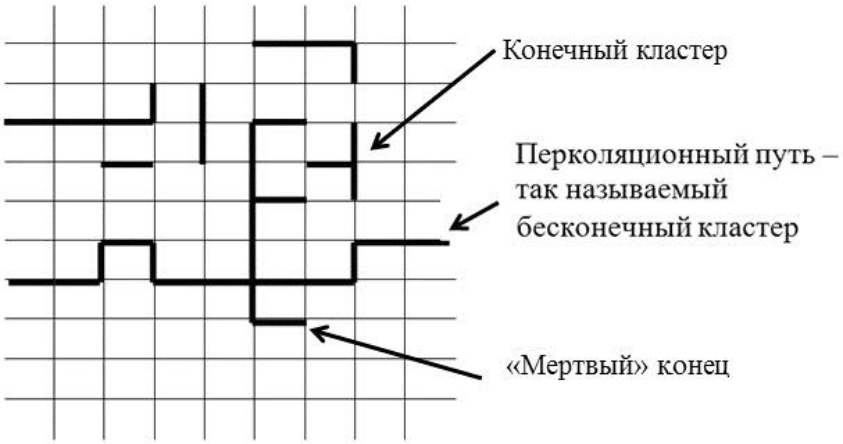


Figure. 1.2.7 – Square lattice with randomly torn bonds (thin lines)

For different lattices – triangular, square, cubic, etc. its percolation threshold p_c, the numerical value of which can (with rare exceptions) only by numerical simulation. Table 1.2.1 shows the values of the percolation threshold p_c for different gratings.

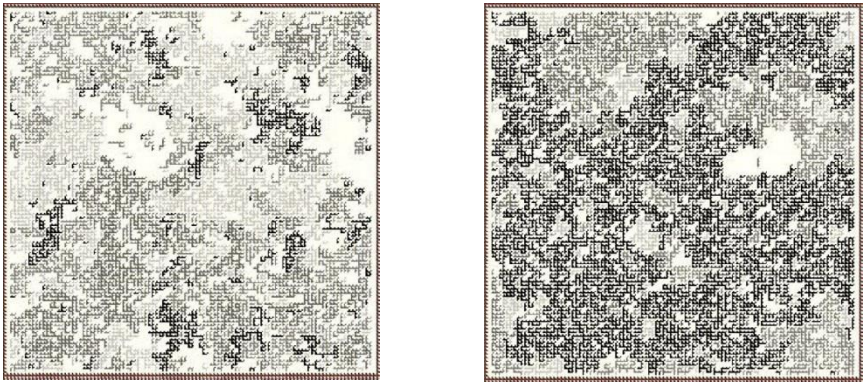


Figure. 1.2.8 – Large size percolation network. On the left, the case when the concentration of bonds is less than the threshold, on the right – more

Table 1.2.1. The numerical value of the percolation threshold p_c

Lattice	Node task	Communication task
Hexagonal	0.69	0.65
To the address	0.59	$\underline{\frac{1}{2}}$
triangular	$\underline{\frac{1}{2}}$	0.35
cubic	0.31	0.25
$d = 4$ hypercubic	0.197	0.16
$d = 5$ hypercubic	0.14	0.12
$d = 6$ hypercubic	0.11	0.09

Values are underlined p_c that are defined exactly. The first column is the task of the nodes, the second is the task of the links.

When calculating, p_c the size of the lattice is chosen to be large enough (in theory, infinite) so that the value p_c ceases to depend on the total number of nodes – N or connections.

The main thing in percolation is the formation of the so-called infinite cluster, which allows one to pass through the links through the entire network (they say from infinity to infinity, meaning in the limit the infinite size of the network). The number of links in an infinite cluster is of the order of all links in the network.

In this respect, the infinite cluster is similar to Giant Cluster in complex networks, such as ER. There is both common and different in the properties of an infinite cluster in percolation theory and in Giant Cluster in the theory of complex networks.

Giant for short Cluster is like GC an infinite cluster, as PC we note the following.

For percolation on the Kelly lattice $p_c = 1/(z-1)$, where z is the coordinate number, for GC ER the network the coordinate number from N the bonds is $N-1$, and for $N >> 1$ $p_c = 1/N$. Thus, with an increase N p_c decreases, which is similar to an increase in the spatial dimension of the percolation network. In this case, the theory of complex networks (ER) in the limit $N \to \infty$ is analogous to percolation in an infinite dimensional space.

Both in the problem about GC and in the problem about PC when $p < p_c$ the probability of occurrence of GC and PC is equal to zero.

Above the percolation threshold, at, $p > p_c$ the size GC is $\left(f(p_c N) - f(pN)\right)N$, where f is an exponentially decreasing function of c $f(1) = 1$, while the size PC is $(p - p_c)N$.

Another difference lies in the structure GC - it is trees, while PC it has a fractal structure.

Percolation networks are described in more detail in the second part of this tutorial.

1.3. Examples of real networks

The study of a significant number of complex artifact (artificially created) networks, some of which are described here, was initiated by the desire to understand and describe numerous real networks – from communication networks to ecological networks. Here are some of them:

1. World Wide Web : The number of websites is 919,533,715 (as of March 2014 according to the Netcraft service, Figure 1.3.1), covering over a trillion ($\approx 10^{12}$) web pages. One-way links between individual web pages are implemented as hyperlinks.
2. Internet – "Physical network" (Fig. 1.3.2, 1.3.3);
3. Protein networks (Fig. 1.3.4);
4. Metabolism network;
5. Ecological networks;
6. Phone calls network;
7. Terrorist network;
8. Citation network (acyclic);
9. Linguistic (a network of related words);
10. Neural networks.

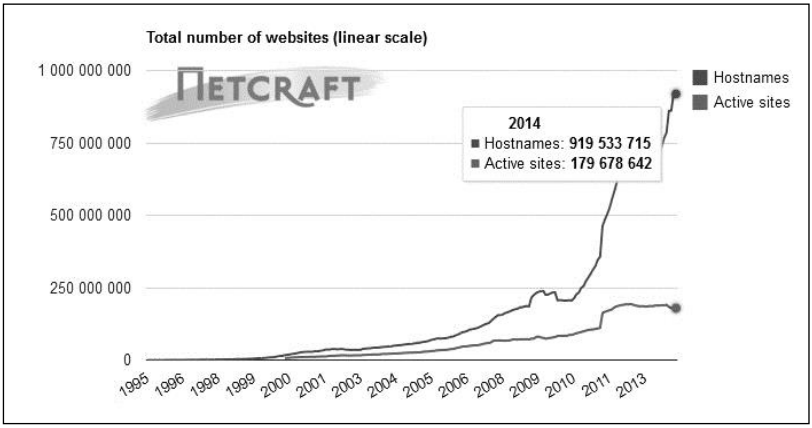


Figure. 1.3.1 – The dynamics of the development of the World Wide Web according to the Netcraft service (http://netcraft.com)

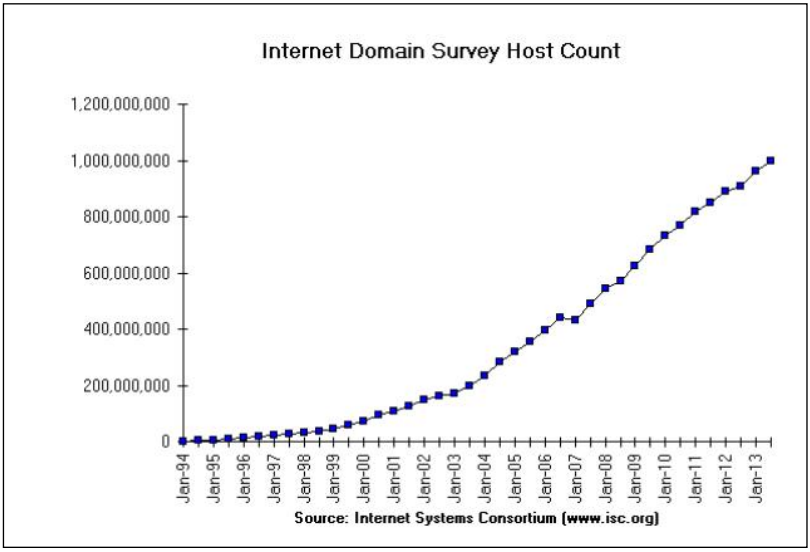


Figure. 1.3.2 – Dynamics of growth in the number of Internet domains according to the isc service.org

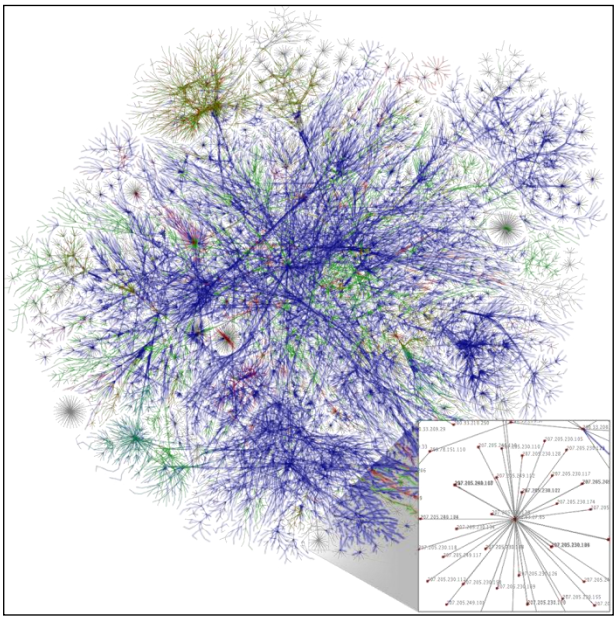


Figure. 1.3.3 – Link map of Internet servers as a complex network (according to wikipedia.org)

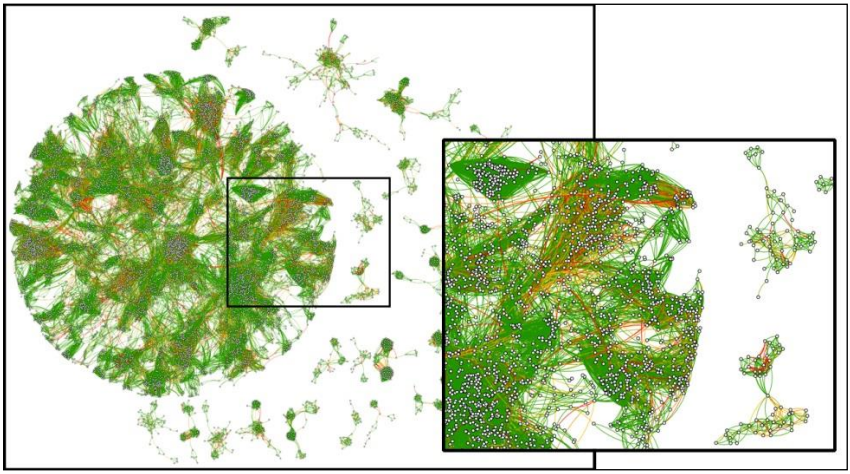


Figure 1.3.4 – Protein network (Protein Structure Initiative project, site www.helixscript.com)

2. Problems of searching in networks

2.1. Vector-spatial search model

Gerhard Solton

(*1927-1995)*

Most of the well-known information retrieval systems are based on the use of the Vector Space Model proposed by G. Salton in 1975 and applied by him in the SMART system. This model is classical algebraic. Within this model, a document is described by a vector in Euclidean space, in which each term used in the document is assigned its weight value, which is determined on the basis of statistical information about its appearance both in a separate document and in the entire document array. The description of the query corresponding to the topic required by the user is also a vector in the same Euclidean space of terms. The scalar product of the corresponding request and document vectors is used to estimate the proximity of the request and the document.

Within this model, each term t_i in the document d_j corresponds to some non-negative weight w_{ij}.

In this model, the query q, which is also a set of terms that are not interconnected by any logical operators, also corresponds to a vector of weight values w_{iq}.

Lande D.V., Snarskii A.A., Bezsudnov I.V. Internet: Navigation in complex networks: models and algorithms. – M.: Librokom (Editorial URSS), 2009. – 264 p.

Thus, each document and query can be represented as a $n-$dimensional vector, where is n – the total number of terms in the model dictionary. In accordance with the considered model, the proximity of the document d_j to the query q, which, as in the previous models, are considered as information vectors $d_j = (w_{1j},\ w_{2j},...,w_{nj})$ and $q = (w_{1q},\ w_{2q},...,w_{nq})$, are evaluated as their scalar product. In this case, the weight of

individual terms can be calculated in different ways. One of the possible simplest approaches is to use as the weight of a term w_{ij} in a document the normalized frequency $freq_{ij}$ of its occurrence in a given document, that is:

$$w_{ij} = tf_{ij} = freq_{ij} / \max_{1 \le i \le n}(freq_{ij}). \qquad (2.1.1)$$

However, this approach does not take into account how often a given term is used in the entire array of documents, the so-called discriminatory power of a term. Therefore, in the case where statistics on the use of terms in the entire document array is available, the following rule for calculating the weight is more efficient:

$$w_{ij} = tf_{ij} \cdot \log \frac{N}{n_i}, \qquad (2.1.2)$$

where is n_i – the number of documents in which the term is used t_j, and is N – the total number of documents in the array.

It should be noted that the above formula has been repeatedly refined in order to most closely match the documents issued by the systems to user requests. In 1988, Salton proposed this option for calculating the weight of a term t_i from a query:

$$w_{iq} = \left(0.5 + \frac{freq_{iq}}{\max\limits_{1 \le l \le n} freq_{lq}} \right) \cdot \log \frac{N}{n_i}, \qquad (2.1.3)$$

where $freq_{iq}$ is the frequency of a term t_i from a query in the text of a document consisting of n terms.

Typically, the weights w_{ij} are normalized, which allows the document to be treated as an orthonormal vector. This method of term weighting has a standard notation – $TF \cdot IDF$, where TF indicates the frequency of occurrence of the term in the document (term frequency), and IDF – by the reciprocal of the number of array documents containing the given term (inverse document frequency).

When the task of determining the thematic proximity of two documents or a document and a query arises, this model uses a simple scalar product $sim(d_1, d_2)$, of two corresponding vectors of weight values $(w_{i1}, ..., w_{n1})$ and $(w_{i2}, ..., w_{n2})$, which, obviously, corresponds to the cosine of the angle between the vectors – images of documents d_1 and d_2. Obviously, $sim(d_1, d_2)$ belongs to the range [0, 1]. The larger the value, $sim(d_1, d_2)$ – the closer the documents d_1 and d_2. For any document, d we have $sim(d,d) = 1$. Similarly, the measure of the proximity of the document d_j and the query q is the value:

$$sim(d_j, q) = \frac{d_j \cdot q}{|d_j| \cdot |q|} = \frac{\sum_{i=1}^{n} w_{ij} w_{iq}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2} \sqrt{\sum_{i=1}^{n} w_{iq}^2}}. \qquad (2.1.4)$$

The vector-spatial model of data representation provides systems built on its basis with the following capabilities:

- processing requests without restrictions on their length;
- ease of implementation of the search mode for similar documents (each document can be considered as a request);
- saving search results with the ability to perform a refinement search.

At the same time, the vector-spatial model does not provide for the implementation of queries that implement logical operations, which significantly limits its applicability. In addition, being the methodological basis of others, including network search models, the classical vector -spatial model is focused on searching for information arrays that do not have an explicit network structure.

2.2. Peer-to-Peer Search Models

Gurkin Yu.N. Semenov Yu.A. P2P file-sharing networks: basic principles, protocols, security. Networks and communication systems, 2006. – No. 11. – p. 62.

We will consider models of search in a network environment using the example of search in the so-called peer-to-peer networks. Peer-to-peer networks (Peer-to-peer, P2P – peer-to-peer) are computer networks based on the equality of participants. In such networks, there are no dedicated servers, and each node (peer) is both a client and a server. The phrase "peer-to-peer" was first used in 1984 by Parbawell Yohnuhuitsman when developing IBM's Advanced Peer to Peer Networking architecture.

P2P is a network protocol that provides the ability to create and operate a network of peer-to-peer nodes and their interaction. In many cases, P2Ps are overlay networks, using the existing transport protocols of the TCP/IP stack – TCP or UDP. It should be noted that, in practice, peer-to-peer networks consist of nodes, each of which interacts only with a certain subset of other network nodes (due to limited resources). To implement the P2P protocol, client programs are used that provide the functionality of both individual nodes and the entire peer-to-peer network.

Search procedures in peer-to-peer networks imply taking into account their diverse topology, often decentralized. Today there are no unified approaches to the organization of search procedures, so a wide variety of methods are used. It is thanks to peer-to-peer networks that many methods of searching in a network environment have been developed, a detailed description of which will be discussed below.

Quite often, peer-to-peer networks are supplemented by dedicated servers that carry organizational functions, such as authorization. In particular, library peer-to-peer networks are known that use dedicated servers that play the role of centers for authorization, hashing and replication of bibliographic data.

The centralized client/server architecture implies that the network depends on central nodes (servers) that provide the terminals (clients) connected to the network with the necessary services. In this architecture, a key role is assigned to the

servers that define the network, regardless of the presence of clients. Despite the fact that all nodes in P2P have the same status, their actual capabilities may differ significantly.

It is obvious that the growth in the number of clients in a client/server network leads to an increase in the load on the server part, as a result of which it may become overloaded.

A decentralized peer-to-peer network, on the other hand, becomes more productive as the number of nodes connected to it increases. Indeed, each node adds its resources (disk space and computing capabilities) to the P2P network, as a result, the total network resources increase.

Compared to the client / server architecture, P2P has such advantages as self-organization, fault tolerance in case of loss of communication with network nodes (high survivability), the ability to share resources without being tied to specific addresses, increase the speed of copying information through the use of several sources at once, wide bandwidth, flexible load balancing.

In addition to the advantages of peer-to-peer networks mentioned above, they also have a number of disadvantages.

The first group of disadvantages is associated with the complexity of managing such networks compared to client-server systems. You have to spend considerable effort on maintaining a stable level of their performance, data backup, anti-virus protection, protection against information noise and other malicious user actions.

The big problem is the legitimacy of the content transmitted in such P 2 P networks. The unsatisfactory solution of this problem has already led to the scandalous closure of many such networks (for example, Napster in July 2001). There are other problems that have a social nature. So in the Gnutella system, for example, 70% of users do not add any files to the network at all. More than half of the resources in this network are provided by one percent of users, i.e. the network is evolving towards a client-server architecture.

Another problem of P 2 P networks is related to the quality and reliability of the content provided. A serious problem is the falsification of files and the distribution of fake resources. Protecting a distributed network from hacker attacks, botnets, viruses and Trojan horses is a very difficult task. Often, information about the participants in P 2 P networks is stored in an open form, available for interception. Another problem is the possibility of falsifying node IDs.

The main task of information retrieval in peer-to-peer networks is to quickly and efficiently find the most relevant responses to a query transmitted from a node to the entire network. In this case, of course, the search is implemented without the participation of the central server, i.e. decentralized. In particular, with such a search organization, the problem of obtaining a high-quality result with a general decrease in network traffic is relevant.

Kalogeraki V., Gunopulos D., Zeinalipour-Yazti D. A Local Search Mechanism for Peer-to-Peer Networks. Proc. of CIKM'02, McLean VA, USA, 2002.

Zeinalipour-Yazti D. Information Retrieval in Peer-to-Peer Systems. M.Sc Thesis, Dept. of Computer Science, University of California – Riverside, June 2003.

Zeinalipour-Yazti D., Kalogeraki V., Gunopulos D. Information Retrieval in Peer-to-Peer Networks. IEEE CiSE Magazine, Special Issue on Web Engineering, 2004. – pp. 1-13.

Let us consider in detail such methods (algorithms) of decentralized search:

– resource search algorithm by keys;

– method of broad primary search (Breadth First Search);

– method of random wide primary search (Random Breadth First search);

– intelligent search engine (Intelligent Search mechanism);

– the method of "most results according to the past heuristic" (> RES);

– algorithm of “random walks” (Random walkers algorithm).

Algorithm for searching resources by keys

In most peer-to-peer networks focused on file sharing, two types of entities are used, to which appropriate identifiers (ID) are assigned: nodes and resources, characterized by keys (Key), i.e. the network can be represented by a two-dimensional matrix MN, where is M – the number of nodes, N – the number of resources. In this case, the search task is reduced to finding the ID of the node where the resource key is stored. On fig. Figure 2.2.1 shows the process of searching for a resource, launched from a node with ID 0.

In this case, a search for a resource with key 14 is started from the node with ID 0. The request goes through a certain route and reaches the node where the key 14 is located. Then the node with ID 14 sends to the node with ID 0 the addresses of all nodes that have the resource corresponding to the key 14.

Let's consider search algorithms in peer-to-peer networks, limiting ourselves to the main methods of searching by keywords.

Broad Primary Search Method

The method of broad primary search (Breadth First Search, BFS) ***widely*** used in real P2P file-sharing networks such as Gnutella (www.gnutella.com). The BFS method (Fig. 9) in the network P2P of dimension N is implemented as follows. The node q generates a request that is addressed to all neighbors (the closest nodes according to some criteria). When a node p receives a request, a lookup is performed in its local index. If a node r receives a request (Query) and processes it, then it generates a response message (QueryHit) to return the result.

The QueryHit message includes information about relevant documents that is delivered over the network to the requesting node (Figure 2.2.2).

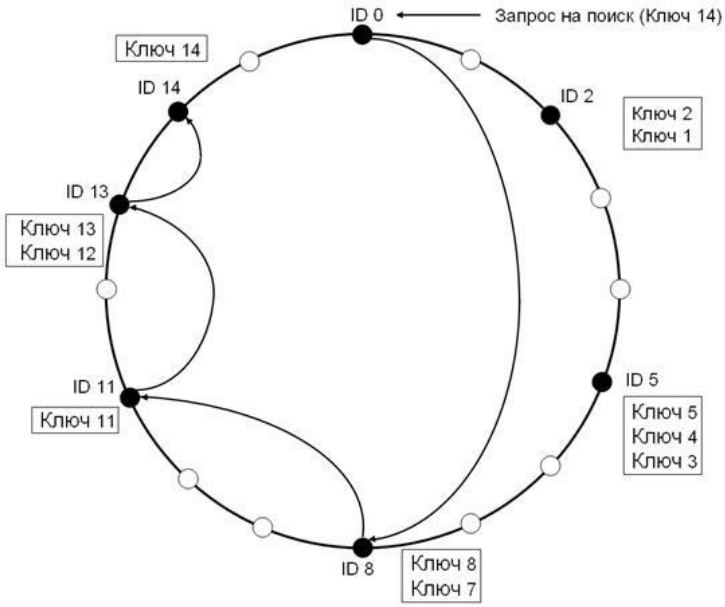


Figure 2.2.1 – Resource search model by key (black nodes contain documents with keys – white nodes do not contain)

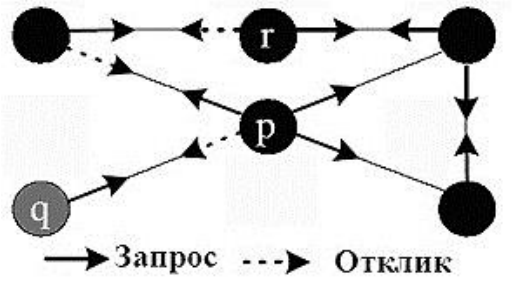


Figure 2.2.2 – BFS method

When a node *q* receives QueryHits from more than one node, it can download the file from the most available resource. QueryHit messages are returned in the same way as the original query. In BFS, each request causes excessive network load, as it

is transmitted over all links (including nodes with high latency). Therefore, a node with low bandwidth can become a bottleneck. One method to avoid overloading the entire network with messages is to assign each request with a time-to-live (Time – to – level, TTL) parameter. The TTL parameter specifies the maximum number of hops over which a request can be forwarded. In a typical search, the initial value for TTL is typically 5-7, which decreases each time the request is forwarded. When the TTL becomes 0, the message is no longer transmitted. BFS guarantees a high level of match quality through a large number of messages.

Random Wide Primary Search Method

Random wide primary search method (Random Breadth First Search, RBFS) has been proposed as an improvement on the "naive" BFS approach. In the RBFS method (Fig. 2.2.3), the node *q* forwards the search prescription only to a randomly selected part of the network nodes. Which part of the nodes is the parameter of the RBFS method.

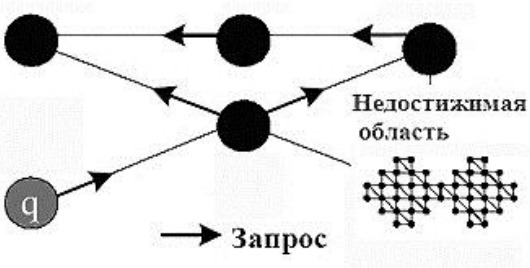


Figure 2.2.3 – RBFS method

The advantage of RBFS is that global information about the state of the network's content is not required; the node can get local solutions as fast as it needs to. On the other hand, this method is probabilistic. Therefore, some large network segments may not be reachable.

Intelligent search engine

Intelligent search engine (Intelligent Search mechanism, ISM) is a new search method in P2P networks (Fig. 2.2.4). Improving the speed and efficiency of information retrieval using this method is achieved by minimizing the cost of communications, that is, the number of messages transmitted

between nodes, and minimizing the number of nodes that are polled for each search request. To achieve this, for each query, only those nodes that best match the query are evaluated.

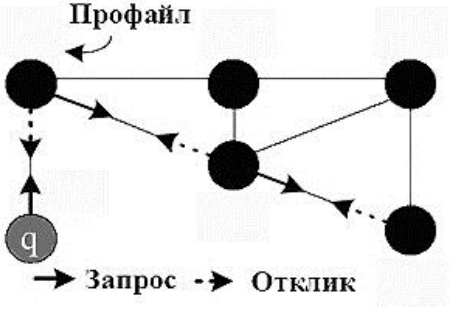


Figure 2.2.4 – ISM method

An intelligent search engine consists of two components – a profile (profile) and a way to rank it, the so-called relevance rank. Each network node builds an information profile for each of the neighboring nodes. The profile contains the latest responses from each node. Relevance Rank ranks node profiles to select those neighbors that will produce the most relevant documents for a query.

The profile mechanism is used to store the latest queries as well as the quantitative characteristics of the search results.

When implementing the ISM model, a single request stack of size is used $O(TN)$, which stores T requests per N node. As soon as the stack is full, the "last least used" (Least Recently Used, LRU) to save recent requests. Relevance Rank function (Relevance Rank, RR) is used by a node P_l to perform an online classification of its neighbors to determine which ones should be polled first on request q. To calculate the relevance rank of each node P_i, P_l compares q with all requests in the profile structure, for which the list of answers to previous requests is known, and calculates $RR(P_i, q)$:

$$RR(P_i, q) = \sum_{j \in Q} Sim(q_j, q)^{\alpha} \cdot S(P_i, q_j). \qquad (2.2.1)$$

In this formula Q, the set of requests to which the node had an answer P_i, $S(P_i, q_j)$ – the number of results returned by the node P_i for the query q_j,, and the metric Sim is calculated according to the cosine rule, similar to that considered in the vector-spatial search model:

$$Qsim(\boldsymbol{q}_j, \boldsymbol{q}) = \frac{\boldsymbol{q}_j \cdot \boldsymbol{q}}{|\boldsymbol{q}_j||\boldsymbol{q}|}. \qquad (2.2.2)$$

RR provides a higher rank node that returns more results. In addition, a parameter is used α, that allows you to increase the weight of requests that are most similar to the original one.

In the case where is α large, queries with high similarity $Qsim(q_j, q)$ dominate the above formula. Consider the situation where a node P_1 matches queries q_1 and q_2 similarity values for query q: $Qsim(q_1, q) = 0.5$ and $Qsim(q_2, q) = 0.1$, and a node P_2 matches queries q_3 and q_4 values $Qsim(q_3, q) = 0.4$ and $Qsim(q_4, q) = 0.3$. If you choose $\alpha = 10,$ then $Qsim(q_1, q)$ dominates, since $0.5^{10} + 0.1^{10} > 0.4^{10} + 0.3^{10}.$

However, for $\alpha = 1$ all queries weigh the same, and P_2 gives a higher relevance. When $\alpha = 0$ only counts the number of results returned by each node.

ISM method works effectively in networks where the nodes contain some specialized information. In particular, a study of the Gnutella network shows that the quality of the search is very dependent on the "environment" of the node from which the request is made. Another problem with the ISM method is that paging messages can loop and therefore fail to reach certain parts of the network. To solve this problem, the following approach was proposed. A small random subset of nodes was selected (an additional random node was selected in the experiment) and added to the set of relevant nodes for each query. As a result, the ISM mechanism began to cover a large part of the network.

Methods of "most results by past heuristics"

In the past heuristic majority (> RES) method (Figure 2.2.5), each node forwarded a query to a subset of its nodes based on some generalized statistics.

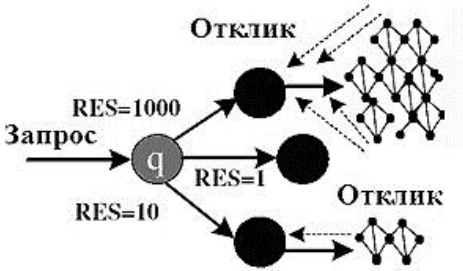


Figure 2.2.5 – Method > RES

A request in the > RES method is satisfactory if Z or more results are returned (Z is some constant). In the > RES method, the node q forwards requests to k the nodes that produced the highest results for the most recent m requests. In their experiments k it varied from 1 to 10 and in this way the > RES method varied from BFS to a Depth – first – search approach. The > RES method is similar to the ISM method discussed, but uses simpler node information. Its main drawback compared to ISM is the lack of analysis of the parameters of the nodes whose content is associated with the request. Therefore, the > RES method is characterized more as a quantitative rather than a qualitative approach. We know from experience that > RES is good in that it routes requests to large network segments (which may also contain more relevant responses). It also captures neighbors that are less congested, starting with those that typically return more results.

Method of "random walks"

The key idea of the random walk algorithm (Random walkers Algorithm, RWA) is that each node randomly forwards a request message, called "sending" to one of its nodes. To reduce the time needed to get results, the idea of a single "batch" is extended to " k -batch", where k is the number of independent

bursts sequentially launched from the search node. It is expected that "k -parcels" after T the steps will achieve the same results as one parcel per kT steps. This algorithm is similar to the RBFS method, but in RBFS each node forwards a part request message from its neighbors. In addition, RBFS assumes an exponential increase in transmitted messages, while the random walk method assumes a linear increase. Both RBFS and RWA methods do not use any explicit rules to direct the search query to the most relevant content.

Another technique similar to RWA is "adaptive probabilistic search" (Adaptive Probabilistic Search, APS). In APS, each node deploys a local index on its resources containing conditional probabilities for each neighbor that can be selected for the next hop for a future request. The main difference from RWA in this case is that in APS the node uses feedback from previous searches (in the form of conditional probabilities) instead of completely random transitions. Therefore, the APS method often gives better results than RWA.

Another algorithm developed at the University of California, based on the random walk method, uses the principle of the link percolation threshold, that is, the threshold for leaking or leaking links between closely connected nodes in the network. At the link percolation stage, the request goes to one of the base servers, which are connected to each other by powerful communication channels. It turned out that a full-fledged search process can be carried out "locally", that is, when polling only neighboring servers. With this approach, each request generates relatively little traffic.

There are many areas where P2P technology is successfully applied, for example, parallel programming, data caching, data backup.

Due to such characteristics as survivability, fault tolerance, and the ability to self-develop, peer-to-peer networks are increasingly used in production and organization management systems (for example, P2P technology is currently used in the US State Department). In this case, the possible failure of some nodes or servers does not significantly affect the manageability of the entire system. The Domain Name System (DNS) on the Internet is also effectively a peer-to-peer data exchange network.

The implementation of P2P technology is also the currently popular system of distributed computing GRID. Another example of distributed computing is the distributed project.net, whose members are engaged in the legal cracking of cryptographic ciphers in order to test their reliability.

2.3. Rank characteristics

Ranking is the process by which a search engine arranges search results in a certain order based on the principle of the greatest relevance to a particular query. Thus, the presentation of search results depends on the ranking algorithm that is used in the search engine.

As a result of the search, the user can get a large list of relevant documents. Sorting this list in such a way that the most important documents for the user are at the beginning, in information retrieval technologies, is commonly called the ranking of information retrieval system responses.

Ranking search results by relevance level is not possible for all search models (for example, it is impossible for a Boolean model).

A promising approach to ranking is the use of multi-profile scales formed on the basis of metadata, network properties, and user data.

For example, the implementation of story chains in thematic information arrays and their weighting are considered as one of the ranking algorithms. The ranking of text and hypertext documents has significant differences. Ranking of text documents can be carried out according to the level of relevance and other parameters, including those extracted from texts.

The ranking of hypertext documents is also possible according to the properties determined by the network structure, hyperlinks.

The Internet uses the analysis of the topology of the network formed by documents and the corresponding hyperlinks to determine the authority of a web page as a source of information or an intermediary. Two link-based web page ranking algorithms, HITS (hyperlink induced topic search) and PageRank, were developed in 1996 at IBM by J. Kleinberg [94] and at Stanford University by S. Brin and L. Page [74].

Both algorithms are designed to solve the "redundancy problem" inherent in wide queries, increasing the accuracy of search results based on complex network analysis methods.

Kleinberg JM Authoritative sources in a hyperlink environment. In Processing of ACM-SIAM Symposium on Discrete Algorithms, 1998, 46(5):604-632.

2.3.1.HITS algorithm

HITS (Hyperlink Induced Topic Search) algorithm proposed by J. _ Kleinberg, provides a selection from the information array of the best "authors" (primary sources to which links will be entered) and "intermediaries" (documents from which citations come). It is clear that a page is a good intermediary if it contains links to valuable primary sources, and vice versa, a page is a good author if it is mentioned by good intermediaries.

J. Kleinberg

For each document d_j ($j = 1,...,N$) recursively calculates its importance as an author $a(d_j)$ and an intermediary $h(d_j)$ according to the formulas:

Brin S., Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. WWW7, 1998.

$$a(d_j) = \sum_{i \to j}^{N} h(d_i), \quad h(d_j) = \sum_{j \to i}^{N} a(d_i). \tag{2.3.1}$$

If we introduce the concept of an incidence matrix $\boldsymbol{A}$, the element of which a_{ij} equals one when the document d_i contains a document reference d_j, and zero otherwise, then the HITS algorithm ensures that the most authoritative documents (authors or resellers) are selected. These documents correspond

to the eigenvectors of matrices AA^T and A^TA with the largest moduli of eigenvalues (here A^T, the transposed matrix A).

The algorithm for calculating the HITS ranks leads to an increase in the ranks of documents with an increase in the number and degree of relatedness of documents of the corresponding community. In this case, the results of issuing an information retrieval system using the HITS algorithm may include a large number of documents on topics other than the information needs of the user, i.e. some of the output results may deviate from the dominant topic, so-called topic drift occurs.

To solve this problem, the PHITS algorithm was proposed as an alternative to the standard HITS algorithm. Within the framework of this algorithm, it is assumed: $D-$a set of citing documents, $C-$a set of references, $Z-$a set of classes (documents that are close by some criterion) into which documents are divided. It is also assumed that the event $d \in D$ (that the document chosen at random is document d) occurs with probability $P(d)$.

Conditional probabilities $P(c|z)$ (probability that a reference from a class z is c) and $P(z|d)$ (the probability that the selected document d belongs to the class z) are used to describe the dependencies between the presence of the link $c \in C$, the factor $z \in Z$ and the document $d \in D$.

The likelihood function is estimated:

$$L(D,C) = \prod_{c\in C, d\in D} P(d,c) = \\ = \prod_{c\in C, d\in D} P(d)P(c|d), \qquad (2.3.2)$$

g de

$$P(c|d) = \sum_{z\in Z} P(c|z)P(z|d). \qquad (2.3.3)$$

The goal of the PHITS algorithm is to fit $P(z)$, $P(c|z)$, $P(d|z)$ to maximize $L(D,C)$.

After that:

$P(c|z)$ – ranks of authors;

$P(d|z)$ – ranks of intermediaries.

The likelihood function in this case shows how likely it is to have links in the selected documents.

To calculate the ranks, you must specify the number of factors in Z, and then $P(c|z)$ it will characterize the quality of the page as an author in the context of the topic z. The disadvantages of the method include the fact that the iterative process most often stops not at the absolute, but at the local maximum of the likelihood function L. However, in situations where there is no clear dominance of the query subject in the set of found web pages, PHITS outperforms the HITS algorithm.

2.3.2. PageRank Algorithm

PageRank algorithm was invented by the founders of Google to rank web pages. It was named after one of its inventors, Larry Page).

The main idea of this algorithm can be described in the following words: the significance (rank) of a page is the higher, the more links to it from other significant pages. Those. PageRank calculates the probability that a person randomly following links will reach a certain page. The more links pointing to a given page from other popular pages, the more likely the experimenter will stumble upon it purely by chance. In our terminology, a page is a network node, and a link to a page is a directed link.

PageRank algorithm is close in ideology to the literary citation index, which is calculated for an arbitrary document, taking into account the number of links from other documents to this document, but at the same time, in PageRank, as in HITS, unlike the literary citation index, not all links are considered equivalent.

PageRank of a web page is as follows: a model is considered – a process in which a certain Internet user opens a random web page, from which he clicks on a randomly selected hyperlink. Then it navigates to another web page and activates a random hyperlink again, and so on, constantly jumping from page to page, never coming back. Sometimes he gets bored with such wandering, and he again goes to a random web page – not by link, but by manually typing some URL. In this case, the probability that a user surfing the Web will go to some particular web page is its rank. Obviously, the PageRank of a web page is higher the more other pages link to it, and the more popular these pages are.

Authors PageRank

Larry Page

Sergey Brin

Let there be n pages $D=\{d_1, ..., d_n\}$, that link to this document (web page A), and $C(A)-$ the total number of links from a web page A to other documents. In accordance with the above model of user behavior, some fixed value is determined δ (damping factor) as the probability that the user, when viewing some web page from the set D, will go to the page A by a link, and not by explicitly typing its URL. Within the framework of the model, the probability of this user continuing to surf the web from N web pages without using hyperlinks, by manually entering the address (URL) from a random page, is $1-\delta$ (an alternative to following hyperlinks). PageRank Index $PR(A)$ for a page A is considered as the probability that the user will be on this page at some random time:

$$PR(A)=(1-\delta)/N+\delta\sum_{i=1}^{n}\frac{PR(d_i)}{C(d_i)}. \qquad (2.3.4)$$

With this formula, the page index is easily calculated by a simple iterative algorithm. In practice, up to 30 iteration steps are used to achieve sustainable results.

Despite the differences between HITS and PageRank, these algorithms have in common that the authority (weight) of a node depends on the weight of other nodes, and the level of the "intermediary" depends on how authoritative the nodes to which it refers.

The calculation of the authority of individual documents is now widely used in applications such as determining the order in which documents are scanned on the web by search engine robots, ranking search results, generating thematic overviews, and etc.

Let's illustrate the above with an example. Consider a small part of the network consisting of nodes A with rank $r_A = 0.5$, B with rank $r_B = 0.3$ and determine the rank of node C – see Figure 2.3.1.

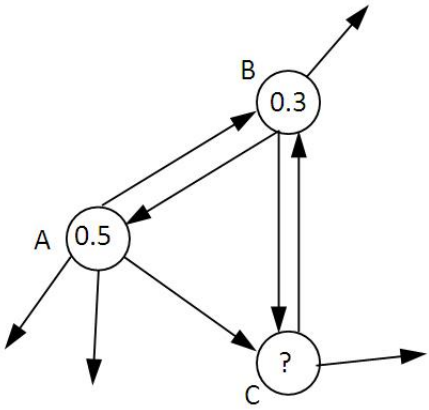


Figure. 2.3.1 – Network fragment

Each of the network nodes *A* and *B* refers to the node *C* and for them the rank is already known. Node A refers to three more nodes, and node B to two more nodes. To calculate the rank of node *C,* the ranks of each node that links to *C* are divided by the total number of links for that node, after which the resulting values are added.

$$r_C = r_A \cdot \frac{1}{4} + r_B \cdot \frac{1}{3} = 0.225. \qquad (2.3.5)$$

In this example, for all nodes that point to *C,* the rank has already been calculated. But it is impossible to calculate the rank of a node until the ranks of the nodes that link to it are known, and these ranks can only be calculated by knowing the ranks of the nodes that link to them. So how do you calculate the rank value for a set of nodes whose ranks are not yet known? The solution is to give all nodes an arbitrary initial rank and run several iterations.

In general, we can write the following formula for $(n+1)$ the -th iteration step:

$$r_i^{(n+1)} = \sum_{j \in E(i)} \left(r_j^{(n)} \cdot \frac{1}{kout_j} \right), \qquad (2.3.6)$$

where is the summation over $j \in E(i)$ means the summation over all nodes that have a link to i the -th node, and $kout_j$ is the number of outgoing links for node j. In matrix form (2.3.6) is written as follows:

$$\mathbf{r}^{(n+1)T} = \mathbf{r}^{(n)T} \cdot \mathbf{H}, \qquad (2.3.7)$$

Where $H_{ij} = A_{ij} \Big/ \sum_j A_{ij}$ is the normalized incidence matrix and $\mathbf{A}$ is the network incidence matrix.

For the iterative process (2.3.7), a number of questions arise, namely:

- Does this process converge?
- What properties must the matrix have $\mathbf{H}$ in order for the process to converge?
- Does the final vector depend $\boldsymbol{r}^{(\infty)}$ on the initial conditions?

Let's look at two simple examples. First:

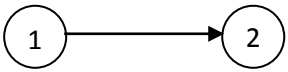


Figure. 2.3.2 – Example 1

After iterations (2.3.7) we get:

Iteration Node Rank	0	1	2	3
1	1	0	0	0
2	0	1	0	0

The result is clearly not as expected.

Let's look at another similar example.

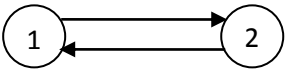


Figure. 2.3.3 – Example 2

Here we get the following table:

Iteration Node rank	0	1	2	3
1	1	0	1	0
2	0	1	0	1

From the above table it can be seen that the convergence of the process is not observed. Many more counterexamples can be given, showing the limitations of formula (2.3.8). *On* the other hand, one can see that (2.3.8) resembles " power method " to calculate the matrix eigenvector $\mathbf{H}$ (corresponding to the eigenvalue – one) applied to Markov chains with transition probability matrix $\mathbf{P} = \mathbf{H}$. In this case, we are talking about the "left" eigenvector. And since the theory of Markov chains is very well studied, we can immediately answer the question of what properties the transition probability matrix should satisfy $\mathbf{P}$ so that the process converges, does not depend on the initial conditions, etc.

The transition probability matrix must be stochastic, irreducible and non-periodic. The first condition is satisfied by passing to the matrix $\mathbf{S}$.

$$\mathbf{S} = \mathbf{H} + \mathbf{a} \cdot \left(\frac{1}{n} \cdot \mathbf{e}^T \right), \qquad (2.3.8)$$

where $a_i = 1$ if i no link comes out of the -th node (the so-called " dangling node ") and is equal to zero in the other case, $\mathbf{e}$ is a vector consisting of n units, n is the number of nodes in the network. And to satisfy the remaining two conditions, we write the matrix $\mathbf{G}$:

$$\mathbf{G} = \alpha \cdot \mathbf{S} + (1-\alpha) \cdot \frac{1}{n} \cdot \left(\mathbf{e} \cdot \mathbf{e}^T \right) =$$
$$= \alpha \cdot \mathbf{H} + \left(\alpha \cdot \mathbf{a} + (1-\alpha) \cdot \mathbf{e} \right) \cdot \frac{1}{n} \cdot \mathbf{e}^T, \qquad (2.3.9)$$

where is α the so-called damping factor. It means that the user will continue to follow the links available on the current page with a probability α, which is in the range from zero to one. Usually accepted $\alpha = 0.85$. So, our task is reduced to calculating the left eigenvector of the Google matrix $\mathbf{G}$.

Consider the first counterexample. For him:

$$\mathbf{H} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \ \mathbf{a} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \ \mathbf{e} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \qquad (2.3.10)$$

From (2.3.10) we obtain a matrix $\mathbf{G}$ (for definiteness, we set $\alpha = 0.85$).

$$\mathbf{G} = \begin{pmatrix} 0.075 & 0.925 \\ 0.5 & 0.5 \end{pmatrix}. \qquad (2.3.11)$$

Find the left eigenvalues of the matrix $\mathbf{G}$.

$$\mathbf{r}^T = \begin{pmatrix} 0.351 & 0.649 \end{pmatrix}. \qquad (2.3.12)$$

It can be seen that the second node is more significant than the first, which is fully consistent with the intuitive idea. Because the first node refers to the second one (see Figure 2.3.2). In exactly the same way, for the second counterexample we obtain the following ranks:

$$\mathbf{r}^T = \begin{pmatrix} 05 & 0.5 \end{pmatrix}. \qquad (2.3.13)$$

As it should be (see Fig. 2.3.3), we got equivalent values for the ranks.

Consider a more complex example of the network shown in Fig. 2.3.4.

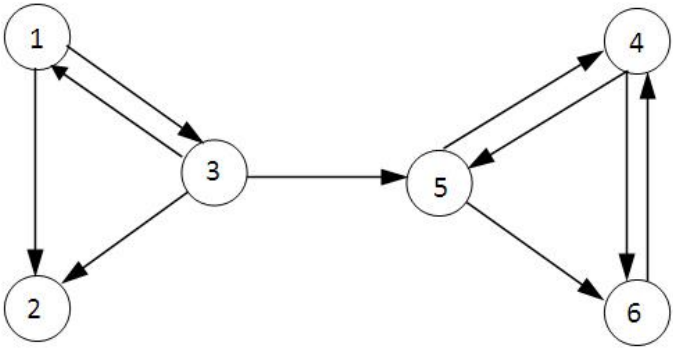


Figure. 2.3.4 – Network example

Let's write a matrix $\mathbf{H}$ and a vector $\mathbf{a}$ for this network.

$$\mathbf{H} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \qquad (2.3.14)$$

$$\mathbf{a}^T = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}. \qquad (2.3.15)$$

Let's find the matrix $\mathbf{G}$ by taking $\alpha = 0.85$:

$$\mathbf{G} = \begin{pmatrix} 0.025 & 0.45 & 0.45 & 0.025 & 0.025 & 0.025 \\ 0.167 & 0.167 & 0.167 & 0.167 & 0.167 & 0.167 \\ 0.308 & 0.308 & 0.025 & 0.025 & 0.308 & 0.025 \\ 0.025 & 0.025 & 0.025 & 0.025 & 0.45 & 0.45 \\ 0.025 & 0.025 & 0.025 & 0.45 & 0.025 & 0.45 \\ 0.025 & 0.025 & 0.025 & 0.875 & 0.025 & 0.025 \end{pmatrix} \qquad (2.3.16)$$

.

Find the left eigenvector (one for the eigenvalue):

$$\mathbf{r}^T = \begin{pmatrix} 0.052 & 0.074 & 0.057 & 0.349 & 0.2 & 0.269 \end{pmatrix} \qquad (2.3.17)$$

.

For clarity, let's renumber the nodes in accordance with their ranks – the first number will be assigned to the node with the highest rank – see fig. 2.3.5.

It can be noted that without calculations (intuitively), even such a simple network is almost impossible to rank.

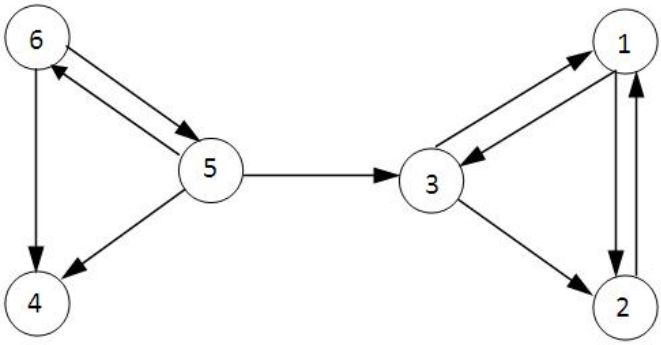


Figure. 2.3.5 – Network with ranked nodes

2.3.3. Salsa algorithm

Algorithm ranking Salsa (Stochastic Approach for Link-Structure Analysis – Stochastic Algorithm Analysis structures ties) was suggested Sh. Moran and R. Lempel as a kind of symbiosis of the PageRank and HITS algorithms, which makes it possible to reduce the consequences of the formation of the so-called TKC (Tightly-Knit Community) – closely connected communities. This effect consists in the presence in the issuance of the search system of many documents that are closely related to each other, the subject matter of which is somewhat different from the information needs of the user, i.e. there is a “topic draft” effect when displaying results, some of which may deviate from the dominant topic.

Lempel R. and Moran S. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. In Proc. of the 9th International WWW Conference, Amsterdam, The Netherlands, 2000. – pp. 387–401.

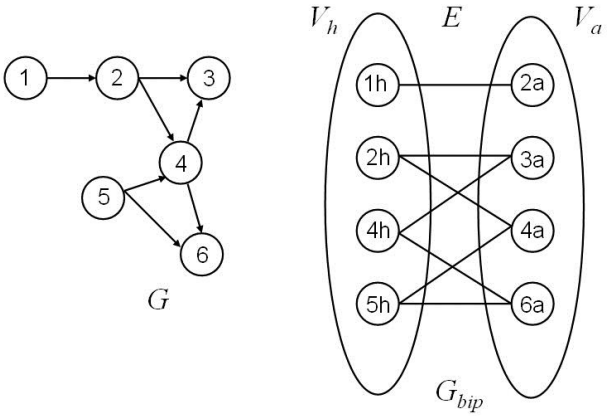
As in the PageRank method, in the case of Salsa, a random walk model of the user across the network (web graph) is assumed, but the presence of two-way "surfing" is assumed. According to the Salsa algorithm, a simple two-step procedure is performed a predetermined number of times:

1. From an arbitrary node v, the user randomly returns to the node u, randomly selected from the set of nodes referring to

v. The choice v is made randomly (the nodes v and u belong to the network).

A transition is made at random w from the node to the node u if there is a connection $(u,\ w)$.

A web graph G (Fig. 2.3.6 *a)* can be converted into a bipartite undirected graph G_{bip}, (Fig. 2.3.6 *b)* and defined as a set $G_{bip} = (V_h,\ V_a,\ E)$, where h denotes intermediaries, V_h - a set of intermediary nodes (those from which links come), a – authors, V_a – set of author nodes (those to which links lead).



a b

Figure 2.3.6 – Salsa : bipartite graph construction

It should be noted that the same nodes can be both authors and intermediaries at the same time.

Each non-isolated node $s \in G$ is represented by G_{bip} one or two nodes s_h and s_a. In this bipartite graph, Salsa implements two different random transitions. With each transition, it is possible to "visit" nodes from only one of the two parts of the graph G_{bip}.

Each path of length two in G_{bip} represents a bypass of one hyperlink (when passing from the share of intermediaries to the share of authors in G_{bip}), and a retreat along the hyperlink (when passing in the opposite direction). This movement in the opposite direction resembles the salsa dance, which is associated with the name of this algorithm.

Since topic-related proxies and authors t must be explicit in G_{bip} (accessible from many nodes via direct links or shortcuts), it is assumed that V_a topic-related proxies t and proxies from V_h will be the most frequently visited on random "wanderings" of users.

Salsa algorithm explores two different Markov chains that are associated with these random walks: the author-side chain G_{bip} (author chain), and the broker-side chain G_{bip} .

This approach allows us to introduce two stochastic transition matrices for Markov chains, which are defined as follows: the incidence matrix of the W directed graph is constructed G. Denote as W_r the matrix obtained by dividing each nonzero element W by the sum of the values of the corresponding row, and denote by W_c the matrix obtained by dividing each nonzero element W by the sum of the elements in the corresponding column. Then, the matrix H corresponding to the mediators will consist of non-zero rows and columns $W_r W_c^T$, and the matrix of authors A, respectively, will consist of non-zero rows and columns $W_c^T W_r$. Within the framework of the Salsa algorithm, the rows and columns of matrices and which consist entirely of zeros H, are ignored A, since, by definition, all nodes G_{bip} have at least one connection. As a result, matrices A and H are used to calculate ranks in the same way as in the HITS algorithm.

Converging during the iterative process, the probability of transition to the node v as the author has a very simple form:

$$\pi_v = c_1 \cdot InDegree(v), \qquad (2.3.18)$$

and the probability of returning to the node u as an intermediary:

$$\pi_u = c_2 \cdot OutDegree(u), \qquad (2.3.19)$$

where c_1 and c_2 are some constants, and $InDegree$ and $OutDegree$ are the number of outgoing and incoming links, respectively.

R. Lempel and S. Moran demonstrated that the Salsa algorithm is less sensitive to the effect of closely related communities than HITS, but under the condition that links that are not related to the topic under study are manually removed from the documents. This requirement in practice leads to high costs, as a result of which the authors are not yet aware of cases of using this ranking algorithm in real systems.

Many of the real complex networks are content, i.e. those whose nodes store text documents (web pages, blog posts, official documents, etc.). Therefore, let us dwell in detail on the methods of grouping, namely, on the problems of their classification and clustering. These methods allow you to combine various network nodes into categories, which significantly affects the reduction in the number of different meaningful objects in networks. In what follows, we will refer to all objects as documents, and the object parameters as terms, which does not limit the generality of subsequent conclusions.

2.4. Classification

2.4.1. Formal description of the classification

Let $D = \{d_1, ..., d_{|D|}\}$ - a set of objects (network nodes or, for example, their content elements – documents), $C = \{c_1, ..., c_{|C|}\}$ - a set of categories, Φ - an objective function that $< d_i, c_j >$ determines by a pair whether a document belongs d_i to a category c_j (1 or True) or not (0 or False). The task of classification is to construct a function Φ' that is as close as possible to Φ.

Machine learning methods that are used for classification

provide for a collection of objects previously classified by experts, i.e. those for which the value of the objective function is already known exactly. In order to evaluate its efficiency after building the classifier, this collection is divided into two parts, not necessarily of equal size:

1. Training (training-and-validation, TV) collection. The classifier Φ'is built on the basis of the characteristics of these objects.

2. Test (test, Te) collection. It checks the quality of the classification. Objects from Te should not be used in the process of building a classifier.

The classification under consideration is called clear binary, i.e. it is assumed that there are only two categories that do not intersect. Many tasks are reduced to such a classification, for example, classification according to a set of categories $C = \{c_1, ..., c_{|C|}\}$ is divided into $|c|$ binary classifications by sets $\{c_i, \bar{c}_i\}$.

Ranking is often used, in which the set of objective function values is the segment [0, 1]. When ranking an object, it can belong not only to one, but to several categories at once with different degrees of membership, i.e. categories may overlap.

2.4.2. Ranking and clear classification

Suppose that for each category c_i a function is built CSV_i .

Consider the problem of moving from a ranking function to an accurate classification. The easiest way is c_i to select a limit value (threshold) for each category τ_i. If $CSV_i(d) > \tau_i$, then the document d matches the category c_i. Another approach: for each document, d choose k the closest categories, i.e. k categories on which $CSV_i(d)$ they take the highest values.

To select a threshold value:

- proportional method. The study collection is divided into two parts. For each category c_i on one part of the training collection, it is calculated which part of the documents belongs to it. The threshold values are chosen so that on

the other part of the training collection the number of remaining documents assigned to c_i is the same.

- Method of k nearest categories. Each document d_i is considered to belong to *the* k closest categories and a threshold value is selected accordingly.

2.4.3. A measure of the proximity of an object and category

In this method, the classifier rule is the dot product. Let each category C_i correspond to a vector $\boldsymbol{C_i} = (c_{i1}, \ldots, c_{iN})$, where N is the dimension of the space of terms. The following formula is used as a classifier rule:

CSV (*Categorization Status Value* – classification status) – a function that maps a set of documents D to the segment [0; 1], which specifies the degree of belonging of the document to the category.

$$CSV_i(d) = \boldsymbol{d} \cdot \boldsymbol{C}_i = \sum_{j=1}^{N} c_{ij} d_j. \qquad (2.4.1)$$

Normalization is usually carried out in such a way that the final formula for $CSV_i(d)$ is a normalized scalar product – the cosine of the angle between the category vector c_i and a vector of weight values of the terms included in the document d - $\boldsymbol{d} = (d_1, \ldots, d_N)$:

$$CSV_i(d) = \frac{\boldsymbol{d} \cdot \boldsymbol{C_i}}{|\boldsymbol{d}| \cdot |\boldsymbol{C_i}|}. \qquad (2.4.2)$$

Vector coordinates C_i are determined in the course of training, which is carried out for each category independently of the others.

2.4.4. Rocchio Method

Profile (profile) – a prototype of a document, category or array of documents, most often a set of weighted terms

Some classifiers use a so-called profile to define a category. A profile is a list of weighted terms, the presence or absence of which makes it possible to most accurately distinguish a particular category from other categories. These classification methods include the Rocchio method, which refers to linear classifiers in which each document is represented as a vector of weighted terms. The category profile i will be considered as a vector $\boldsymbol{C_i} = (c_{1i}, ..., c_{Ni})$ (N is the number of terms in the dictionary), the values of the elements of which, c_{ki} within the Rocchio method, are calculated by the formula:

$$c_{ki} = \frac{\alpha}{|POS_i|} \cdot \sum_{d_j \in POS_i} w_{kj} - \frac{\beta}{|NEG_i|} \cdot \sum_{d_j \in NEG_i} w_{kj}, \qquad (2.4.3)$$

where w_{kj} is the weight of the term t_k in the document d_j (calculated, for example, according to the *TF IDF principle)*, $POS_i = \{d_j \mid \Phi(d_j, c_i) = 1\}$ and $NEG_i = \{d_j \mid \Phi(d_j, c_i) = 0\}$. In this formula, α and β are control parameters that characterize the significance of positive and negative examples. For example, if α = 1 and β = 0, C_i will be the center of mass of all documents belonging to the corresponding category.

The function $CSV_i(d)$ is defined either as the reciprocal of the distance from the vector of the weight values of the terms included in the document d to the category profile i – C_i, or as the scalar product of these vectors. The Rocchio method gives satisfactory results when documents from the same category are close to each other in distance.

2.4.5. Linear regression method

Regression analysis is used when the features of categories can be expressed quantitatively as some combination of vectors of weight values of terms included in documents from the

training collection. The resulting combination can be used to determine the category to which the new document will belong. In the simplest case, standard statistical methods such as linear regression are used to solve this problem.

The regression method is a variant of linear classification trained on the entire collection at once. When applying regression analysis to text classification, a set of terms (F) and set of categories (C). In this case, the training collection of documents is associated with two matrices:

- the matrix of documents $\boldsymbol{D}$ in the training collection, where each row is a document and each column is – term, number of lines N – number of documents in the training collection;
- response matrix $\boldsymbol{O} = \left\| o_{i,j} \right\|_{|}$, in which the row i corresponds to the document ($i = 1, ..., N$), the column j corresponds to the category ($j = 1, ..., K$), and $o_{i,j}$ to the value $CSV_j(d_i)$.

The regression method is based on the algorithm for finding the matrix of rules $\boldsymbol{M}$, which minimizes the value of the norm of the matrix $\left\| \boldsymbol{MD - O} \right\|_F$, that is:

$$M = \arg \min_M \left\| \boldsymbol{MD - O} \right\|_F . \qquad (2.4.4)$$

Recall that in linear algebra, the norm of a matrix is a function that assigns a numerical characteristic to a matrix. The matrix norm reflects the order of magnitude of the matrix elements. In this case, it is recommended to use the Frobenius norm $\left\| \cdot \right\|_F$, equal to the square root of the sum of the squares of all elements of the corresponding matrix:

$$\left\| A \right\|_F = \sqrt{\sum_{i,j} a_{ij}^{\,2}}. \qquad (2.4.5)$$

The element m_{ij} of the required matrix $\boldsymbol{M}$ will reflect the degree of membership i of the -th term j of the -th category.

2.4.6. DNF classifier

The DNF classifier consists of a set of rules, the conditions of which are specified by some DNF formula (DNF – disjunctive normal form), which is a disjunction of several expressions, the elements of which are connected by a certain number (possibly zero) of conjunctions. In this case, a document is categorized if it satisfies this formula, i.e. satisfies at least one term of the disjunction.

At the initial stage c_i, for each category that consists of documents, $\{d_1^i, ..., d_{|c_i|}^i\}$, the following formula is determined:

$$\text{IF}\,(x = d_1^i)\ \text{OR}\,(x = d_2^i)\ \text{OR...}\ \text{OR}\ (x = d_{|c_i|}^i),\ \text{THEN}\ c_i.$$

A classifier based on such a set of formulas works absolutely correctly on the training collection, but, firstly, it cannot work on other documents, and secondly, it is inconvenient to use such a classifier due to the large number of rules. In actually working DNF classifiers, there is a transition from documents to sets of terms, which are determined based on the analysis of the content of documents belonging to one category or another. In addition, a number of simplifications are carried out related to the combination or removal of certain conditions. Let's take a small example:

IF ((coffee & espresso) OR
(coffee & milk) OR
(tea & glass & lemon) OR
(coffee & cup & ¬grains))
TO Drink
AKA ¬Drink

Such actions improve the classification completeness index, but the accuracy can be significantly affected even on the training collection.

2.4.7. Bayesian logistic regression

In the Bayesian logistic regression model, the conditional probability of a document belonging D to the class is considered

$C : p(C \mid D)$.

It is assumed that the document is determined by the terms included in it, i.e. within the framework of this model, the document is a vector: $D = (w_1, \ ..., \ w_N)$, where w_i is the weight of the term i, and N is the size of the dictionary.

The Bayesian logistic regression model is given by the formula:

$$p(C \mid D) = \varphi(\beta \cdot D) = \varphi(\sum_{i=1}^{N} \beta_i \cdot w_i), \qquad (2.4.6)$$

where $C \in \{0,1\}$, $\beta = \{\beta_1, \ ..., \ \beta_N\}$ is the vector of model parameters, and φ is the logistic function, which is recommended to use:

$$\varphi(x) = \frac{1}{1 + \exp(-x)}. \qquad (2.4.7)$$

The β_i main idea of the approach is to use the previous distribution of the parameter vector, β in which each specific value β_i can take a value close to 0 with a high probability. β_i independent.

2.4.8. Naive Bayes Model

Consideration is given to the conditional probability that an object belongs to a class C, given that it has the attributes $F_1, \ ..., \ F_n$:

$$p(C \mid F_1, \ ..., \ F_n). \qquad (2.4.8)$$

According to Bayes' theorem:

$$p(C \mid F_1, \ ..., \ F_n) = \frac{p(C)\, p(F_1, \ ..., \ F_n \mid C)}{p(F_1, \ ..., \ F_n)}. \qquad (2.4.9)$$

By definition of conditional probability:

$$p(C \mid F_1,\ ...,\ F_n) = p(C)\,p(F_1,\ ...,\ F_n \mid C) =$$

$$= p(c)\,p(F_1 \mid C)\,p(F_2,\ ...,\ F_n \mid C, F_1) = \qquad (2.4.10)$$

$$= p(c)\,p(F_1 \mid C)\,p(F_2 \mid C)\,p(F_3,\ ...,\ F_n \mid C, F_1, F_2).$$

In accordance with the "naive" Bayesian approach, it is assumed that the events F_i, F_j are independent for any $i \neq j$:

$$p(F_i \mid C,\ F_j) = p(F_i \mid C). \qquad (2.4.11)$$

Respectively:

$$p(C \mid F_1,\ ...,\ F_n) =$$

$$= p(C)\,p(F_1 \mid C)\,p(F_2 \mid C)\cdot ...\cdot (F_n \mid C) = \qquad (2.4.12)$$

$$= p(C)\prod_{i=1}^{n} p(F_i \mid C).$$

Let's move on to the classification of documents. In the case of binary classification, the “naive” Bayesian probability of a document belonging to a class is determined by the formula:

$$p(D \mid C) = \prod_i p(w_i \mid C). \qquad (2.4.13)$$

According to Bayes' theorem:

$$p(C \mid D) = \frac{p(C)}{p(D)}\,p(D \mid C). \qquad (2.4.14)$$

Let's say classification occurs only in two classes – C and $\bar{C}$. Then, in accordance with the Bayes formula, we have:

$$p(C \mid D) = \frac{p(C)}{p(D)}\prod_i p(w_i \mid C); \qquad (2.4.15)$$

$$p(\bar{C} \mid D) = \frac{p(\bar{C})}{p(D)}\prod_i p(w_i \mid \bar{C}). \qquad (2.4.16)$$

The following ratio of probabilities is considered as a criterion for a document to belong to a category:

$$\frac{p(C \mid D)}{p(\overline{C} \mid D)} = \frac{p(C)}{p(\overline{C})} \prod_i \frac{p(w_i \mid C)}{p(w_i \mid \overline{C})}. \qquad (2.4.17)$$

In practice, the logarithm of the ratio of probabilities is used:

$$\ln \frac{p(C \mid D)}{p(\overline{C} \mid D)} = \ln \frac{p(C)}{p(\overline{C})} + \sum_i \ln \frac{p(w_i \mid C)}{p(w_i \mid \overline{C})}. \qquad (2.4.18)$$

If the inequality is true, $\ln \frac{p(C \mid D)}{p(\overline{C} \mid D)} > 0$, then the document is considered D to belong to the category C.

2.4.9. Support vector machine

Support Vector Machine (Support vector Machine, SVM), proposed by V.N. Vapnik belongs to the group of boundary classification methods. It determines the belonging of objects to classes using the boundaries of areas.

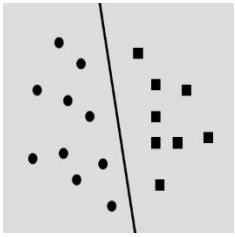
Considered, i.e. only over two categories C and $\overline{C}$ (it is taken into account that this approach can be extended to any finite number of categories). In addition, it is assumed that each classification object is a vector in N-dimensional space. Each coordinate of the vector is a certain feature, quantitatively the greater, the more this feature is expressed in the given object.

V.N. Vapnik

SVM – Support Vector Machine

Vapnik V.N. Statistical Learning Theory. – NY: John Wiley, 1998.

It is assumed that there is a training collection – it is a set of vectors $\{x_1, ..., x_n\} \in R^N$ and numbers. $\{y_1, ..., y_n\} \in \{-1, 1\}$. The number y_i is equal to 1 if the corresponding vector belongs to x_i the category c, and – 1 – otherwise.



Class separation direct

As shown above, a linear classifier is one of the simplest ways to solve a classification problem. In this case, a straight line (hyperplane in N – dimensional space), separating all points of one class from points of another class. If it is possible to find such a line, then the classification problem is reduced to determining the relative position of the point and the line: if a new point lies on one side of the line (hyperplane), then it belongs to the class, if on the other, c to the class $\overline{c}$.

Let's formalize this classification: it is necessary to find a vector $\boldsymbol{w}$ such that for some limit value b and a new point x_i the following holds:

$$y_i = \begin{cases} +1, \ \textit{если } w \cdot x_i \geq b, \\ -1, \ \textit{если } w \cdot x_i < b, \end{cases} \qquad (2.4.19)$$

where $\boldsymbol{w} \cdot \boldsymbol{x}_i$ is the scalar product of vectors $\boldsymbol{w}$ and $\boldsymbol{x}_i$:

$$\boldsymbol{w} \cdot \boldsymbol{x}_i = \sum_{j=1}^{N} w_j x_{i,j}. \qquad (2.4.20)$$

The equation $\boldsymbol{w} \cdot \boldsymbol{x}_i = b$ describes a hyperplane that separates the classes. That is, if the scalar product of the vector w is x_i not less than the value b, then the new point belongs to the first class, if less, to the second. It is known that the vector $\boldsymbol{w}$ is perpendicular to the desired dividing line, and the value b depends on the shortest distance between the dividing line and the origin. Obviously, if there is one dividing line, then it is not unique. The question arises, which of the lines separates the classes best?

The SVM method is based on the following postulate: the best separating line is the one that is as far as possible from the

nearest points of both classes to it. That is, the task of SVM is to find a vector w and a number such b that for some $\varepsilon > 0$ (half the width separating surface) was performed:

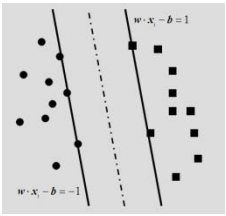
$$\begin{cases} \boldsymbol{w} \cdot \boldsymbol{x}_i \geq b + \varepsilon \;\Rightarrow\; y_i = +1, \\ \boldsymbol{w} \cdot \boldsymbol{x}_i \leq b - \varepsilon \;\Rightarrow\; y_i = -1. \end{cases} \quad (2.4.21)$$

We then multiply both sides of the inequality by $1/\varepsilon$ and, without loss of generality, we choose ε equal to one. Thus, for all vectors x_i from the training collection, it will be true:

$$\begin{cases} w \cdot x_i - b \geq +1, \text{ если } y_i = +1, \\ w \cdot x_i - b \leq -1, \text{ если } y_i = -1. \end{cases} \quad (2.4.22)$$

Condition $-1 < w \cdot x_i - b < 1$ specifies a strip that separates classes. The strip boundaries are two parallel hyperplanes with direction vector w. The points closest to the separating hyperplane are located exactly on the borders of the strip.

The wider the bandwidth, the more confident the documents can be classified, so the SVM method assumes that the widest bandwidth is the best.



dividing strip

Let us formulate the conditions for the problem of the optimal dividing strip, defined by the inequality: $y_i(w \cdot x_i - b) \geq 1$ (this is how the system of equations is rewritten, based on the fact that $y_i \in \{-1, 1\}$). None of the points in the training sample can lie inside this separating band. Under these restrictions x_i, and y_i are constant, as elements of the training collection, and w and b are variables.

From geometric considerations, it is known that the width of

the dividing strip is $2/\|w\|$. Therefore, it is necessary to find such values w and b so that the given linear restrictions are satisfied, and at the same time the vector norm is as small as possible w, that is, it is necessary to minimize:

$$\|w\|^2 = w \cdot w. \qquad (2.4.23)$$

This is a well-known quadratic optimization problem under linear constraints.

If we assume that experts may have made mistakes on training documents during classification, then it is necessary to introduce a set of additional variables $\xi_i \geq 0$, characterizing the magnitude of errors on objects $\{x_1, ..., x_n\}$.. This allows us to soften the restrictions:

$$y_i(w \cdot x_i - b) \geq 1 - \xi_i. \qquad (2.4.24)$$

It is assumed that if $\xi_i = 0$, then there is no error on the document x_i. If $\xi_i > 1$, so, there is an error in the document x_i. If $0 < \xi_i < 1$, then the object falls inside the dividing strip, but belongs to its class by the algorithm.

The problem of finding the optimal separating strip in this case can be reformulated as follows: under certain restrictions, minimize the sum:

$$\|w\|^2 + C\sum_i \xi_i. \qquad (2.4.25)$$

C factor is a method setting parameter that allows you to adjust the relationship between maximizing the width of the separating band and minimizing the total error. The above problem remains a quadratic programming problem, which can be rewritten in the following form:

$$\begin{cases} \dfrac{\|w\|^2}{2} + C\sum_i \xi_i \to \min; \\ y_i(w \cdot x_i - b) + \xi_i \geq 1, \quad i = 1, \ ..., \ n. \end{cases} \qquad (2.4.26)$$

According to the well-known Kuhn-Tucker theorem, such a problem is equivalent to the dual problem of finding the saddle point of the Lagrange function:

$$\begin{cases} \frac{1}{2} w \cdot w + C \sum_i \xi_i - \\ \quad - \sum_i \lambda_i (\xi_i + y_i (w \cdot x_i - b) - 1) \to \min_{w,b} \max_{\lambda}; \qquad (2.4.27) \\ \xi_i \geq 0, \quad \lambda_i \geq 0, \quad i = 1, \ ..., \ n. \end{cases}$$

A necessary condition for the Lagrange method is the equality to zero of the derivatives of the Lagrangian with respect to the variables w and b, whence we obtain:

$$w = \sum_{i=1} \lambda_i y_i x_i, \qquad (2.4.28)$$

those. the desired vector is a linear combination of training vectors for which $\lambda_i \neq 0$. If $\lambda_i > 0$, then the training collection document is called a support vector.

Thus, the equation of the dividing plane has the form:

$$\sum_{i=1} \lambda_i y_i x_i \cdot x - b = 0 . \qquad (2.4.29)$$

Equating the derivative of the Lagrangian with respect to b zero, we obtain:

$$\sum_{i=1} \lambda_i y_i = 0. \qquad (2.4.3\ 0)$$

Substituting the last expression and the expression for w into the Lagrangian, we obtain an equivalent quadratic programming problem containing only dual variables:

$$\begin{cases} \sum_i \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j (x_i \cdot x_j) \to \min\limits_{\lambda}; \\ \sum_{i=1} \lambda_i y_i = 0; \\ C \geq \lambda_i \geq 0, \ \ i = 1, \ ..., \ n. \end{cases} \qquad (2.4.31)$$

At the same time, it is very important that the objective function does not depend on specific values x_i, but on the scalar products between them.

It should be noted that the objective function is convex, so any of its local minimum is global.

The separating stripe classification method has two disadvantages:

- when searching for a dividing strip, only boundary points are important;
- in many cases it is not possible to find an optimal separating band.

To improve the method, the idea of extended space is applied, for which:

1. of vectors to a new, extended space $\boldsymbol{x}$ is selected $\phi(x)$.
2. A new dot product function is automatically applied, which is used when solving a quadratic programming problem, the so-called kernel function as a function : $K(x, y) = \phi(x) \cdot \phi(y)$.

Here is an example of a kernel function that maps a two-dimensional space into a three-dimensional one:

$$\phi(x) = \phi(x_1, x_2) = \left(x_1^2, x_2^2, \sqrt{2} x_1 x_2 \right).$$

Then the core of the function will be:

$$K(x, y) = \phi(x) \cdot \phi(y) = x_1^2 y_1^2 + x_2^2 y_2^2 + 2 x_1 x_2 y_1 y_2 = \left(x, y \right)^2.$$

In practice, one usually chooses not a mapping $\phi(x)$, but a function right away $K(x,y)$, which could be an inner product for some mapping $\phi(x)$.

The kernel function is the main tuning parameter of the support vector machine.

3. We find a separating hyperplane in the new space: using the function, $K(x,y)$ a new matrix of coefficients for the optimization problem is established. In this case, $x_i \cdot x_j$ the value is substituted $K(x_i,x_j)$, and a new optimization problem is solved.
4. Having found w and b, we obtain a surface that classifies $w \cdot \phi(x) - b$ in a new, expanded space.

Let us give a more rigorous definition of a kernel function. A function $K: X \times X \to R$ is called a kernel if it can be represented in the form $K(x,y) = \phi(x) \cdot \phi(y)$ under some mapping $\phi: X \to H$, where is H – the space with the scalar product.

Not every function can be a kernel, but the class of admissible kernels is quite wide. Mercer's theorem is known that a function $K(x,y)$ is a kernel if and only if it is symmetric and non-negatively defined, i.e. when $\int_X \int_X K(x,y)g(x)g(y)dxdy \geq 0$ for any function $g: X \to R$.

Let's consider some properties of kernel functions that allow us to build them in practical problems:

1. Any scalar product is a kernel.
2. The identical unit ($K(x,y) = 1$) is the kernel.
3. The product of kernels is a kernel.
4. For any function, $\phi: X \to R$ the product $K(x,y) = \phi(x) \cdot \phi(y)$ is the kernel.
5. A linear combination of kernels with non-negative coefficients is a kernel.

6. The composition of an arbitrary function and a kernel $K(x,y)=K_0\left(\phi(x),\phi(y)\right)$ is a kernel.

7. If $s: X\times X\to R$ a symmetric integrable function, then $K(x,y)=\int_X s(x,z)s(y,z)dz$ is a kernel.

8. The limit of a locally uniformly convergent sequence of kernels is a kernel.

9. The composition of an arbitrary kernel K_0 and an arbitrary function $f: R\to R$ representable as a convergent power series with non-negative coefficients $K(x,y)=f\left(K_0(x,y)\right)$ is a kernel. In particular, kernels are functions of $f(z)=\exp(z)$ and $f(z)=1/(1-z)$ from the kernel.

For example, in a news content classification system using the well-known LibSVM package (http://www.csie.ntu.edu.tw/~cjlin/libsvm), it is recommended to use the radial basis function as a kernel function:

$$K(x,y)=\exp(-\gamma\|x-y\|^2), \qquad (2.4.3\ 2)$$

where γ is a custom parameter.

Consider a clear example of the transition to an extended space, shown in Fig. 2.4.1. Apparently, round and square figures are not separated by a linear stripe. If, on the other hand, we "bend" the space, going to the third dimension, then these figures can be divided by a plane that cuts off part of the surface with square points. Thus, by bending the space using the mapping $\phi(x)$, one can find a separating hyperplane.

The SVM method has the following advantages:

- outperforms other methods on tests with document arrays;
- when choosing different cores, it allows you to emulate other approaches. For example, a large class of neural networks can be represented using SVMs with specific cores;

– the final rule is chosen not with the help of some heuristics, but by optimizing some objective function.

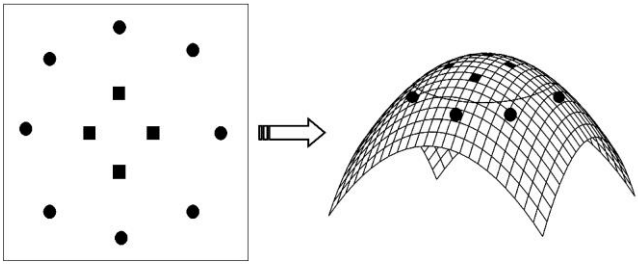


Figure. 2.4.1 – Example of transition to extended space

The disadvantages include:

- few parameters to tune: after the kernel is fixed, the only parameter that varies is the error rate C;
- there are no clear criteria for choosing a core;
- rather slow learning of the classification system.

2.5. Clustering _

All the classical information retrieval models discussed above have a common drawback associated with large dimensions. To ensure efficient work, it is necessary to group both terms and thematically similar documents. Only in this case can the processing of modern information arrays in real time be ensured. In this case, two main techniques come to the rescue – classification and clustering. Classification is the assignment of each document to a certain class with previously known features obtained at the stage of system training. The number of classes in the classification is strictly limited.

Clustering – splitting a set of documents into clusters – subsets, the semantic parameters of which are not known in advance. The number of clusters can be arbitrary or fixed. If classification allows attributing to documents certain features known in advance, then clustering is a more complex process that allows not only the attribution of certain features to

documents, but also the identification of these features themselves – classes.

Classification and clustering are two levels of human involvement in the process of grouping documents. The classification engine is usually trained on selected documents only after the training stage ends by automatically identifying classes (clusters).

The task of clustering is to automatically identify groups of semantically similar documents. However, unlike the classification, the thematic orientation of these groups is not known in advance. The goal of all clustering methods for arrays of documents is to maximize the similarity of documents that fall into the cluster. Therefore, cluster analysis methods are based on such definitions of a cluster as sets of documents, the value of semantic proximity between any two elements of which is not less than a certain threshold, or the value of proximity between any document of the set and the center of the cluster is also not less than a certain threshold.

When using numerical methods of cluster analysis for determining proximity, the following main metrics are used:

Euclidean distance:

$$D(x_i, x_j) = \sqrt{\sum_{k=1}^{N} (x_{ik} - x_{jk})^2}, \qquad (2.5.1)$$

which is a special case of the Minkowski metric for $p = 2$:

$$D_p(x_i, x_j) = \left(\sum_{k=1}^{N} (x_{ik} - x_{jk})^p \right)^{\frac{1}{p}}. \qquad (2.5.2)$$

To group documents represented as vectors of weight values of the terms included in them, a metric based on the scalar product of weight vectors is often used:

$$Sim(x_i, x_j) = \hat{x}_i \cdot \hat{x}_j = \sum_{k=1}^{N} \hat{x}_{ik} \cdot \hat{x}_{jk}, \qquad (2.5.3)$$

where x_i, x_j – documents x_{ik} – is an element of the matrix of weight values of terms included in x_i, $(i = 1, ..., N)$, $x_i^{'}$ is the normalized vector $\hat{x}_i = x_i / |x_i|$.

The initial space of features is usually chosen as the space of terms, which is formed as a result of the analysis of a large array of documents. To conduct such an analysis, different approaches are used – weight, probabilistic, semantic, etc.

In the field of information retrieval, cluster analysis is most often used to solve two problems – grouping documents in databases (information arrays) and grouping search results.

For static documentary arrays, cluster analysis methods have now received great development and popularity. At the same time, the question of applying these methods to dynamically changing information flows, which, in addition to dynamics, are also characterized by large volumes, remains open.

Cluster analysis methods are widely used in the procedures for ranking the responses of information retrieval systems, in the construction of personalized search folders, personal search interfaces for users of information retrieval systems.

2.5.1. Method *k* -means

An iterative algorithm for k – means (*k -means) cluster analysis of grouping documents by a fixed number of clusters is as follows:* k vectors are randomly selected, which are defined as centroids (the most typical representatives) of clusters. Then k clusters $\{C_1, C_2, ..., C_k\}$ are filled – for each of the vectors that remain, the proximity to the centroid of the corresponding cluster is determined in some way. Proximity can be defined in many ways, in particular, as a normalized dot product:

$$Sim(x, c^j) = \frac{\sum_{k=1}^{N} x_k c_k^j}{|x||c^j|}, \quad (2.5.4)$$

where x – document, c^j $(j = 1, ..., k)$ – cluster centroid C_j, N – dimension of term space.

After that, the vector is assigned to the cluster to which it is closest. Further, the vectors are grouped and renumbered according to their belonging to the clusters. Then, for each of the new clusters, the centroid $c^i = (c_1^i, ..., c_N^i)$ – vector is re-determined, which is closest to all vectors from this cluster, the coordinates of which are determined, for example, as follows:

$$c_k^i = \frac{1}{|C_i|} \sum_{x \in C_i} x_k. \qquad (2.5.5)$$

After that, the cluster filling process is carried out again, then the calculation of new centroids, etc., until the cluster formation process stabilizes (or if the decrease in the sum of the distance from each element to the center of its cluster is less than some specified threshold value).

k – means algorithm maximizes the clustering quality function Q:

$$Q(C_1, ..., C_k) = \sum_{i=1}^{k} \sum_{x \in C_i} Sim(x, c^j). \qquad (2.5.6)$$

Unlike the LSI method, k – means can be used to group dynamic information flows due to its computational simplicity – $O(kn)$, where n is the number of grouping objects (documents). The disadvantage of the method is that each document can fall into only one cluster.

2.5.2. Hierarchical grouping-union

Hierarchical grouping-union (Hierarchical Agglomerative Clustering, HAC) begins with the fact that each object is assigned a separate cluster, and then the clusters that are closest to each other are combined, in accordance with the selected criterion. The algorithm ends when all objects are combined into a single cluster. The history of associations forms a binary tree of the hierarchy of clusters.

Varieties of the HAC algorithm differ in the choice of proximity (similarity) criteria between clusters. For example, the proximity between two clusters can be calculated as the maximum proximity between objects from these clusters.

Hierarchical clustering is very often used in sociological analysis, biology, economics, etc. Mainly where the number of clusters is not known in advance.

For hierarchical clustering, it is necessary to somehow determine the distance between the nodes of our graph (network). Those. we need to obtain a quantitative estimate of the proximity of nodes, similar to the distance in the usual Euclidean space. Let's look at two of the most commonly used definitions. The first is the Euclidean distance (Euclidean distance) is defined as follows:

$$x_{i,j} = \sqrt{\sum_{k \neq i,j}^{N} \left(A_{ik} - A_{jk} \right)^2} \text{ , (2.5.7)}$$

where N is the number of nodes in the network. The Euclidean distance is exactly zero for completely structurally equivalent nodes, and increases for nodes that have no common neighbors. The second definition is based on the Pearson correlation between rows (columns) of the incidence matrix.

$$x_{i,j} = \frac{\frac{1}{N} \cdot \sum_{k=1}^{N} \left(A_{ik} - \mu_i \right) \cdot \left(A_{ik} - \mu_i \right)}{\sigma_i \cdot \sigma_j}, \qquad (2.5.8)$$

Where

$$\mu_i = \frac{1}{N} \cdot \sum_{j}^{N} A_{ij} \text{ , } \sigma_i^2 = \frac{1}{N} \cdot \sum_{j}^{N} \left(A_{ij} - \mu_i \right)^2 \text{ . (2.5.9)}$$

Here structurally equivalent nodes are those that have a large correlation coefficient.

After we have determined the distance matrix in any way, we can proceed directly to hierarchical clustering. Let us dwell in more detail on the Anglomerative algorithm. Let us first describe the idea of the algorithm. Let's say we have points $x_1, x_2 \ldots x_N$ and

a matrix of relative distances x_{ij}. At the first step, each point is considered a separate cluster. Then we unite the nearest (in terms of distance) points and consider them as one cluster, and so on until all points are involved. At the output, we get a tree (dendogram). When calculating distances between clusters, one of the following two algorithms is most commonly used. Single – link algorithm calculates the minimum of possible distances between pairs of nodes in the cluster, Complete – link algorithm calculates the maximum of these distances.

Let's give a step-by-step example for Single – link algorithm. Let there be a graph shown in Fig. 2.5.1.

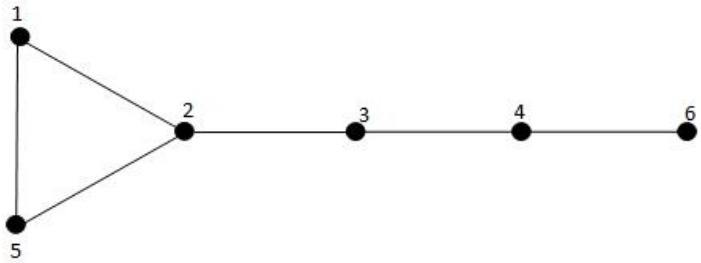


Figure. 2.5.1 – Graph example

Having written the matrix of Euclidean distances (1), we see that the minimum distance equal to zero corresponds to nodes 1 and 5. Therefore, at the first step, we combine these nodes into one cluster.

	(1)	(2)	(3)	(4)	(5)	(6)
(1)	0	1	1.414	2	0	1.732
(2)	1	0	1.732	1.732	1	2
$x =$ (3)	1.414	1.732	0	1.414	1.414	1
(4)	2	1.732	1.414	0	2	1
(5)	0	1	1.414	2	0	1.732
(6)	1.732	2	1	1	1.732	0

At the second step, we again write the distance matrix using the Single – link algorithm. The next distance after the zero distance is equal to one. And at this step, we get two clusters, which in total include all network nodes.

If we cut the dendrogram along the dotted line 1, then we will get two clusters containing nodes (1, 5, 2) and (3, 4, 6), respectively. Which also corresponds to the intuitive division into clusters, see fig. 1. If we cut along the second dotted line, we get three clusters (1, 5), (2) and (3, 4, 6).

	$\begin{pmatrix}1\\5\end{pmatrix}$	(2)	(3)	(4)	(6)
$\begin{pmatrix}1\\5\end{pmatrix}$	0	1	1.414	2	1.732
(2)	1	0	1.732	1.732	2
(3)	1.414	1.732	0	1.414	1
(4)	2	1.732	1.414	0	1
(6)	1.727	2	1	1	0

On fig. 2.5.2 shows the final dendrogram of the network.

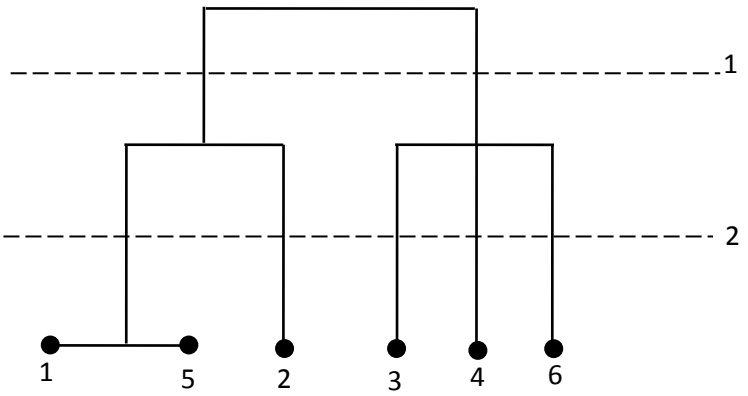


Figure. 2.5.2 – Single – link algorithm used

If we use the Complete – link algorithm for hierarchical clustering, we will get a slightly different dendrogram, see Fig. 2.5.3.

The difference in the algorithms will appear at the second step of construction. When using the Single – link algorithm, node 3 is added to one cluster to nodes (4, 6), since it is connected to node 6 by a unit distance, the same as the intracluster distance. And when using the Complete – link algorithm, node 3 is connected in the next step, because here it is necessary to look at the maximum distance, i.e. by the distance from node 3 to node 4, which is greater than the intracluster distance between nodes 4 and 6.

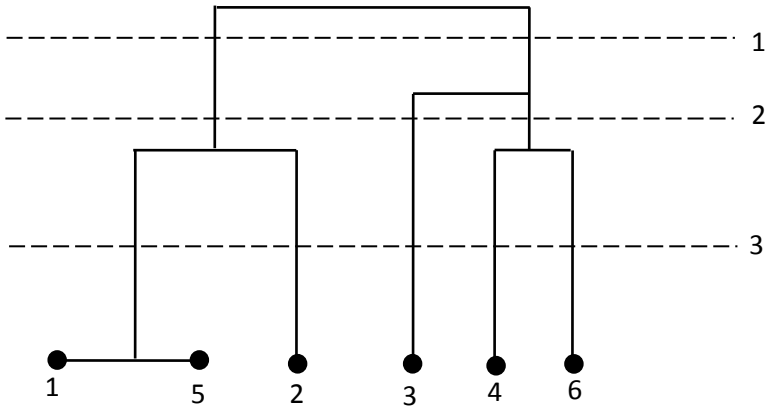


Figure. 2.5.3 – Complete algorithm used – link

2.5.3. Latent semantic analysis

The method of cluster analysis LSA / LSI (latent semantic analysis / indexing) [is based on singular value decomposition of matrices (SVD, S ingular V alue D ecomposition). Let an array of documents $D=\{d_j \mid j=1, \ ..., \ n\}$ be associated with a matrix A whose rows correspond to documents and whose columns correspond to the weight values of terms (the size of the term dictionary is m). The singular $s_{ii} \geq 0.$value r decomposition of a Amatrix U of $s_{ij}=0,$dimension $m \times r$ rank $r \times n$ is $m \times n$ its S –

decomposition $V-$ of $A=USV^T$ the $i \neq j$ form S are called the singular values of the matrix A.

Orthogonal matrices U and V have the following property:

$$UU^T = V^T V = I. \qquad (2.5.9)$$

It is proved that the above partition of the matrix A has the peculiarity that if S only k the largest singular values are left in the matrix (we denote such a matrix as S_k), and in the matrices U and $V-$ only columns corresponding to these values (respectively, matrices U_k, V_k), then the matrix $A_k = U_k \cdot S_k \cdot V_k^T$ will be the best Frobenius approximation of the original matrix A by a matrix with a rank not exceeding k. Recall that the matrix norm X dimension $M \cdot N$ according to Frobenius is the expression:

$$\|X\|_F = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij}^2}. \qquad (2.5.10)$$

Thus, for matrices A and A_k it is proved that:

$$A_k = \underset{X:rank(X)=k}{\arg\min} \|A-X\|_F. \qquad (2.5.11)$$

In accordance with the LSA method, not all, but only k the largest singular values of the matrix are taken into consideration A, and each such value is assigned one cluster.

A_k defines k an -dimensional factor space onto which both documents (using the matrix V) and terms (using the matrix U) are projected. In the resulting factor space, documents and terms are grouped into areas that have some common hidden meaning. Those. the resulting areas are clusters.

Choosing the best dimension k for LSA is a matter of individual research. Ideally, k it should be large enough to display the entire real-life data structure, but at the same time small enough not to take into account noise – random dependencies.

In the practice of information retrieval, special importance is given to matrices U_k and V_k^T. Rows of a matrix U_k are considered as images of terms in $k-$a dimensional real space. Similarly, matrix columns V_k^T are considered as images of documents in the same $k-$dimensional space. In other words, these vectors define the desired representation of terms and documents in $k-$the dimensional space of hidden factors.

There are also methods for incrementally updating all values used in an LSA. When replenishing with a new document d (for example, a new result of a search for a query) an information array for which a singular decomposition has already been carried out, it is possible not to calculate the decomposition again. It is enough to approximate it by calculating the image of a new document based on the previously calculated images of terms and weights of factors. Let $d-$the vector of weights of the terms of the new document (a new column of the matrix A), then its image can be calculated by the formula: $d' = S_k^{-1} U_k^T d$.

If q $-$the user's request vector of dimension m, $i-$th element of which is equal to 1, if the term with the number i is included in the request, and 0 otherwise, then the image of the request q in the space of latent factors will look like: $q' = q^T U_k S_k^{-1}$.

In this case, the measure of proximity between the query q and the document d is estimated by the value of the scalar product of the vectors q' and $V_k^T \{d\}$ (here $V_k^T \{d\}$ denotes $d-$the th column of the matrix V_k^T).

In information retrieval, as a result of the fact that the least significant singular values are discarded, a space of orthogonal factors is formed that plays the role of generalized terms. As a result, there is a “convergence” of documents from subject areas that are close in content, and the problems of synonymy and homonymy of terms are partially solved.

The LSA method is widely used in ranking the output of information retrieval systems based on citation. This is the HITS (Hyperlink Induced Topic Search) algorithm – one of the two

most popular today in the field of information retrieval. If we recall the concept of the incidence matrix A (section 2.8), then the HITS algorithm ensures the selection of the most authoritative documents (authors - a_p or intermediaries – h_p) that correspond to the eigenvectors of the matrices AA^T and $A^T A$ with the largest modules of eigenvalues. Let us show that the HITS algorithm is equivalent to LSA. Indeed, let, in accordance with the singular value decomposition: $A = USV^T$, S – a square diagonal matrix. Then $A^T A = USV^T VSU^T = = USISU^T = US^2U^T$, Where S^2 – diagonal matrix with entries S_{ii}^2. Obviously, as in LSA, the eigenvectors that correspond to the largest singular values AA^T and/or $A^T A$, will correspond to the statistically most important authors and/or contributors.

Along with the fact that the LSA method does not need to be pre-tuned to a specific set of documents and qualitatively reveals latent factors, its disadvantages include low performance (the SVD calculation speed corresponds to the order $O(N^2 \cdot k)$, where $N = |D| + |T|$, D – many documents T – many terms, k – dimension of the space of factors) and the fact that it does not provide for the possibility of intersection of clusters, which is contrary to practice. In addition, due to its computational complexity, the LSA method is used only for relatively small matrices.

Part II. Algorithms, methods, phenomena

3. Some methods and techniques

The second part of the manual is largely methodological notes for the lecturer. The introduction of this material into the course of lectures depends both on the preferences of the lecturer and on the preparedness of the audience.

Part of the material is necessary for a deeper understanding of the main part (for example, material related to the theory of percolation or the theory of phase transitions of the second kind). The other part of the material is what is considered to be “every educated person knows”, but as a rule (surprisingly) is not read in standard courses. Basically, these are the so-called tricks that allow you to get, albeit approximately and not always mathematically rigorously, a result. "Pure" mathematicians are not interested in such methods, and if they are presented, then at such a level of "deep" theory that there is no question of a specific application.

As an example of such techniques, we can name the small parameter method (without which not a single application worker can do), or, for example, the extremely useful Padé approximant method.

Even when the problem can be solved exactly, the rigorous solution can be so cumbersome that it is more convenient to use an approximate solution. Here we give an example from Greenberg's book, which clearly shows that the solution obtained "on the forehead", by the standard method of separation of variables, can give a completely "inedible" solution.

3.0. A simple boundary value problem

1. An example of one simple boundary value problem of mathematical physics, or about how to and how not to solve problems.

Boundary value problem of the first kind or Dirichlet problem (internal) for a rectangle

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = 0. \qquad (3.0.1)$$

The standard solution method is separation of variables:

$$\varphi(x, y) = X(x)Y(y) \qquad (3.0.2)$$

the analytical implementation of which required five pages of text.

In classical textbooks (for example, Tikhonov and Samarsky), they practically *never* teach techniques, but teach general methods.

Grinberg G.A. Selected questions of the mathematical theory of electrical and magnetic phenomena. – M.; L.: Academy of Sciences of the USSR, 1948.

Tikhonov A.N., Samarsky A.A. Equations of Mathematical Physics (5th ed.). – M.: Nauka, 1977. – 735 p.

G.A. Grinberg, who had to deal with the calculation of specific devices, needed to obtain a specific solution, and not a "theorem about its existence in a given class of functions." G.A. Greenberg developed a method that makes it possible to obtain solutions with well-converging series for both the solution itself and its derivatives. This method is fundamentally different from the method of separation of variables – in the mentioned book this method is clearly formulated – it is emphasized that the solution obtained in this way is not the sum of particular solutions of the proposed equation, but gives an expansion of the solution in a series in terms of some eigenfunctions of the equations.

Consider a very simple example given in Greenberg's book, which does not even require the application of the above method.

The standard solution of the Dirichlet problem is based on the separation of variables and leads to the expression:

$$\varphi = u^{(1)} + u^{(2)}, \qquad (3.0.2)$$

where $u^{(1)}(x, y)$ and $u^{(2)}(x, y)$ are the sums of the products:

$$u^{(1)} = \frac{2}{b}\sum_{k=1}^{\infty} \frac{sh\frac{k\pi(a-x)}{b}\int_0^b F_1(y)\sin\frac{k\pi y}{b}dy + sh\frac{k\pi x}{b}\int_0^b F_2(y)\sin\frac{k\pi y}{b}dy}{sh\frac{k\pi a}{b}}\sin\frac{k\pi y}{b},$$

$$u^{(2)} = \frac{2}{a}\sum_{k=1}^{\infty} \frac{sh\frac{k\pi(b-y)}{a}\int_0^a \psi_1(x)\sin\frac{k\pi x}{a}dx + sh\frac{k\pi x}{a}\int_0^a \psi_2(x)\sin\frac{k\pi x}{a}dx}{sh\frac{k\pi b}{a}}\sin\frac{k\pi x}{a},$$

(3.0.3)

where the boundary conditions are presented in general form:

$$\varphi|_{x=0} = F_1(y), \varphi|_{x=a} = F_2(y),$$
$$\varphi|_{y=0} = \psi_1(x), \varphi|_{y=b} = \psi_2(x). \qquad (3.0.4)$$

We are now interested in the specific case where

$$F_1(y) = 0, \quad \psi_1(x) = 0,$$
$$F_2(y) = ay, \quad \psi_2(x) = bx, \qquad (3.0.5)$$

i.e.:

$$\varphi|_{x=0} = 0, \varphi|_{x=a} = ay,$$
$$\varphi|_{y=0} = 0, \varphi|_{y=b} = bx. \qquad (3.0.6)$$

Then, according to the above, substituting these values into the series for $u^{(1)}$ and $u^{(2)}$ we find:

$$\varphi(x,y) = \frac{2ab}{\pi}\left\{\sum_{k=1}^{\infty}(-1)^{k+1}\frac{sh\frac{k\pi x}{b}}{ksh\frac{k\pi a}{b}}\sin\frac{k\pi y}{b} + \sum_{k=1}^{\infty}(-1)^{k+1}\frac{sh\frac{k\pi y}{a}}{ksh\frac{k\pi b}{a}}\sin\frac{k\pi x}{a}\right\}.$$

(3.0.7)

This is a very complex expression, representing two infinite sums, from the form of which it is impossible to understand how the solution behaves. It is easy to verify directly that the equation satisfies the condition:

$$\varphi(x,y) = x \cdot y, \qquad (3.0.8)$$

and since the solution is unique, then this is the solution, i.e. instead of complex series, only a simple expression can be used.

So what is the “reception” in the considered case? Its essence can be formulated in this way: Before solving the equation mat. physics, try to make a substitution to simplify:

$$\varphi(x,y) = < \text{ some function } > + u(x,y).$$

3.1. Small parameter

The small parameter method is used in many areas of science and technology, for example, theoretical physics, the theory of deterministic chaos, it is difficult to say where it is not used. In the book of Blekhman, Myshkis and Panovko, it is noted that "the perturbation method is one of the most common in applied mathematics."

A large number of monographs are devoted to the perturbation method, for example, Naife's monograph, which contains many examples. Here we consider only one simple example – one of the most famous and useful problems – the anharmonic oscillator problem. The zero approximation of this problem is a one-dimensional harmonic oscillator (“ball on a spring”), described by a linear equation

Blekhman I.I., Myshkis A.D., Panovko Ya.G. Mechanics and applied mathematics. Logic and features of applications of mathematics. – M.: Mir, 1983.

Naife A. Introduction to perturbation methods. – M.: Mir, 1984.

Myshkis A.D. Elements of the theory of mathematical models. – M.: Nauka, 1994.

$$m\frac{d^2x}{dt^2} = -kx, \qquad (3.1.1)$$

where m is the mass, k is the stiffness of the spring, x is the deviation from the equilibrium position, $F = -kx$ is Hooke's law.

This equation can be conveniently written as:

$$\frac{d^2x}{dt^2} + \omega^2 x = 0\,, \qquad (3.1.2)$$

where $\omega = \sqrt{k/m}$ is the cyclic frequency.

In this form, this equation describes a huge class of phenomena, not only a "ball on a spring", but also a pendulum (in this case, the x deflection angle), oscillations in LC the -loop (x _ capacitor charge) and much more.

The solution of equation (2), as it is easy to verify by a simple substitution

$$x(t) = x_0 \sin(\omega t + \varphi_0)\,, \quad (3.1.3)$$

where φ_0 is the initial phase and x_0 is the amplitude.

There are situations when a linear approximation is not enough and Hooke's law must be replaced by a more complex, non-linear one.

$$F(x) = -kx + \gamma x^2 + \ldots (3.1.4)$$

where it is assumed that each next expansion term is much less than the previous one, i.e.

$$\gamma x_0^2 << k|x_0|\,, \ \ldots \ (3.1.5)$$

Generally speaking, (3.1.4) is a consequence of the Taylor series expansion of the potential energy $U(x)$, recall that the force $F = -gradU$. An oscillating particle is in a potential well, and in the simplest case, when the well is parabolic, $U(x) \sim x^2$ Hooke's law holds.

In a more general case, taking into account only the first two terms in the expansion (3.1.4), instead of the harmonic oscillator equation (3.1.1), we obtain the nonlinear equation

$$m\frac{d^2x}{dt^2} + kx = \gamma x^2\,, \quad (3.1.6)$$

The solution of which in an analytical form is not possible.

For what follows, it is convenient to write (3.1.6) in the dimensionless form

$$\frac{d^2\psi}{d\tau^2}+\psi=\varepsilon\psi^2\text{ , (3.1.7)}$$

where $\psi=x/x_0$ is a dimensionless coordinate, a τ is time, and $\varepsilon=\gamma x_0/\omega$ is a dimensionless parameter, which, in the accepted approximation ($\gamma x_0^2 << k|x_0|$), is a small parameter of $\varepsilon \ll 1$.

An approximate solution of equation (3.1.7) will be sought in the form of an expansion in a small parameter

$$\psi(\tau)=\psi_0(\tau)+\varepsilon\psi_1(\tau)+\text{... (3.1.8)}$$

Substituting (5) into (4) we get

$$\frac{d^2\psi_0}{d\tau^2}+\varepsilon\frac{d^2\psi_1}{d\tau^2}+\psi_0+\varepsilon\psi_1=\varepsilon{\psi_0}^2+\varepsilon^2\psi_0\psi_1+\text{... (3.1.9)}$$

In the zeroth approximation in the small parameter, i.e. leaving in (3.1.9) only terms c ε^0 and discarding all terms c ε^1, ε^2 , etc. we have

$$\frac{d^2\psi_0}{d\tau^2}+\psi_0=0\text{ , (3.1.10)}$$

Linear equation (harmonic oscillator equation), the solution of which can be written as

$$\psi_0(\tau)=\sin\tau\text{ . (3.1.11)}$$

In the first approximation with respect to the small parameter ε in (3.1.9), the terms with ε^2 and above are discarded and we obtain

$$\frac{d^2\psi_1}{d\tau^2}+\psi_1={\psi_0}^2\text{ . (3.1.12)}$$

This equation is linear in ψ_1, and the function ψ_0 was found earlier (3.1.11). The solution of equation (3.1.12) ($\sin^2\tau=(1-\cos 2\tau)/2$) has the form

$$\psi_1(\tau)=\frac{1}{2}+\frac{1}{6}\cos 2\tau\text{ . (3.1.13)}$$

Restricting ourselves to the first approximation in ε for (3.1.9), we have:

$$\psi(\tau)=\psi_0+\varepsilon\psi_1=\sin\tau+\frac{1}{2}\varepsilon+\frac{1}{6}\varepsilon\cos 2\tau\text{. (3.1.14)}$$

Or returning from a dimensionless form to the original notation

$$x(t) = x_0 \sin(\omega t) + \frac{\gamma x_0^2}{2\omega} + \frac{\gamma x_0^2}{6\omega}\cos(2\omega t). \quad (3.1.15)$$

Thus, the solution of a nonlinear equation by the small parameter method has been reduced to the sequential solution of linear equations.

Even the first approximation with respect to the small parameter (3.1.15) allows us to draw two nontrivial conclusions about the influence of anharmonicity:

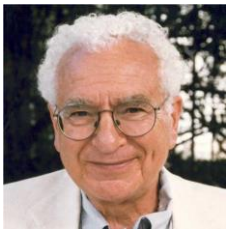
1. in addition to oscillations with frequency ω, which take place in the linear case, there are oscillations with a doubled frequency
2. the midpoint of the oscillations, which was chosen to be zero in the harmonic oscillator, is now shifted (see the second term in (3.14.15)).

3.2. Asymptotic series and expansions

One has to be surprised that the approximate solution of equations using the method of expansion in terms of a small parameter is not included in the standard courses of mathematical analysis.

A small parameter is usually denoted by the letter ε / Up to what values $(\varepsilon \ll 1)$ can be considered small is the most acute question.

Example: two rows $n-$e whose members are proportional: 1) $\frac{1000^n}{n!}$; 2) $\frac{n!}{1000^n}$. Formally, the 1st series converges quickly, its millionth term is only 1/999999; 2nd row diverges.



Murray Gell-Mann, Nobel laureate: "In fact, every theorist in his own work assumes some parameters are small, and then attacks others who do the same, accusing them of unnaturalness."

Poincaré: “on the contrary (having received an approximate solution in the form of such series, with *a finite* number of terms), astronomers will consider the 1st series to be divergent,

because the first 1000 members of the series increase; and the 2nd – convergent, since its first 1000 members are decreasing.

In order to find the sum of the series $\sum_n a_n$, it is necessary near 140 the terms, moreover, before a_n will become small enough and stop changing the sum, their values will grow to a value $\sim 10^{42}$ (see Fig. 3.2.1).

Consider now two rows, for $x \ll 1$

$$S_1(x,N) = \sum_k a_k x^k = \sum_{k=1}^{N} \frac{100^k}{k!} x^k \qquad (3.2.1)$$

And

$$S_2(x,N) = \sum_k b_k x^k = \sum_{k=1}^{N} \frac{k!}{100^k} x^k \qquad (3.2.2)$$

The first row $S_1(x,N)$ is the expansion in terms of a small parameter $x \ll 1$, its sum sets the function more accurately, the larger N. The series at $N \to \infty$ converges.

The second row $N \to \infty$ diverges at, and as we are taught in standard courses of mathematical analysis, it makes no sense to use it.

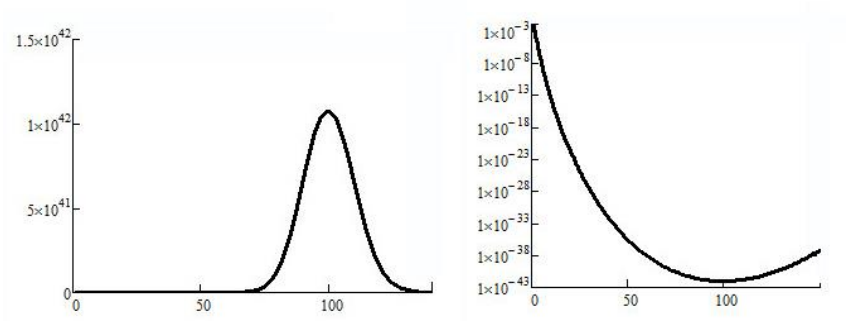


Figure. 3.2.1 – Behavior a_n and b_n growth n

In practice, the opposite happens. Despite the convergence of the series, it is $S_1(x,N)$ extremely inconvenient to use it, the number of terms in the series must be (see Fig. 3.2.2) not less

than 20. As for the second row $S_2(x,N)$, for $x \ll 1$ already at small values, N the sum stops growing.

Below are examples of asymptotic series for two functions.

The function $f(x)$ is given by the integral :

$$f(x) = \int_0^\infty \frac{xe^{-\xi}}{x+\xi} d\xi. \qquad (3.2.3)$$

Jules Henri Poincare
(1854 – 1912)

You need to get the behavior $f(x)$ for $x \gg 1$.

We expand $x/(x+\xi)$ in a series in powers ξ:

$$\frac{x}{x+\xi} = \sum_{n=0}^{\infty} \frac{(-1)^n \xi^n}{x^n}. \qquad (3.2.4)$$

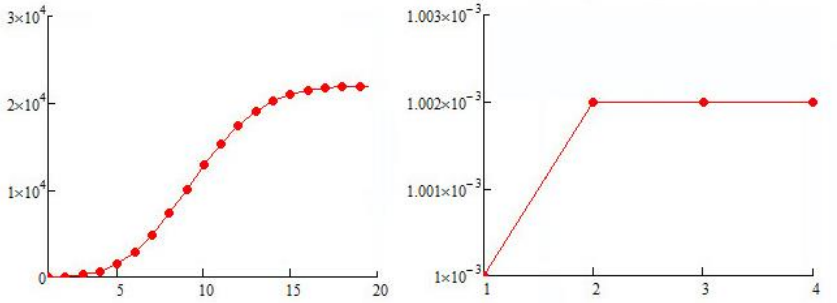


Figure. 3.2.2 – Dependence of the sum of rads $S_1(x,N)$ and $S_2(x,N)$ on N

Substituting this expansion into the integral, we find:

$$f(x)=\sum_{n=0}^{\infty}\frac{(-1)^n n!}{x^n} \qquad (3.2.5)$$

This series, of course, diverges, i.e. this sum is infinity. However, it can be used. To do this, we represent this sum in the form of two terms – a truncated series and a remainder:

$$f(x)=\sum_{n=0}^{N}\frac{(-1)^n n!}{x^n}+R_N(x) \qquad (3.2.6)$$

It can be shown that

$$\left|R_N(x)\right|<\frac{N!}{x^N} \qquad (3.2.7)$$

those. this replacement $f(x)$ by a series truncated at N the i-th term gives an error not exceeding the first discarded term, namely $N+1$ the i-th. Thus, for a fixed N and $x\gg 1$ $(x\to\infty)$ the error is $R_N\to 0$ (arbitrarily small) and $f(x)$ can be represented as an asymptotic series

$$f(x)\sim\sum_{n=0}^{N}\frac{(-1)^n n!}{x^n} \qquad (3.2.8)$$

In the considered example, the small parameter is $\varepsilon=1/x$, which gives, for example, $\varepsilon=0.1$ an error at the tenth step $R_{10}(x=1/\varepsilon=10)\approx 4\cdot 10^{-4}$, which is quite acceptable for many approximate calculations. If the small parameter is ten times smaller $\varepsilon=0.01$, then the error becomes negligible $R_{10}(x=1/\varepsilon=100)\approx 4\cdot 10^{-14}$. Now we can briefly formulate the difference between the expansion in a small parameter and the asymptotic expansion. In the case of expansion in a small parameter, the passage to the limit in the sum from $n=0$ to N has the form:

$\varepsilon\ll 1$ - of course $N\to\infty$, (3.2.9)

and for the asymptotic series

Andrianov I. V., Barantsev R. G., Manevich L. I. Asymptotic Mathematics and Synergetics: A Path to Integral Simplicity. – M.: Editorial URSS, 2004.

Andrianov IV, Manevich LI Asymptotology: ideas, methods, results. – M.: Aslan, 1994.

$\varepsilon \to 0$, N - of course. (3.2.10)

Let us give the second example, the expansion of the zero-order Bessel function $J_0(x)$ for $x \gg 1$ (small parameter $\varepsilon = 1/x$). The asymptotic expansion $J_0(x)$ has the form

$$J_0(x) = \sqrt{\frac{2}{\pi x}}\left[u(x)\cos\left(x-\frac{\pi}{4}\right)+v(x)\sin\left(x-\frac{\pi}{4}\right)\right] \quad (3.2.11)$$

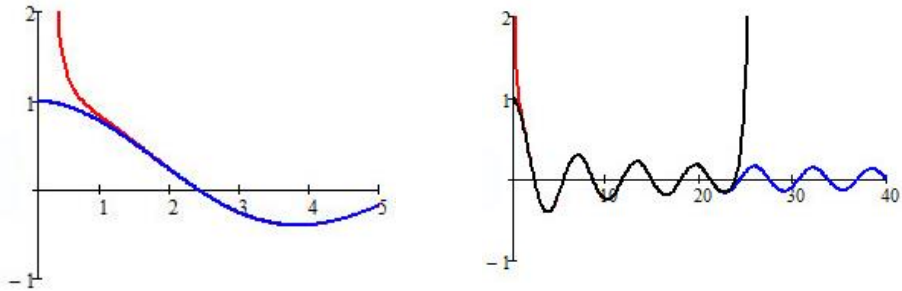
,

Where

$$u(x) = 1 - \frac{1^2 \cdot 3^2}{4^2 \cdot 2^2 \cdot 2! x^2} + \frac{1^2 \cdot 3^2 \cdot 5^2 \cdot 7^2}{4^4 \cdot 2^4 \cdot 4! x^4} + \ldots \quad (3.2.12)$$

$$v(x) = \frac{1}{4 \cdot 2x} - \frac{1^2 \cdot 3^2 \cdot 5^2}{4^3 \cdot 2^3 \cdot 3! x^3} + \ldots \quad (3.2.13)$$

The series $u(x)$ and $v(x)$ diverge, however, as can be seen from Fig. 3.2.3 already the first few members of the series perfectly describe $J_0(x)$ at $x > 1$, i.e. for a small order parameter of unity.



A b

Figure. 3.2.3. Zero-order Bessel function: a – $J_0(x)$ on the interval $(0 \div 5)$, 1 – exact value, 2 – asymptotic expansion; b – in the interval $(1 \div 40)$, solid line – the exact value $J_0(x)$ and asymptotic expansion

3.3. Padé approximants and expansion in a small parameter

G. Stanley "Phase transitions and critical phenomena" M.: Mir, 1973.

Padé approximation of a function $f(x)$ expanded in a power series

$$f(x) = \sum a_n x^n \text{ , (3.3.1)}$$

is called a fractional power function

$$R(x) = \frac{\sum_{n=0}^{N} p_n x^n}{\sum_{m=0}^{M} q_m x^m} \text{ , (3.3.2)}$$

such that, when expanded in a power series, it coincides with the expansion $f(x)$ up to coefficients of x^{N+M} .

Behind this dry definition lies an idea which, while being very simple, in a number of cases makes it possible to obtain "fantastic results".

As an example, consider the approximation of a function $tg\ x$ by a power series – fig. This expansion, of course, does not work near $x = \frac{\pi}{2}$, where $tg\ x$ it diverges, since the polynomial cannot diverge (take an infinite value) at a finite value of the argument. At small values, the x power x expansion $tg\ x$ describes well $tg\ x$, but the closer x to $\pi/2$, the greater the discrepancy between the function and the expansion. As can be seen from fig. an increase in the number of terms in the expansion practically does not improve the situation near $\pi/2$. So in fig. 3.3.1. bottom line – expansion $tg\ x$ up to the 9th degree x :

$$f_9(x) = x + \frac{x^3}{3} + \frac{2}{15}x^5 + \frac{17}{315}x^7 + \frac{1382}{155925}x^9, \qquad (3.3.3)$$

second from the bottom up to the 15th power, and finally third from the bottom up to the 29th power, so that the last term of this expansion is

$$\frac{689005380505609448}{263505041412702261046875}\cdot x^{29} \qquad (3.3.4)$$

But even such a decomposition (which is unlikely to be done by hand) is of little help.

Let us now see how the Padé approximant method copes with this problem. We choose it in the form:

$$R(x)=\frac{ax}{1+bx^2+cx^4}, \qquad (3.3.5)$$

i.e., in the form of a ratio of polynomials not higher than the fourth degree.

power series x expansion in $R(x)$ has the form:

$$R(x)\approx ax-abx^3-a\left(c-b^2\right)x^5, \qquad (3.3.6)$$

Comparing this decomposition with the decomposition $tg\ x$

$$tg\ x\approx x+\frac{x^3}{3}+\frac{2}{15}x^5+..., \qquad (3.3.7)$$

and equating the coefficients at the same powers of the variable, we obtain a system of equations for the coefficients a, b and c:

$$a=1.\ ab=\frac{1}{3}.\ a\left(c-b^2\right)=\frac{2}{15}. \qquad (3.3.8)$$

whence $a=1$, $b=-1/3$, $c=-0.022$, and, consequently, the Padé approximant for the function $tg\ x$ has the form:

$$R(x)=\frac{x}{1-\frac{1}{3}x^2-0.022\ x^3}, \qquad (3.3.9)$$

see the third curve from the bottom in fig. 3.3.1, the topmost curve is the function $tg\ x$.

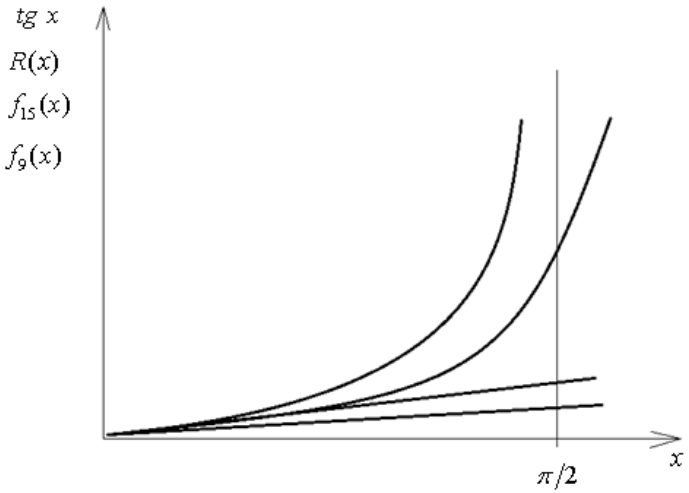


Figure. 3.3.1. Comparison of the function $tg\ x$, Padé approximants and two series expansions $f_{15}(x)$ (up to the 15th power x) and $f_9(x)$ (up to the 9th). Function graphs are arranged from top to bottom according to the notation on the vertical axis

A comparison between $tg\ x$, $R(x)$, $f_{15}(x)$ and $f_9(x)$ shows that the Padé approximant of only the fourth degree describes the behavior much better $tg\ x$ in the entire range from 0 to $\pi/2$. And what is very important describes (albeit not very accurately) the divergence near $\pi/2$.

The second example is related to a real physical problem – determining the viscosity of a suspension, which is a liquid with a viscosity μ_0 in which there are hard particles with a concentration of p.

The greater the concentration of particles, the greater the viscosity of the suspension – μ_e. A. Einstein was the first to solve such a problem for $p << 1$ in 1905:

$$\mu_e = \mu_0(1+\frac{5}{2}p).\ p << 1. \qquad (3.3.10)$$

obtaining an approximate solution up to the first order of a small particle concentration.

Only 67 years later, in 1972 G. Batchelor and J. Green (after very complex calculations) managed to obtain the following approximation

$$\mu_e = \mu_0\left(1+\frac{5}{2}\mathrm{p}+\left(5,2\pm 0,3\right)\mathrm{p}^2\right). \qquad (3.3.11)$$

Let us now construct the Padé approximant for these two approximations.

$$R_1(p) = \mu_0 \frac{1}{1+ap} \approx \mu_0(1-ap), \qquad (3.3.12)$$

whence, according to A. Einstein's approximation $a = 5/2$, and, therefore:

$$\mu_e \approx \mu_0 \frac{1}{1-\frac{5}{2}p} \qquad (3.3.13)$$

from which a fundamentally new fact immediately follows – when approaching p a certain value, in this case, to, $p_c = 2/5$ the effective viscosity μ_c diverges. Such a divergence does indeed take place in reality, the experiment shows that at a certain concentration of hard particles, the liquid will stop flowing.

We also note that if we expand the Padé approximant (3.3.13) $\mu_e = \mu_0 / \left(1-\frac{5}{2}p\right)$ in a series up to p^2:

$$\mu_e \approx \mu_0 \frac{1}{1-\frac{5}{2}p} \approx \mu_0\left(1+\frac{5}{2}p+\frac{25}{4}p^2\right), \qquad (3.3.14)$$

then we get a good second approximation, where about p^2 stands $25/4 \approx 6,25$ instead of $5,2\pm 0,3$, obtained by Batchelor and Green.

To clarify p_c - the value at which the effective viscosity becomes very large, you can construct the Padé approximant using the second approximation obtained by Batchelor and

Green, assuming that the coefficient at p^2 is exactly 5 (Fig. 3.3.2).

$$R_2 = \mu_0 \frac{1+Ap}{1-Bp} \approx$$
$$\approx \mu_0 \left(1+\left(A+B\right)p + B\left(A+B\right)p^2\right), \qquad (3.3.1\ 5)$$

where

$$A+B=\frac{5}{2}.\ B\left(A+B\right)=5. \qquad (3.3.1\ 6)$$

and therefore

$$\mu_e \approx \mu_0 \frac{1+p/2}{1-2p}, \qquad (3.3.1\ 7)$$

what for p_c will give $p_c = 1/2$.

This value agrees very well with the experimental value.

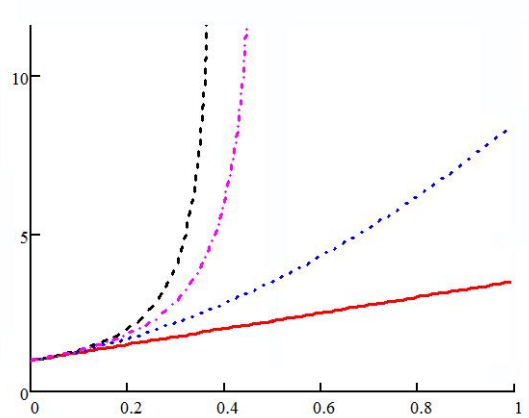


Figure. 3.2.2 – Effective viscosity. The lower line is the Einstein approximation, above it is the expression of Batchelor and Green, even higher is the Padé approximant of the Einstein approximation, the uppermost line is the Padé approximant of the Batchelor-Green approximation

3.4. Probability distributions

The most frequent (as is usually considered) universal distribution laws of random variables found in various natural science studies are the normal law – the Gaussian distribution and the so-called lognormal distribution (Fig. 3.4.1):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}, \quad (3.4.1)$$

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{\ln x^2}{2\sigma^2}}, x > 0 \qquad (3.4.2)$$

The frequent occurrence of the normal law is explained by the fact that when the distribution of a random variable is associated with the sum of independent processes, the distribution approaches normal. It is this statement that is the content of the central limit theorem of probability theory. Note that often in specific studies, the Gaussian distribution of a random variable is taken as a matter of habit or convenience.

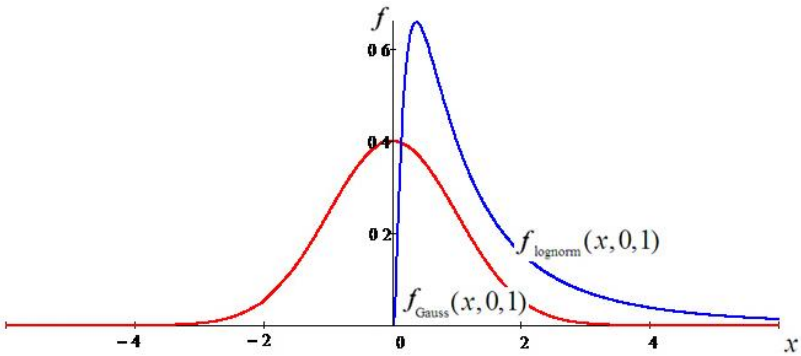


Figure. 3.4.1 – Graphs of normal and lognormal distribution. The mean for the normal distribution is chosen to be zero

B. Mandelbrot was one of the first who paid close attention to the fact that no less universal, often encountered distribution law of a random variable is a power (often called hyperbolic) distribution with a probability density:

$$f\ x = \frac{B}{x^{\beta}}, \qquad (3.4.3)$$

or

$$P\ X \geq x = \frac{A}{x^{\alpha}}, \quad x > 0, \quad \alpha = \beta - 1, \qquad (3.4.4)$$

where $P\ X \geq x$ is the probability that $X \geq x$, and A and α are some positive constants, the distribution parameters.

Respectively,

$$P\ X \geq x = \int_{x}^{\infty} \frac{B}{x^{\beta}} dx, \qquad \frac{dP\ x}{dx} = -f\ x\ . \qquad (3.4.5)$$

It should be noted that the above distribution was considered by Mandelbrot as a refinement of Zipf's law and is often referred to as the Zipf-Mandelbrot distribution. At the same time, it turned out that α is a value close to unity, which can vary depending on the properties of the text and language.

In fairness, it should be noted that the power distribution functions were considered even by Cauchy. As a clear example of the Cauchy distribution, one can give a model of firing from a machine gun rotating at a constant angular velocity ω in a horizontal plane (Fig. 3.4.3).

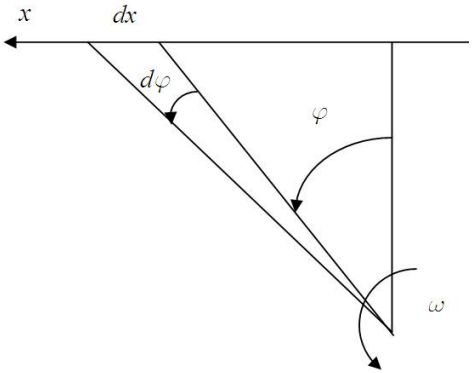


Figure 3.4.2. – Cauchy distribution example

If, when firing single shots, you press the trigger with equal probability at any of its positions, then the function of the distribution of shots along the angle will φ be a constant value: $F(\varphi) = const$. On the other hand, the probability of hitting an infinitely small area dx of an infinite flat target is equal to $f(x)\,dx = F(\varphi)\,d\varphi$. Whence, taking into account $x = a \cdot tg\,\varphi$, after elementary transformations we find the Cauchy distribution:

$$f(x) = \frac{1}{\pi}\frac{a}{a^2 + x^2}, \quad -\infty < x < \infty. \quad (3.4.6)$$

Since for this function the mean $\langle x^\alpha \rangle$ of x^α ($\langle x^\alpha \rangle = \int_{-\infty}^{\infty} x^\alpha f(x)dx$) is not defined for $\alpha \geq 1$, then neither the mathematical expectation (i.e., the mean of x^α), nor the variance of $\langle x^2 \rangle$, nor the higher-order moments of this distribution are defined. In this case, the expected value is said to be undefined and the variance is infinite.

Recall that a special case of a power-law distribution – the hyperbolic distribution A/x is named after V. Pareto, and the discrete distribution law with a ranged variable was named after

J. J. Zipf, who formulated it to describe the frequency of the use of words.

3.5. Scaling. Homogeneous functions

3.5.1. Homogeneous function of one variable

A homogeneous function can be defined like this:

$$f(\lambda x) = g(\lambda) f(x). \qquad (3.5.1)$$

Example:

$$f(x) = \alpha x^3. \qquad (3.5.2)$$

In this case

$$f(\lambda x) = \alpha(\lambda x)^3 = \alpha\lambda^3 x^3 = \lambda^3 \alpha x^3 = \lambda^3 f(x). \quad (3.5.3)$$

What is the "benefit" of a homogeneous function?

Turns out it's huge!

For example, suppose we know that $f(x)$ is a homogeneous function. Then if we know its value at one single point, then we can find out at any other:

Let us know in x_0 – $f(x_0)$, but we want to x. Let's denote $x = \lambda x_0$.

$$f(x) = f(\lambda x_0) = g(\lambda) f(x_0), \qquad (3.5.4)$$

where $\lambda = x / x_0$.

If $f(x)$ is continuous and differentiable, then it can be shown that the definition $\boxed{f(\lambda x) = g(\lambda) f(x)}$ implies $\boxed{g(\lambda) = \lambda^p}$ (up to a factor).

And thus, a homogeneous continuous and differentiable function is a power function.

Are there other (homogeneous) non-power functions for which $f(\lambda x) = g(\lambda) f(x)$?

Yes, they exist, but you have to give up smoothness.

For example, δ the Dirac function:

$$\delta(x)=\begin{cases}0,\ x\neq 0\\ \neq 0,\ x=0\end{cases}\quad;\quad \int_{-\infty}^{+\infty}\delta(x)dx=1. \tag{3.5.5}$$

3.5.2. Homogeneous functions of several variables

For a homogeneous function of several variables, we have:

$$f(\lambda x_1,\ \lambda x_2,\ ...,\ \lambda x_n)=g(\lambda)f(x_1,\ x_2,\ ...,\ x_n), \tag{3.5.6}$$

in particular, for a function of two variables

$$f(\lambda x,\ \lambda y)=g(\lambda)f(x,y),\ g(\lambda)=\lambda^p. \tag{3.5.7}$$

One can extend the notion of a homogeneous function of several variables by introducing a generalized function of several variables. For example, for two variables, the generalized homogeneous function

$$f\left(\lambda^a x,\lambda^b y\right)=\lambda^p f\left(x,y\right), \tag{3.5.8}$$

which means that each variable has its own scale constant – λ^a for x and λ^b For y.

Again the question arises, what is the "benefit" from the fact that the function of, for example, two variables is homogeneous? It turns out that generalizing a homogeneous function to several variables leads to much more important questions than for a function of one variable.

There is no need to explain to programmers how important it is if a *two-dimensional* array can be replaced with *a one-dimensional one.*

The function $F\left(z\right)$ is called the scaling (or

Indeed, a homogeneous function of two variables

self-similar) function.

can be reduced to a function of one variable. Indeed, let $f(\lambda x, \lambda y) = \lambda^p f(x, y)$, then choose $\lambda = 1/y$, from where:

$$\lambda^p f(x,y) \Leftarrow f(\lambda x, \lambda y) \Rightarrow f\left(\frac{x}{y}, 1\right); \qquad \lambda^p = y^{-p}. \quad (3.5.9)$$

Hence,

$$y^{-p} f(x,y) = f\left(\frac{x}{y}, 1\right). \quad (3.5.10)$$

Denote $f\left(\frac{x}{y}, 1\right) = F\left(x/y\right)$, then finally:

$$f(x,y) = y^p F\left(x/y\right). \quad (3.5.11)$$

Thus, a function of two variables $f(x,y)$ is expressed in terms of a function of one $F\left(x/y\right)$.

Let's look at a few examples.

Planck's formula for thermal radiation:

$$u(\omega, T) = \frac{\hbar\omega^3}{\pi^2 c^3} \frac{1}{e^{\frac{\hbar\omega}{KT}} - 1}, \quad (3.5.12)$$

where $\hbar$ is the Planck constant, c is the speed of light, k is the Boltzmann constant, ω is the cyclic frequency, T is the absolute temperature.

We choose more convenient notation

$$f(x,y)=\frac{x^3}{e^{\frac{x}{y}}-1}, \qquad (3.5.13)$$

where $x=\hbar\omega$, $y=kT$, $f(x,y)=\pi^2\hbar^2c^3u(\omega,T)$.

The function $f(x,y)$ is a function of two variables x and y and to set it, you need a two-dimensional array, for example, a set of dependencies $f(x,y)$ as a function of x for different values y of Fig. 3.5.1.

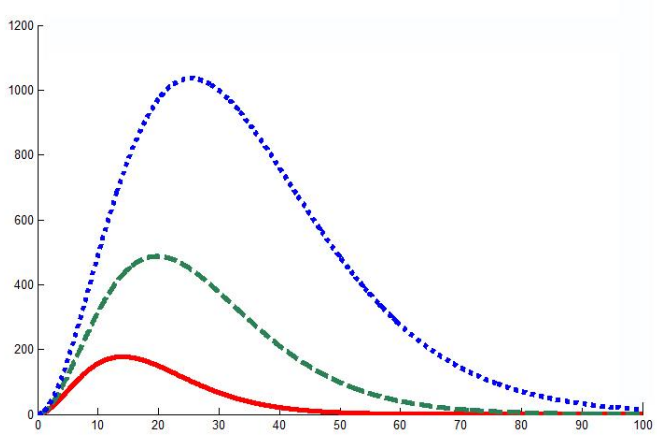


Figure. 3.5.1. Dependences of the function $f(x,y)$ on x at different values y: 5 (solid line), 7 (dashed line), 9 (dotted line)

On fig. 3.5.2 shows the dependence of the function $f(x,y)$ on y for different x.

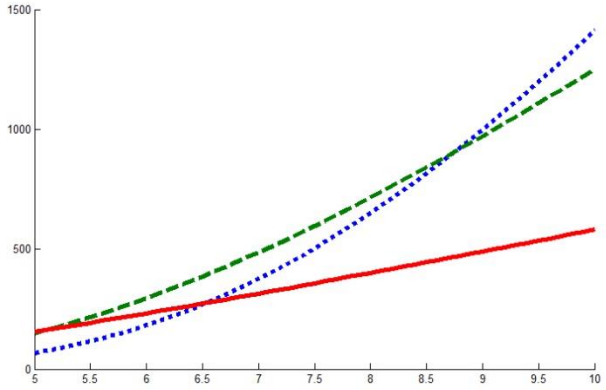


Figure. 3.5.2. Dependences of the function $f(x,y)$ on y at different values x: 10 (solid line), 20 (dashed line), 30 (dashed line)

A two-dimensional array – $f(x,y)$ can be depicted in the same way as a surface – Figure 3.5.3.

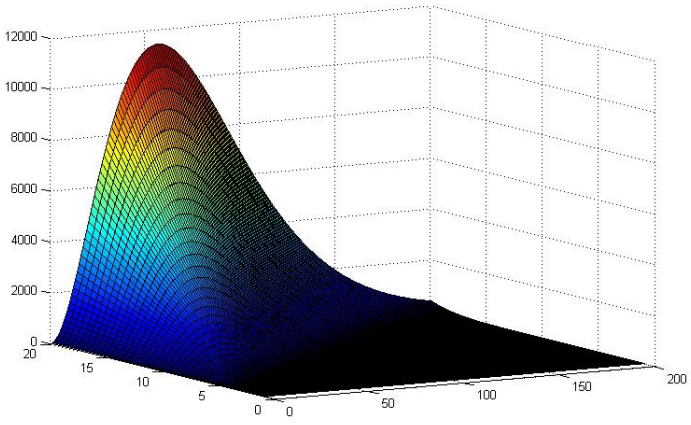


Figure. 3.5.3. Surface defined by the function $f(x,y)$

If the task was to describe the function $f(x,y)$ experimentally, then according to Fig. or fig. it would be

necessary to measure its value for different x and y, some values of which could be difficult to measure (for example, very high temperatures).

Before Planck wrote his formula for thermal radiation, Wien's law was known, according to which

$$u(\omega,T)=\omega^3 F\left(\frac{\omega}{T}\right), \qquad (3.5.14)$$

or in our notation

$$f(x,y)=x^3 F\left(\frac{x}{y}\right). \qquad (3.5.15)$$

Note that this is a typical scaling relation, which drastically simplifies the task of experimental determination of $f(x,y)$. Indeed, introducing the function

$$\varphi(x,y)=\frac{f(x,y)}{x^3}. \qquad (3.5.16)$$

We find that it is a function of one variable, and that a one-dimensional array is sufficient to define it (Fig. 3.5.4).

Another example comes from so-called economic physics.

Demand plays an important role in the economy.

demand function for a given product (or a group of similar products) depends on:

1) the quantity of goods *Q,* consumed per unit of time;

2) whether the consumer has money *U* (or income *D)*;

3) commodity prices – p

$$Q=Q(U,p),$$

Chernavsky D. S., Starkov N.I., Malkov C.Yu., Kosse Yu.V., Shcherbakov A.V. About econophysics and her place in modern theoretical economics, UFN, 2011, vol. 181, no. 7, p. 15

Barenblatt G.I. Similarity, self-similarity, intermediate asymptotics / 2nd ed., revised. and additional – Leningrad: Gidrometeoizdat, 1982. – 256 p.

g de U and p are conditional and are measured in different units ($, €, ¥, ...), so it seems plausible that is $Q(U,p)$ a homogeneous function of these variables.

Indeed, if we write $Q(U,p)=F\left(U/p\right)$, then as it turns out, the empirical data fit well on one curve.

This approach allows us to formulate a problem and obtain an analytical representation of the demand function and identify a very important parameter $r_{\min}$.

Closely related to the concept of scaling are the concepts of self-similarity (scaling), similarity, and intermediate asymptotics.

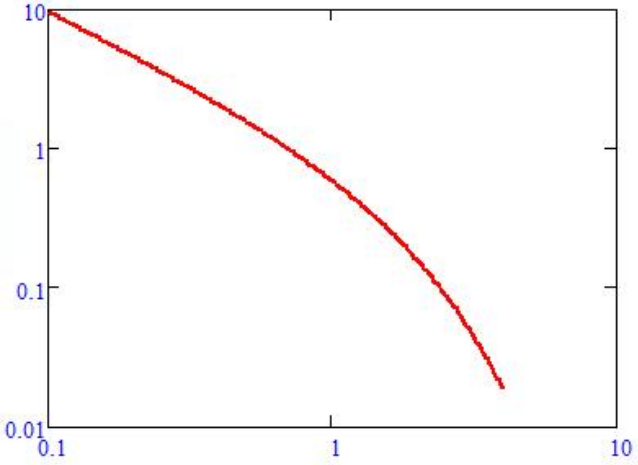


Figure 3.5.4 – $\varphi(x,y)$ as a function of one variable $z=x/y$

When processing experimental data, self-similarity led to the fact that a seemingly disordered cloud of experimental points in ordinary coordinates fell on a single curve or surface.

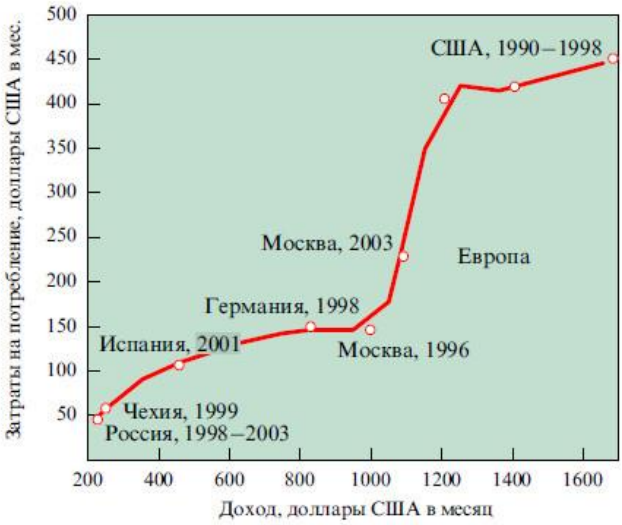


Figure. 3.5.5 - $r = \dfrac{U}{p}$ – purchasing power

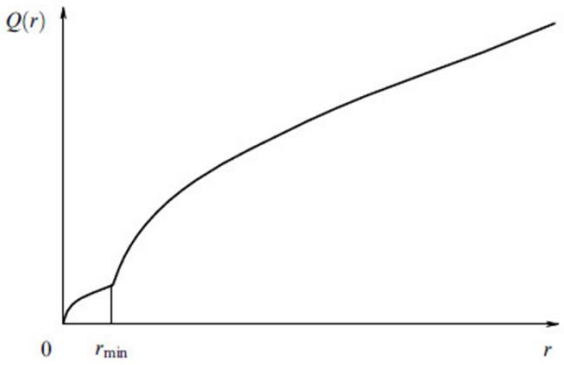


Figure. 6. $r < r_{\min}$ - the consumer does not buy durable goods (cars, TVs, cottages, elite goods), everything goes to food, clothing, housing and communal services

3. 6. Generating functions

The generating function $f(x)$ of an arbitrary infinite sequence $a_0, a_1, a_2, ..., a_k, ...$ is the expression

$$f(x) = \sum_{k=0}^{\infty} a_k x^k, \qquad (3.6.1)$$

Landau S.K. Lectures on generating functions. – M.: MTsNMO, 2007.

Wilf H.S. Generatingfunctionology. – Academic Press, 1994.

in this case, the variable x is formal and the sum of the series itself has no meaning.

Such a statement is quite unexpected (it can be confusing – what kind of function is this, a variable that does not make sense). The series can diverge and the only thing that is defined is the value at the point $x = 0$, i.e. $f(0) = a_0$.

Despite such surprising statements, the generating function is a powerful tool (method) that allows you to solve very complex problems in a simple elegant way. Here we will consider the method of generating functions for solving recurrence relations, which allows us to obtain many important results in the theory of complex networks.

The generating function turns a discrete set – a sequence a_k $(k = 0, 1, ...)$ into a continuous function, which allows you to "turn on" the mathematical apparatus that works with continuous objects – the analysis of continuous functions (solution of equations, differentiation, integration, etc.), after which you can again return to the "discrete" world.

The statement of the problem of recurrent relations looks like this: let there be a relation

$$x_{k+1} = f(x_k). \qquad (3.6.2)$$

it is necessary to find an explicit dependence

$$x_k = F(k). \qquad (3.6.3)$$

More general settings are also possible, for example, when the recurrence relation is given

$$x_{k+2} = f\left(x_k, x_{k+1}\right) \qquad (3.6.4)$$

Let's look at a few examples.

The first example, trivial, is for a simple demonstration of the method. Let there be an iterative relation $a_{k+1} = a_k + 1,\ k = 1,\ 2,\ ...,\ a_1 = 1$ and it is necessary to find the dependence a_k on k. We introduce a generating function:

$$G(x) = \sum_{k=1}^{\infty} a_k x^k, \qquad (3.6.4)$$

multiply it by x and carry out a number of elementary transformations:

$$\begin{aligned} &xG(x) = x\sum_{k=1}^{\infty} a_k x^k = \sum_{k=1}^{\infty} a_k x^{k+1} = \left(a_k = a_{k+1} - 1\right) = \\ &= \sum_{k=1}^{\infty} a_{k+1} x^{k+1} - \sum_{k=1}^{\infty} x^{k+1} = \sum_{n=2}^{\infty} a_n x^n - \sum_{n=2}^{\infty} x^n = \\ &= -a_1 x + \left(a_1 x + \sum_{n=2}^{\infty} a_n x^n\right) - \left(1 + x + \sum_{n=2}^{\infty} x^n\right) + 1 + x. \end{aligned} \qquad (3.6.5)$$

The first term in brackets is $G(x)$, the second $\sum_{n=0}^{\infty} x^n = 1/(1-x)$ is, and, given that $a_1 = 1$, we find:

$$xG(x) = -x + G(x) - \frac{1}{1-x} + 1 + x, \qquad (3.6.6)$$

whence follows the expression for the generating function:

$$G(x) = \frac{x}{(1-x)^2}. \qquad (3.6.7)$$

Expanding the resulting expression $G(x)$ into a Taylor series, we find:

$$G(x) = \frac{x}{(1-x)^2} = x\left(1 + 2x + 3x^2 + ...\right) = \sum_{k=1}^{\infty} k \cdot x^k. \quad (3.6.8)$$

Since $G(x)=\sum_{k=1}^{\infty} a_k x^k$, for a_k we get :

$$a_k = k, \qquad (3.6.9)$$

and, thus, the desired dependence is found.

Of course, in the considered simple example, from the recurrence relation, $a_{k+1} = a_k + 1$ one can immediately guess that $a_1 = 1$ we are talking about a sequence 1, 2, 3, ..., , i.e., that $a_k = k$.

However, the second and third examples demonstrate cases where the guess is unlikely, and then the generating function method turns out to be practically the only way out.

The second example, a standard example of a demonstration of the generating function method is the Fibonacci numbers. Fibonacci considered an idealized population of rabbits, when initially there is a newborn pair, which, from the second month after its birth, begins to mate and every month produce a new pair that behaves in exactly the same way. Question: if the rabbits do not die, then how many will there be in n months.

The process is described by the iterative relation

$$F_{n+1} = F_n + F_{n-1}, \quad n \geq 1 \ , \ F_0 = 0, \ F_1 = 1, \qquad (3.6.10)$$

where F_n is the number of newborn couples in n the th month.

We introduce a generating function:

$$G(x)=\sum_{n=0}^{\infty} F_n x^n \ . \qquad (3.6.11)$$

Multiply the iterative relation by x^n and sum:

$$\sum_{n=1}^{\infty} F_{n+1} x^n = \sum_{n=1}^{\infty} F_n x^n + \sum_{n=1}^{\infty} F_{n-1} x^n, \qquad (3.6.12)$$

note that the summation can only be started from $n=1$, since for $n=0$ each sum, an indefinite term appears on the right F_{-1}.

For the left side of expression (3.6.12) we have:

$$\begin{aligned}&\sum_{n=1}^{\infty}F_{n+1}x^n=\frac{1}{x}\sum_{n=1}^{\infty}x^{n+1}F_{n+1}=\\&=\frac{1}{x}\left(1+xF_1+x^2F_2+\ldots-1-F_1x\right)=\\&=\frac{1}{x}\left(\sum_{n=0}^{\infty}F_1x^n-\left(1+x\right)\right)=\\&=\frac{1}{x}\left(G\left(x\right)-\left(1+x\right)\right),\end{aligned}\qquad(3.6.13)$$

where it was taken into account that $F_1=1$.

For the sum $\sum_{n=1}^{\infty}F_nx^n$ it is easy to get:

$$\sum_{n=1}^{\infty}F_nx^n=\sum_{n=0}^{\infty}F_nx^n-1=G(x)-1.\qquad(3.6.14)$$

And finally, for the last sum we have:

$$\begin{aligned}&\sum_{n=1}^{\infty}F_{n-1}x^n=x\sum_{n=1}^{\infty}F_{n-1}x^{n-1}=\\&=x\sum_{n=0}^{\infty}F_nx^n=xG\left(x\right).\end{aligned}\qquad(3.6.15)$$

Thus, the iteration relation can be written as:

$$\frac{G(x)}{x}-\frac{1+x}{x}=G(x)(1-x)-1,\qquad(3.6.16)$$

which gives the following expression for the generating function:

$$G(\mathrm{x})=\frac{x}{1-x-x^2}.\qquad(3.6.17)$$

Expanding $G\left(x\right)$ into simple fractions, we find:

$$\frac{x}{1-x-x^2}=\frac{1}{a-b}\left(\frac{1}{1-ax}-\frac{1}{1-bx}\right)=$$
$$=\frac{1}{a-b}\left(\sum_{k=0}^{\infty}a^k x^k-\sum_{k=0}^{\infty}b^k x^k\right), \qquad (3.6.18)$$

where $b=\frac{1-\sqrt{5}}{2}$, a $a=\frac{1+\sqrt{5}}{2}$ is the golden ratio.

Thus, for $G(x)$ we find:

$$G(x)=\sum_{k=0}^{\infty}\frac{a^n-b^n}{\sqrt{5}}x^n=$$
$$=\sum_{k=0}^{\infty}\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n-\left(\frac{1-\sqrt{5}}{2}\right)^n\right]x^n, \qquad (3.6.19)$$

which for the desired F_n gives:

$$F_n=\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n-\left(\frac{1-\sqrt{5}}{2}\right)^n\right]. \qquad (3.6.20)$$

The last example concerns the study of deterministic SF - networks, namely, (u,v) colors. Earlier it was stated without derivation that from the iterative relation

$$N_n=wN_{n-1}-w,\ n=1,\ 2,\ \ldots,\ N_1=w, \qquad (3.6.20)$$

with follow :

$$N_n=\frac{w-2}{w-1}\cdot w^n+\frac{w}{w-1}. \qquad (3.6.21)$$

Let us show how we can obtain $N_n=f(n)$ by the generating function method

$$G(x)=\sum_{n=2}^{\infty}N_n w^n\ . \qquad (3.6.22)$$

Making obvious transformations

$$G(x) = N_1 w^1 + \sum_{n=2}^{\infty} N_n x^n = wx + \sum_{n=2}^{\infty} \left(wN_{n-1} - w \right) x^n =$$
$$= wx + wx \sum_{n=1}^{\infty} \left(wN_n - w \right) x^n - w \left(\sum_{n=0}^{\infty} x^n - (1+x) \right) = \qquad (3.6.23)$$
$$= wx + wxG(x) - w \left(\frac{1}{1-x} - (1+x) \right),$$

for $G(x)$ we get the following expression :

$$G(x) = w \cdot \frac{x - 2x^2}{(1-wx)(1-x)}, \qquad (3.6.24)$$

Because the

$$\frac{1}{(1-wx)(1-x)} = \frac{1}{1-w} \left(\frac{1}{1-x} - \frac{w}{1-wx} \right), \qquad (3.6.25)$$

The generating function can be written as a sum of fractions:

$$G(\mathrm{x}) = \frac{w}{1-w} \left(\frac{x}{1-x} - \frac{2x^2}{1-x} - \frac{wx}{1-wx} + \frac{2x^2}{1-wx} \right), \qquad (3.6.26)$$

each of which, in turn, can be written as a sum:

$$\frac{x}{1-x} = x \sum_{n=0}^{\infty} x^n = \sum_{n=0}^{\infty} x^{n+1} = \sum_{n=1}^{\infty} x^n, \qquad (3.6.27)$$

$$\frac{\alpha x}{1-wx} = \sum_{n=1}^{\infty} w^n x^n, \qquad (3.6.28)$$

$$\frac{x^2}{1-x} = x^2 \sum_{n=0}^{\infty} x^n = x^2 \left(1 + x + x^2 + \ldots \right) =$$
$$= x^2 + x^3 + \ldots + x - x = \sum_{n=1}^{\infty} x^n - x, \qquad (3.6.29)$$

$$\frac{wx^2}{1-wx}=\frac{1}{w}\sum_{n=1}^{\infty}w^n x^n - x, \qquad (3.6.30)$$

substituting which into the expression for $G(x)$ (free " x" appearing in place of the second and fourth terms are reduced) we find:

$$G(x)=\frac{w}{1-w}\sum_{n=1}^{\infty}\left(\frac{2-w}{w}w^k-1\right)x^n, \qquad (3.6.31)$$

whence follows the expression for N_n.

The given examples show that the task of obtaining an explicit dependence on the step number from the iterative relation is somewhat similar to encryption-decryption – it is difficult to solve, but easy to encrypt. Indeed, having an explicit expression, for example, $N_n=\left(\frac{w-2}{w-1}\right)w^n+\frac{w}{w-1}$ it is very easy to make sure that it satisfies the iterative relation $N_n=wN_{n-1}-w$. The inverse problem is much more complicated, and here the method of generating functions is of great benefit. However, when using this method, there is no clear algorithm. What needs to be done – multiply by x, x^2, ..., divide, split the series into parts, etc. Each specific task has its own action. However, in the general (not completely specific) description of actions, there are several clear, sequential steps, the first is to rewrite (transform) the generating function, taking into account the given iteration relation, so as to obtain an equation for the generating function with respect to the introduced formal variable. After solving this equation, the resulting function of x must be expanded into a series, writing out in an explicit mathematical form the expansion coefficients at the powers of the formal variable, which gives the solution.

In some cases, this path is quite cumbersome. This is where other types of generating functions can come in handy. Until now, the generating function has been understood as a power series

$$G(x)=\sum a_n x^n, \qquad (3.6.32)$$

it turns out that sometimes it is more convenient to use the Dirichlet generating function based on the Riemann zeta function $\zeta(x)=1+\frac{1}{2^x}+\frac{1}{3^x}+...$

$$G(x)=\frac{a_1}{1^x}+\frac{a}{2^x}+...=\sum_{n=1}^{\infty}\frac{a_n}{n^x}, \qquad (3.6.33)$$

or on the exponential generating function

$$G(x)=\sum_n a_n\frac{x^n}{n!}. \qquad (3.6.34)$$

In conclusion, we note that there are also generating functions of several variables.

3.7. Dirac delta function

The delta function (δ-function) was introduced by the English physicist P. A. M. Dirac "out of necessity" when he created the mathematical apparatus of quantum mechanics. Mathematicians "did not recognize" it for some time, after which they created the theory of generalized functions, of which the δ-function is a special case.

According to the (naive) definition, the δ-function is equal to zero everywhere except at one point, but the area covered by this function is equal to one:

$$\delta(x)=\begin{cases}0, & x\neq 0\\ \infty, & x=0\end{cases}, \qquad \int_{-\infty}^{+\infty}\delta(x)dx=1. \qquad (3.7.1)$$

These conflicting requirements cannot be met by a "regular" type function.

Zeldovich Ya.B. Higher mathematics for beginner physicists and technicians. –M.: Nauka, 1982.

In fact, just as the differential dx is not a number (equal to zero), but the phrase "an infinitesimal value" is difficult to understand qualitatively, it is correct to understand dx not as a number, but as a limit (process), so the δ-function is correctly understood as a limit (process). On fig. 3.7.1 and 3.7.2 show several functions (depending on the parameter), the limit of which is the δ-

function. There are infinitely many such functions – everyone can choose their own.

The δ-function has many useful properties, being, in particular, the continual analogue of the Kronecker symbol δ_{ik}

$$\int_{-\infty}^{+\infty} f(x)\delta(x-x_0)dx = f(x_0). \qquad (3.7.2)$$

compare with

$$\sum_i f_i\delta_{ik} = f_k\,. \qquad (3.7.3)$$

Another surprising relation indicates how you can differentiate by integrating:

$$\int_{-\infty}^{+\infty} f(x)\delta'(x-x_0)dx = f'(x_0), \qquad (3.7.4)$$

Where δ' is the derivative of δ- the function.

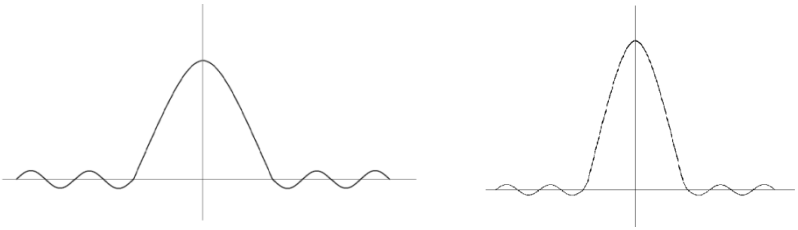


Figure. 3.7.1 – Two successive approximations to the Dirac δ-function. The function is shown $R(x,\omega) = \frac{1}{\pi}\frac{\sin(\omega x)}{x}$

a) $\omega = 10$, b) $\omega = 20$

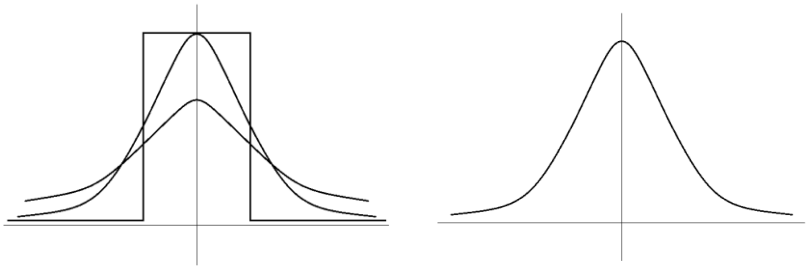


Figure. 3.7.2 – Two functions that $a \to \infty$ give δ-functions in the limit:

a) $R(x,a) = \dfrac{a}{2(1+a^2x^2)^{3/2}}$, b) $R(x,a) = \dfrac{a}{\pi(1+a^2x^2)}$.

Finally, note that the interval from the δ-function:

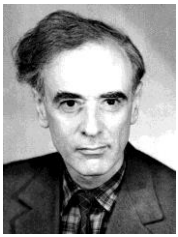
$$\int_{-\infty}^{x} \delta(x)dx = \theta(x), \qquad (3.7.5)$$

where $\theta(x)$ is the Heaviside function,

$$\theta(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0 \end{cases}, \qquad (3.7.6)$$

step, with a break at the point $x = 0$.

3. 8. Phase transitions



Lev Davidovich

Landau (1908-1968)

In order to talk about phase transitions, it is necessary to define what phases are. The concept of phases occurs in many phenomena, therefore, instead of giving a general definition (the more general it is, the more abstract and non-visible, as it should be), we will give a few examples.

First, an example of their physics. For the usual, most common liquid in our life – water, three phases are known: liquid, solid (ice) and gaseous (steam). Each of them is characterized by its own parameter values. It is essential that when external conditions change, one phase (ice) passes into another (liquid). Another favorite object of theorists is a ferromagnet (iron, nickel and many other pure metals and alloys). At low temperatures (below for nickel $T_c = 360^0 C$), a nickel sample is a ferromagnet; when the external magnetic field is removed, it remains magnetized, i.e. can be used as a permanent magnet. At temperatures above T_c this property is lost, when the external magnetic field is turned off, it goes into a paramagnetic state and is not a permanent magnet. When the temperature changes, a transition occurs – a phase transition – from one phase to another.

Let us give one more geometric example from the theory of percolation. Randomly cutting out connections from the grid, in the end, when the concentration of the remaining connections p becomes less than a certain value p_c, it will no longer be possible to pass through the lattice “from one end to the other”. Thus, the grid from the state of percolation – the phase of "percolation", will pass into the state of the phase of "non-percolation".

From these examples, it is clear that for each of the considered systems there is a so-called order parameter that determines which of the phases the system is in. In ferromagnetism, the order parameter is the magnetization in a zero external field; in the theory of percolation, it is the network connectivity, or, for example, its conductivity or the density of an infinite cluster.

Phase transitions are of various kinds. Phase transitions of the first kind are such a transition when several phases can simultaneously exist in the system. For example, at a

temperature 0° C ice floats in water. If the system is in thermodynamic equilibrium (no heat supply and removal), then the ice does not melt and does not grow. For phase transitions of the second kind, the existence of several phases simultaneously is impossible. A piece of nickel is either in a paramagnetic state or in a ferromagnetic state. A mesh with randomly cut connections is either connected or not.

Decisive in the creation of the theory of phase transitions of the second kind, the beginning of which was laid by L.D. Landau, was the introduction of the order parameter (we will denote it η) as a distinctive feature of the phase of the system. In one of the phases, for example, paramagnetic, $\eta = 0$, and in the other, ferromagnetic, $\eta \neq 0$. For magnetic phenomena, the order parameter η is the magnetization of the system.

To describe phase transitions, a certain function of the parameters that determine the state of the system is introduced – $G(\eta, T, ...)$. In physical systems, this is the Gibbs energy. In each phenomenon (percolation, a network of "small worlds", etc.), this function will be determined "independently". The main property of this function, the first assumption of L.D. Landau – in the state of equilibrium, this function takes the minimum value:

$$\frac{\partial G}{\partial \eta} = 0, \quad \frac{\partial^2 G}{\partial \eta^2} > 0. \qquad (3.8.1)$$

In physical systems one speaks of thermodynamic equilibrium, in the theory of complex chains one can speak of stability. Note that the minimality condition is determined by varying the order parameter.

The second assumption of L.D. Landau – during a phase transformation $\eta = 0$. According to this assumption, the function $G(\eta, T, ...)$ near the phase transition point can be expanded in a series in powers of the order parameter η:

$$G(\eta, T) = G_0(T) + A(T)\eta^2 + B\eta^4 + ..., \qquad (3.8.2)$$

where $\eta = 0$ in one phase (paramagnetic, if we are talking about magnetism and incoherent, if we are talking about a grid) and $\eta \neq 0$ in the other (ferromagnetic or connected).

From the condition $\partial G / \partial \eta = 0$ we find:

$$2A\eta + B\eta^3 = 0, \qquad (3.8.3)$$

which gives us two solutions $\eta = 0$ and $\eta^2 = -A/2B \neq 0$.

For $T > T_c$ there must be a solution $\eta = 0$, and for $T < T_c$ a solution $\eta \neq 0$. This can be satisfied if for the case $T > T_c$ and $\eta = 0$ choose $A > 0$. In this case, there is no second root. And for the case, $T < T_c$ the second solution must take place, i.e. must be performed $A < 0$. Thus:

$$A > 0 \text{ at } T > T_c, \; A < 0 \text{ at } T < T_c,$$

Landau's second assumption needs to be fulfilled $A(T_c) = 0$. The simplest form of the function $A(T)$ that satisfies these requirements is

$$A = \alpha \left(T - T_c \right) . \qquad (3.8.4)$$

Then

$$\eta^2 = -A/2B \sim \alpha \left(T - T_c \right) , \qquad (3.8.5)$$

where

$\eta \sim \sqrt{T_c - T}$, or $\eta \sim (T_c - T)^{\beta}$, $T < T_c$, where is $\beta = 1/2$ the so-called critical index, and the function $G(\eta, T)$ takes the form:

$$G\left(\eta, T\right) = G_0\left(T\right) + \alpha \left(T - T_c \right) \eta^2 + B\eta^4 + \ldots \qquad (3.8.6)$$

On fig. 3.8.1 shows the dependence $G(\eta, T)$ for $T > T_c$ and $T < T_c$.

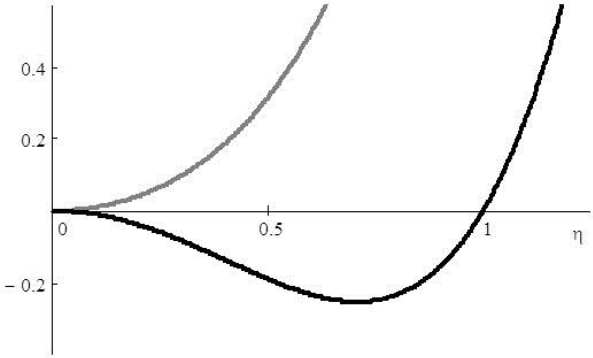


Figure. 3.8.1 – Graphs of the function of parameters $G(\eta,T)$ for $T > T_c$ and $T < T_c$

Qualitative dependence of parameters on $G(\eta,T)$ the order η parameter shown in fig. 3.8.1 $(G_0 = 0)$. The dependence of the order parameter η on temperature is shown in Fig. 3.8.2.

Poston T, Stewart I. Catastrophe theory and its applications. – M.: Mir, 1980.

Gilmore R. Applied Catastrophe Theory. – M.: Mir, 1984.

A more advanced theory takes into account that when $T > T_c$ the order parameter η, although very small, is not exactly zero.

The transition of the system from a state with $\eta = 0$ at $T > T_c$ to a state with $\eta \geq 0$ when decreasing T and reaching values $T \leq T_c$ can be understood as a loss of position stability $\eta = 0$ at $T \leq T_c$. Recently, a mathematical theory has appeared with the sonorous name "The Catastrophe Theory", which describes many different phenomena from a single point of view. From the point of view of the theory of catastrophes, the phase transition of the second kind is the “assembly catastrophe”.

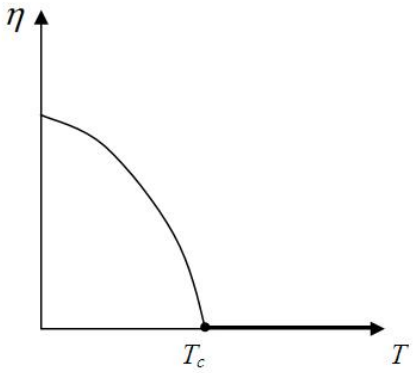


Figure. 3.8.2 – Dependence of the order parameter η on temperature: at $T < T_c$ and near T_c the order parameter η behaves like a power function, and when $T > T_c$ $\eta = 0$

3.9. Cellular automata

J. Fon. Neumann (1903-1957)

of cellular automata was first proposed more than half a century ago by J. Von Neumann and developed by S. Wolfram (S. Wolfram) in the fundamental monograph "A New Kind of Science".

Cellular automata are useful discrete models for the study of dynamical systems. The discreteness of the model, or rather, the ability to represent the model in a discrete form, can be considered an important advantage, since it opens up wide possibilities for using computer technologies.

For a long time, cellular automata were perceived as a fun game that has no practical value. But in recent years, in connection with the development of computer technology, they are beginning to quickly enter the arsenal of tools that are used in practice in various fields of science and technology.

Neumann J. Theory of self-reproducing automata. – M.: Mir, 1971. – 382 p.

A cellular automaton is a discrete dynamic system, a collection of identical cells connected in the same way to each other. All cells form a network (lattice) of cellular automata. The state of each cell is determined by the state of the cells included in its local neighborhood and are called nearest neighbors. The neighborhood of the cell with the number j - $O(j)$ is the set of its nearest neighbors. The state j of the -th cell at the moment of time $t+1$ is defined by some rule F that can be expressed, for example, in the language of Boolean algebra:

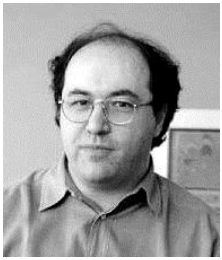
$$y_j(t+1) = F\left(y_j, O(j), t\right). \quad (3.9.1)$$

In many problems, it is considered that the cell itself belongs to its nearest neighbors, i.e. $y_j \in O(j)$, in this case the formula is simplified: $y_j(t+1) = F\left(O(j), t\right)$. Cellular automata satisfy the following rules:

- the change in the values of all cells occurs simultaneously (the unit of measurement is a cycle);
- the network of cellular automata is homogeneous, i.e. the rules for changing states are the same for all cells;
- a cell can be affected only by cells from its local neighborhood;
- the set of cell states is finite.

Cellular automata can have any dimension, but one-dimensional and two-dimensional systems of cellular automata are most often considered.

In the case of a two-dimensional lattice, the elements of which are squares, it is convenient to specify each cell with two indices – $y_{i,j}$. The nearest neighbors included in the neighborhood of the element $y_{i,j}$ are cells located up and down and to the left and right of it (the so-called von Neumann



S. Wolfram

Wolfram S. A New Kind of Science. – Champaign, IL: Wolfram Media Inc., 2002.

Wolfram S. ed. Theory and Applications of Cellular Automats. – Singapore: World Scientific. 1986.

Toffoli T., Margolus N. Machines of cellular automata. – M.: Mir, 1991.

neighborhood: $O^N(y_{i,j}) = \left(y_{i-1,j}, y_{i,j-1}, y_{i,j}, y_{i,j+1}, y_{i+1,j}\right)$, you can add diagonal elements – the neighborhood mura (G. _ Moore): $O^M(y_{i,j}) =$

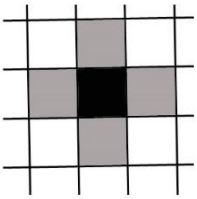
$$= \left(y_{i-1,j-1}, y_{i-1,j}, y_{i-1,j+1}, y_{i,j-1}, y_{i,j}, y_{i,j+1}, y_{i+1,j-1}, y_{i+1,j}, y_{i+1,j+1}\right).$$

In the Moore model, each cell has eight neighbors. To eliminate edge effects, the lattice is topologically "folded into a torus", i.e. the first line is considered a continuation of the last one, and the last line is considered to be the previous one – the so-called boundary conditions.

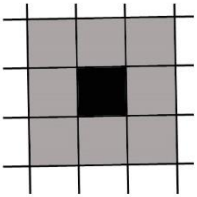
This allows you to determine the overall ratio of the value of the cell at the step $t+1$ compared to step t:

$$y_{i,j}(t+1) = (y_{i-1,j-1}(t), y_{i-1,j}(t), y_{i-1,j+1}(t), y_{i,j-1}(t), y_{i,j}(t), y_{i,j+1}(t), y_{i+1,j-1}(t), y_{i+1,j}(t), y_{i+1,j+1}(t)). \quad (3.9.1)$$

Neighborhood of Neumann:



Mur area:



Let's consider one of the examples of using cellular automata – the innovation distribution model and its generalization – the news distribution model. The model of diffusion (distribution) of innovations functions according to the following rules: each individual who is able to accept an innovation corresponds to one square cell on a two-dimensional plane.

In this case: 1) each cell can be in two states : 1 – the novelty is accepted; 0 – novelty not accepted. 2) the machine, having perceived the innovation once, remembers it forever (state 1, which cannot be changed). 3) the automaton approves the decision regarding the adoption of the novelty, focusing on the opinion of the eight nearest neighbors, i.e. if there are m adherents of the novelty in

Bhargava S.C., Kumar A., Mukherjee A. A stochastic cellular automata model of innovation diffusion. Technological forecasting and social change, 1993. – Vol. 44. – No. 1. – pp. 87-97.

the neighborhood of a given cell p (the Moore neighborhood is used), *is* the probability of its acceptance, (if $pm > R$ (R *is* a fixed value is the threshold), then the cell accepts the innovation (value 1). Cellular modeling allows you to build much more realistic models of the innovation market than traditional approaches.

In the model of information diffusion, it was assumed that a cell can be in one of three states: 1 – "breaking news" (the cell is painted black); 2 – news that is outdated, but saved as information (gray cell); 3 – the cell does not have information transmitted by the news message (the cell is white, the information has not reached or has already been forgotten). The following message propagation rules are adopted in the model:

- first, the entire field consists of white cells, with the exception of one, black, which was the first to "accept" the news (Fig. 3.9.1 a);
- a white cell can only be repainted black or remain white (it can receive news or remain "in the dark");
- the white cell is repainted if a condition similar to the innovation diffusion model is met: $pm > 1$ (this condition is modified for $m \le 2$: $1.5 \cdot pm > 1$) ;
- if the cell is black, and around it are exclusively black and gray, then it is repainted in gray colors (the news becomes outdated, but is saved as information);
- if the cell is gray, and around it are exclusively gray and black, then it is repainted in white (the aging of the news occurs when it is well known).

The described system of cellular automata qualitatively reflects the process of message propagation among individual information sources. It turned out that the state of the system of cellular automata is completely stabilized in a limited number of moves, i.e. the process of evolution turned out to be convergent. An example of the model operation is shown in fig. 3.9.1.

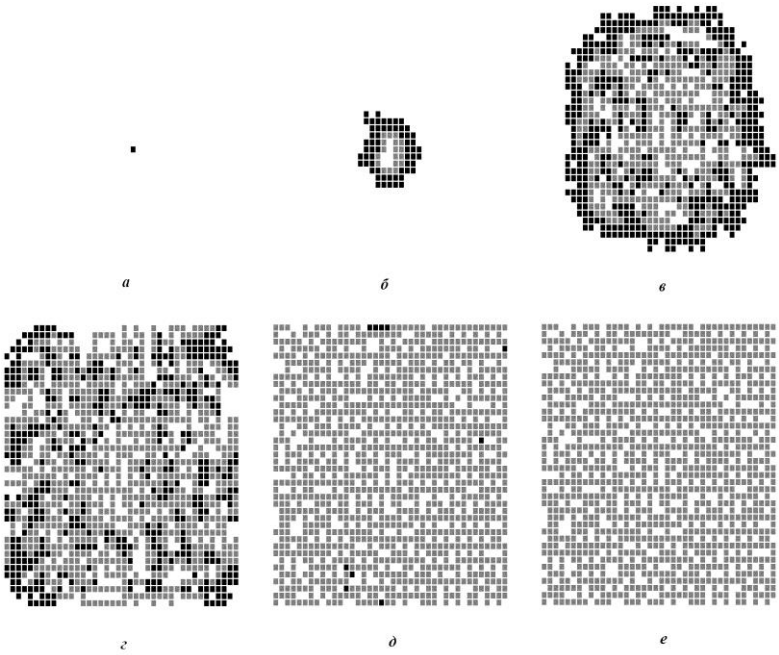


Figure 3.9.1 – The process of evolution of the system of cellular automata "diffusion of news": a – initial state; b-e – intermediate states; e – final state

Typical dependences of the number of cells that are in different states depending on the iteration step are shown in Fig. 3.9.2. In this case, obviously, the total number of cells that are in all three states at each iteration step is constant and equal to the number of cells, and when the system of cellular automata stabilizes, the ratio of gray, white and black cells is approximately: 3:1:0.

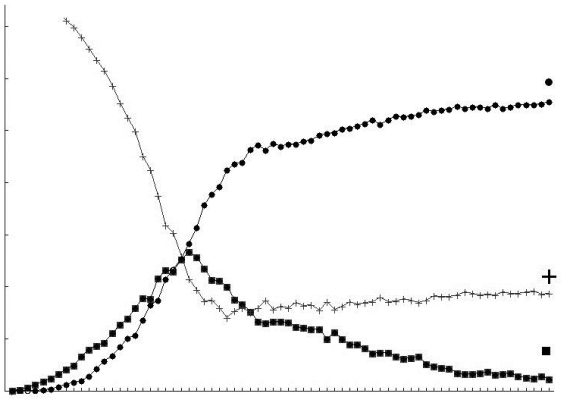


Figure 3.9.2 – The number of cells of each color depending on the step of evolution: white cells – (+); gray cells – (•); black cells – (■)

A detailed analysis of the dependencies obtained made it possible to draw analogies of this model of "information diffusion" with some analytical considerations. The simulation results suggest that the evolution of gray cells is described by some continuous function:

$$x_g = f\left(t, \tau_g, \gamma_g\right), \quad (3.9.3)$$

where t is the time (evolution step), τ_g is the time shift providing the required fragment of the analytical function, γ_g and is the steepness parameter of this function.

Accordingly, the dynamics of white cells x_w (the number of cells at the time t) can be modeled by an "inverted" function x_g with similar parameters:

$$x_w = 1 - f\left(t, \tau_w, \gamma_w\right). \quad (3.9.4)$$

Since, as mentioned above, the balance condition is always satisfied, i.e. the total number of cells at any time is always constant, then the normalization condition can be written as follows:

$$x_g + x_w + x_b = 1, \quad (3.9.5)$$

where x_b is the number of black cells at time t.

Thus, we get:

$$x_b = 1 - x_g - x_w = f\left(t, \tau_w, \gamma_w\right) - f\left(t, \tau_g, \gamma_g\right). \quad (3.9.6)$$

The view shown in fig. 5 5 dependencies suggests that as a function $f\left(t, \tau, \gamma\right)$ the following expression (logistic function) can be chosen:

$$f\left(t, \tau, \gamma\right) = \frac{C}{1 + e^{\gamma(t-\tau)}}, \quad (3.9.7)$$

where C is some normalizing constant.

On fig. 3.9.3 shows graphs of dependence x_g, x_w, x_b on the step of evolution of a system of cellular automata, obtained as a result of analytical modeling.

It should be noted that the dependence of news diffusion obtained as a result of modeling is in good agreement with the "life-like" behavior of thematic information flows on Internet sources (websites), and on local time intervals – with traditional models.

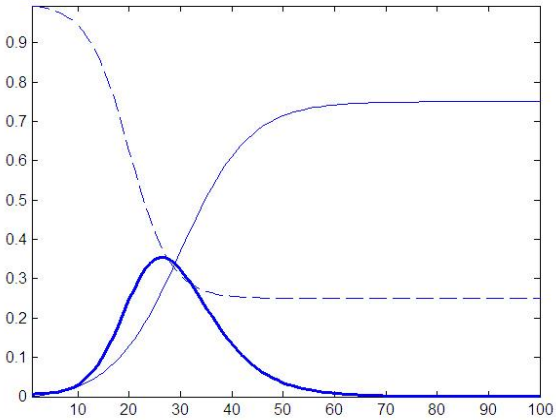


Figure 3.9.3 – Continuous dependencies obtained as a result of analytical modeling, depending on the step of evolution: solid line – gray (x_g); dotted line – white (x_w); solid thick line – black (x_b)

3.10. Self-organized criticality

Per Tank

(1948 – 2002)

Under the system that generates information, most often they mean a real social or economic system, from which one cannot expect simple predictable information or uniform behavior. In a real system, an information event can be considered in some sense as a catastrophe, since it is unexpected. If in most cases the prediction of an individual event seems impossible, then the behavior of the system as a whole, its response to an impact or disturbance is partially predictable and is the object of scientific research.

The term "self-organization" (self – organizing), associated With general theory systems, was introduced V. Ashby in 1947 And perceived by the then new cybernetics, its creators N. Winner, G. Forster, etc. Currently, this concept is most often associated with the name of P. Bak. In 1987-1988 P. Buck, C. Tang and K. Wiesenfeld in his works [70, 71] for the first time described in detail a cellular automaton that brought the system to a statistically the same "critical" state, which they called the state of self-organized criticality. A typical strategy in physics is to reduce the number of degrees of freedom in the problem under study, for example, in mean field theory, where the environment acts on the remaining degree of freedom of the system as some mean field, leaving only one variable to study.

Bak P., Tang C., Wiesenfeld K. Self-organized criticality: An explanation of 1/f-noise. Phys. Rev. Lett. 1987. Vol. 59.-pp. 381-384.

Bak P., Tang C., Wiesenfeld K. Self-organized criticality. Phys. Rev. A., 1988. – Vol. 38. – No. 1. – pp. 364-374.

The most obvious model demonstrating self-organized criticality is a pile of sand, familiar to everyone since childhood. If the sand is dry, then no Easter cake can be built from it, everything crumbles right there. In childhood, few people thought about how this happens. No matter how high the pile

was, the angle of inclination of the shedding cone remained unchanged. This was once again proved in the experiment in the experiment by grown children, at the University of Chicago, under the guidance of H. Yager, they experimented with a real pile of sand.

The state of this heap can be called critical, since by applying a minimum perturbation, throwing one grain of sand from above, the surface of the heap will be out of balance, and an avalanche will come down. And after its descent, there will again be a smaller pile of sand, new falling grains of sand will complete the pile to the same critical slope, and a new thrown grain of sand will again cause an avalanche. The heap is always in a critical state – small perturbations cause a reaction that is unpredictable in size, and always self-organizes – maintains the angle of inclination of the surface (Fig. 3.10.1).

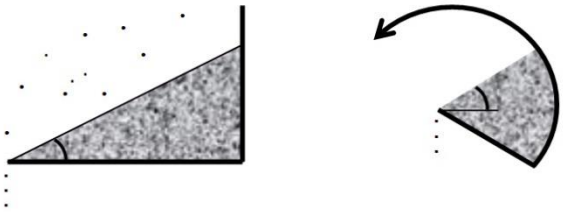


Figure. 3.10.1 – Box and rotating drum with sand with the same angle of inclination of the side plane

When modeling self-organized criticality, we study the statistics of avalanches, when one thrown grain of sand causes an avalanche from others lying on the surface.

Consider a discrete system analogous to a heap of sand in the one-dimensional case. Let $h(x)$ be the height of the heap at the point x $(x=1,\ 2,\ ...,\ N)$. It is convenient to depict a heap in two forms – the original one (Fig. 58 *a*) as a function of height from coordinates $h=h\ x$ and in the form of increments $z\ x\ =h\ x\ -h\ x+1$, which show the difference in heights at neighboring points – fig. 58 *b*. The left parts of Fig. 5 7 show the initial state of the heap.

Let's introduce rule 1: if the height difference at a point x is greater than a certain critical value $h\ x > h_c$, then extra grains of sand roll down to neighboring points. Choosing the critical value $h_c = 3$ rule 1 can be written as follows:

$$\left.\begin{array}{l} z(x) \to z(x) - 2, \\ z(x \pm 1) \to z(x \pm 1) + 1 \end{array}\right|_{z(x)>2} \qquad (3.10.1)$$

The first relation means that the height of the heap at the point x decreases by two grains of sand, the second that at neighboring points (left and right) the height will increase by one grain of sand.

On heap boundaries, boundary conditions 1 will be met:

$$\begin{array}{l} z(1) = 0, \\ \left.\begin{array}{l} z(N) \to z(N) - 1, \\ z(N-1) \to z(N-1) + 1 \end{array}\right|_{z(N)>2} \end{array} \qquad (3.10.2)$$

The first of conditions 1 can be called “closed”, since the particle will never go outside the system, in contrast to the “open” conditions on the other side, when the particle rolls out and falls.

On fig. 3.10.2 on the left is the initial state of the heap – height $h\ x$ and increment $z\ x$, on the right – after shedding. So, for example, when $x = 6$ the increment before shedding was equal to $z\ 6 = 3 > 2$. After shedding, two grains of sand from the position $x = 6$ move one at a time to the left $x = 5$ and to the right $x = 7$ - fig. 3.10.2 on the left.

On the one hand, rule *1* is a discrete non-linear diffusion equation and, on the other hand, it is a cellular automaton in which the state of cell *x* at time $t+1$ is determined by the state / of neighboring cells at the previous time t. Graphically, the action of rule 1 can be represented as it is done in Fig. 58.

It is obvious that a one-dimensional heap, when it collapses according to rule 1 and conditions 1 from a non-equilibrium state c $z\ x >> 2$, has one critical state $z\ x = 2$ for any *x*. In the one-dimensional case, any stable state is, in a certain sense, critical, since any small perturbation will cause it to pass through the entire system, and any decrease in the slope to $z\ x < 2$ at any point will stop it. This is very similar to other one-dimensional critical phenomena such as percolation. It

should also be noted that such a state in the one-dimensional model has no spatial structure.

Similarly to the one-dimensional P. Buck proposed a rule for a two-dimensional heap (rule 2 and conditions 2). In such a system, the action of rules 1 for each of the directions x and y, the critical value is traditionally chosen to be 3:

$$\left.\begin{aligned} &z(x,y) \to z(x,y) - 4, \\ &z(x,y \pm 1) \to z(x,y \pm 1) + 1, \\ &z(x \pm 1,y) \to z(x \pm 1,y) + 1 \end{aligned}\right|_{z(x,y)>3} \qquad \begin{matrix} (3.10.3) \\ \text{(Rule 2)} \end{matrix}$$

$$z(0,y) = z(x,0) = z(N+1,y) = z(x,N+1) = 0 \qquad \begin{matrix} (3.10.4) \\ \text{(Conditions 2)} \end{matrix}$$

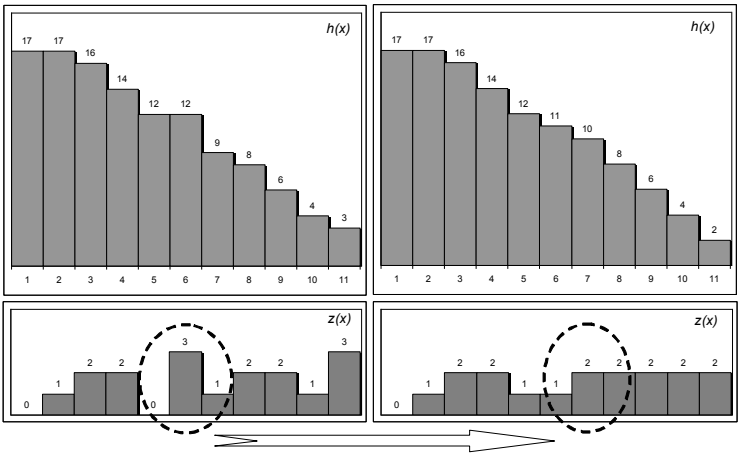


Figure. 3.10.4 – One-dimensional heap before and after shedding. Column 6 and the rightmost column 11 crumbled with "open" boundary conditions

A variant of "closed" boundary conditions 2 in all directions is indicated. Of course, any combination of "open" and "closed" conditions is possible. Condition 2 can also be modified for a "real" pile in the corner of, say, a shoe box. The lattices on which self-organizing critical systems were built are also diverse, for example, experiments were carried out on square lattices in large dimensions, and exact analytical results were even obtained on hexagonal lattices. Similar cellular automata can also be built on irregular lattices.

The correspondence between the value $z(x,y)$ and slope of the heap is not as clear-cut as in 1 D, since now the value of the slope $z(x,y)$ is the average slope along the diagonal of the system, and when shedding particles will start moving in both directions x and y. In the two-dimensional case, it is no longer possible to say that from an unstable state with $z(x,y) \gg 4$ the system will go into the same state, since the instability will propagate in both directions interdependently. The final state of the system will depend significantly on the initial conditions, but the properties of this resulting state, such as slope, will always be the same.

There are two different ways to get a system in a state of self-organized criticality. Either by shedding the system from a random state c $z(x,y) \gg 4$ to an equilibrium state, or by pouring $z\ x,y = 0$ grains of sand onto a flat surface one by one at randomly selected points and performing the procedure according to rule 2 when necessary. You can determine the moment when the system reaches a critical level by the fact that the average slope of the heap will stop changing. The experiment shows that the properties of the systems obtained by both methods do not differ from each other.

The results below are obtained on a square lattice in the two-dimensional case according to the previously defined rules and conditions. First, randomly selected $z\ x,y \gg 4$, after which the "relaxation" of the heap was carried out and it crumbled to a stable state. On fig. 59 shows one such stable state on a 2D 500 x 500 grid, where colors from black to white correspond to values $z\ x,y$ from 0 to 3.

If at one of the most unstable points of the system (in our case $z\ x,y = 3$), start the process according to *rule 2* with *conditions 2,* setting $z\ x,y = 4$, i.e. add one grain of sand, then the system will begin to crumble, an avalanche of sand will roll. For each such point $x,\ y$ in the system, the area affected by shedding will be different. On fig. 3.10.4 shows several such avalanches obtained by shedding. The initial system was shown in Fig. 3.10.3. Avalanche centers are marked with black dots against the background of white snow avalanches, and pairs of

numbers in brackets indicate the time of shedding and the size of the avalanche.

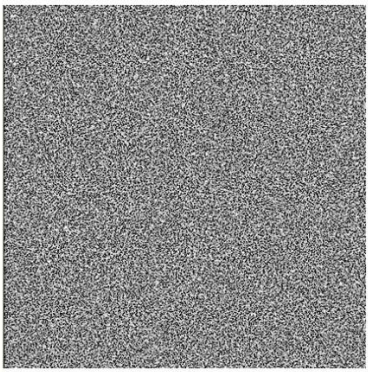


Figure. 3.10.3 – Steady state

Let us define $D(s)$ – the distribution function of the sizes of emerging avalanches. To obtain this function, at each point of the system, where $z\ x,y = 3$ we put $z\ x,y = 4$ and start the avalanche, we determine its area – S, to obtain a sufficient number of avalanches, we process several systems in a self-organized state in this way, obtaining them from a random initial state with $z\ x,y \gg 4$. On fig. 61 *a*) shows the form of dependence $D(s)$ obtained by processing a set of systems of size 500 x 500. The size distribution of avalanches obeys a power law:

$$D(s) \sim s^{-\tau}, \quad \tau_{2D} \approx 1{,}1. \qquad (3.10.5)$$

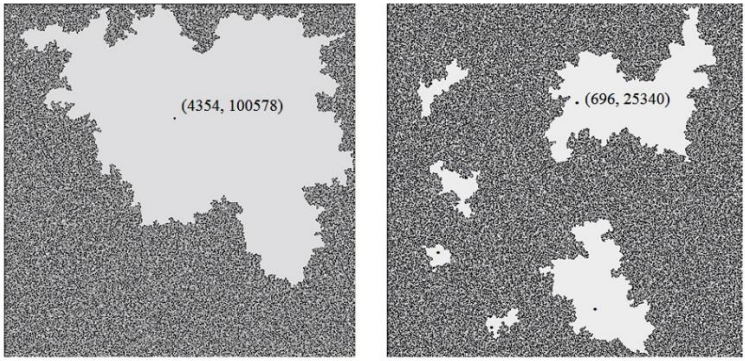


Figure. 3.10.4 – Avalanches caused by shedding

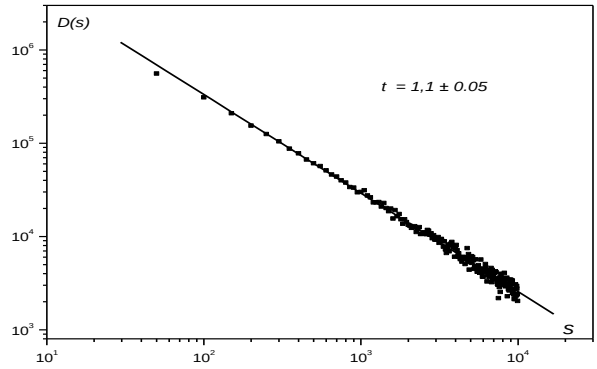
Similarly, it is possible to investigate the time characteristics of this process by introducing $D(t)$ - the distribution function of the times of shedding of these avalanches. In the general case, the area S of the avalanche is greater than its shedding time t, since several grains of sand are shed at one moment. The distribution of avalanche development also obeys a power law:

$$D(t) \sim t^{-\alpha}, \quad \alpha_{2D} \approx 0,43. \qquad (3.10.6)$$

Undoubtedly, the indices τ and α are related both to each other and to other indices characterizing the self-organized critical state.

The size of the "avalanche" of news that occurs in information flows when a new topic appears sometimes seems unpredictable, but it is quite amenable to modeling. For example, the power-law distributions of the number of thematic publications are quite consistent with the above distributions of the sizes of the “sand” avalanches under consideration. In the last decade, the modeling of information processes using cellular automata methods and the theory of self-organized criticality have become widespread.

A



b

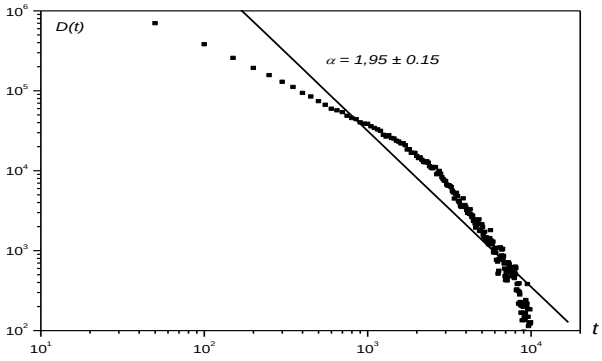


Figure. 3.10.5 – Distributions of (a) sizes of avalanches $D(s)$ and (b) times of their shedding $D(t)$

3.11. Percolation

Percolation networks are well studied and important in practical terms.

Tarasevich Yu.Yu. Percolation: theory, applications, practice. – M.: URSS, 2002.

Let us consider one of the simplest formulations of the percolation problem. Given a square grid (infinite), each connection of which has a resistance r_1, for

convenience we will call such a connection black. Randomly, black (conductive) bonds are broken. We can say that in this case, black connections are replaced by white ones with resistance $r_2 = \infty$. The task is to find such a concentration of black links p_c at and above which there is a connected part of black links, along which you can get from one infinity to another without jumping over the white link. Such a connected part extending to infinity is called an infinite cluster. Of course, the real grid is always finite, therefore, in this case, it is assumed that its size is much larger than the so-called correlation length, in this case, different implementations of the black link structure obtained by random cutting have the same properties. The probability of special degenerate distributions of black bonds is negligible. Here there is an analogy with the random distribution of gas molecules in a certain volume. The probability that all the gas will be collected in one half of the volume is so unlikely that it is never taken into account. At the same time, if there are only two molecules, then this probability is equal to $1/4$.

In addition to the geometric formulation of the percolation problem – the emergence of an infinite black cluster, one can also propose a physical formulation of the problem, for example, about the flow of current through black bonds. Black bonds conduct current, white ones do not. It is necessary to determine the resistance (conductivity) of the grid as a whole.

When $p > p_c$ the conductivity of the entire grid as a whole G is not equal to zero (the current finds its way from one contact at "infinity" to another along the black infinite cluster). At $p < p_c$ $G = 0$, and, accordingly, the resistance of the entire grid $R = 1/G = \infty$.

Snarskii A.A., Bezsudnov I.V., Sevryukov V.A. Transfer processes in macroscopically disordered media. – M.: URSS, 2007. – 304 p.

Of course, in the theory of percolation it is not necessary to consider exactly a two-dimensional square grid. Any dimensions and types of grids (homogeneous on average) are possible. In addition, we can talk not only about the problem of connections, but also about the problem of knots, when all bonds are

conductive, and conductive (black) nodes are randomly cut out with a given probability.

As it turned out, the problem of percolation, which appeared in the formulation of an applied engineering problem of the flow of a gas or liquid through a porous filter, is one of the simplest and most illustrative examples of the theory of second-order phase transitions and critical phenomena. So, many characteristics, described geometric and physical properties near the percolation threshold, p_c behave in a universal way, are described by critical exponents, the numerical value of which does not depend on the type of grid.

Let us consider some geometric characteristics of the percolation grid. There are many such geometric characteristics – the average number of clusters of size s, the distribution of clusters by size, the average size of a cluster, the power of an infinite cluster, the characteristics of various parts of an infinite cluster of the skeleton, skeleton, dead ends,...), etc. Here we will consider only some of the characteristics. The first of them is $n_s \ p$ the size distribution of final clusters, i.e. the number of clusters of s nodes (links) per one node (link) of the lattice. The second characteristic that suits well as an order parameter $P \ p$ is the power of an infinite cluster, the probability that an arbitrary node (link) belongs to an infinite cluster. The power of an infinite cluster $P \ p$ is expressed in terms of $n_s \ p$, for this it is enough to take into account that the probability of hitting a black node p is the sum of the probability of hitting an infinite cluster $P \ p$ or any finite $\sum_s sn_s \ p$:

$$P \ p + \sum_s sn_s \ p = p, \qquad (3.11.1)$$

where:

$$P \ p = p - \sum_s sn_s \ p \ . \qquad (3.11.2)$$

Near the percolation threshold, p_c the power of an infinite cluster behaves similarly to the order parameter in the theory of second-order phase transitions:

$$P\ p\ \sim\ p-p_c\ ^{\beta},\quad p-p_c\ /p_c\ . \qquad (3.11.3)$$

The role of temperature – T and critical temperature – T_c in phase transitions is now played by the concentration – of p well-conducting bonds/sites (black phase) and the percolation threshold – p_c.

The proximity to the percolation threshold will be denoted by:

$$\tau=\frac{p-p_c}{p_c}\ . \qquad (3.11.4)$$

The considered analogy between the theory of phase transitions and the theory of percolation can be deepened by introducing into the theory of percolation an analogue of a dimensionless magnetic field – h. In the geometric characteristics of percolation systems considered here, this is done in a rather skillful and artificial way, the so-called Castellain-Fortuin demon is introduced – a black node outside the lattice, associated with each black node with probability $1-\exp\ -h$. The analogue of free energy in the theory of phase transitions can be written as

$$G\ \tau,h\ =\sum_s n_s e^{-hs}, \qquad (3.11.5)$$

where $e^{-hs}=\ e^{-h\ s}$ is the proportion of finite clusters of s nodes in which none of the nodes is associated with the Castellain-Fortuin demon.

The order parameter in the theory of phase transitions can be found from G as a derivative with respect to the field h, at $h=0$

$$\left.\frac{\partial G}{\partial h}\right|_{h=0} = -\eta, \qquad (3.11.6)$$

and from (3.11.5) we find :

$$\left.\frac{\partial G}{\partial h}\right|_{h=0} = -\sum_s s n_s e^{-hs}\Big|_{h=0} = -\sum_s s n_s, \qquad (3.11.7)$$

which gives the main (singular) part $P\ p$ (3.11.2):

$$P\ p = p - \sum_s s n_s = p - \left.\frac{\partial G}{\partial h}\right|_{h=0}. \qquad (3.11.8)$$

At a zero field $h=0$, the order parameter $P\ p$ below the percolation threshold $p<p_c$ is equal to zero, which is completely analogous to the situation in the theory of phase transitions – at $T>T_c$ a ferromagnetic state (magnetization $m\neq 0$) passes into a paramagnetic state ($m=0$). At $h\neq 0$ and at $T>T_c$ there is a non-zero order parameter (3.11.5) "obliged" to the external magnetic field – h and therefore proportional to it. It is easy to see that the introduction of the Castellain-Fortuin demon also leaves the order parameter of the percolation theory $P\ p$ non-zero below the percolation threshold, i.e. and at $p<p_c$, there is an infinite cluster whose magnitude is proportional to h. Indeed, at, $p<p_c$ formally there is no infinite black cluster, but at $h\neq 0\,(h<<1)$ each black node is connected to the other through the Castellain-Fortuin demon with the probability

$$1-e^{-h} \approx 1-1+h = h, \qquad (3.11.9)$$

proportional to the field.

Which means the existence of an infinite cluster ($P\ p\neq 0$) proportional to h

$$P \quad p < p_c \quad \sim h. \qquad (3.11.10)$$

Let us now turn to the physical characteristics in the percolation theory, which make it possible to explain in a much more visual way the main regularities of phase transitions. We will now assume that black bonds in the percolation network have resistance r_1, and broken (white) ones have resistance $r_2 = \infty$. At grid sizes $L >> \xi$ much larger than the so-called correlation radius, ξ the influence of a particular random distribution of black and white bonds (random implementation of the structure) becomes insignificant and a well-defined value is the impedance – R. In order to abstract from this specific grid size (L), it is convenient to pass from the resistance of the entire sample (lattice) to the specific effective conductivity – σ_e:

$$R = \frac{1}{\sigma_e} \frac{L}{L^{d-2}}, \qquad (3.11.11)$$

where is $d = 2,3...$ the grid dimension.

By definition, on sizes of the order of and more than the correlation radius, all the properties of the network as a whole (in this case, the specific effective conductivity) are the same, respectively, the main, main elements of their flow structure should be the same.

As an order parameter describing the phase transition, it is convenient to introduce a quantity proportional to the effective conductivity – σ_e. As for the order parameter, $P \quad p$ the effective conductivity σ_e at $p > p_c$ is not equal to zero, but below the threshold at is $r_2 = \infty$ equal to $\sigma_e \quad p < p_c \quad = 0$. This behavior is easy to explain – above the flow threshold, current can flow from one contact to another (which can be formally spaced even for an infinite distance) passing through black – conductive, bonds.

This means that there is an infinite black cluster. At, $p < p_c$ only finite clusters exist, with a size less than the correlation length; they are isolated from each other because $r_2 = \infty$ and therefore no current can pass through the grid. That.

at, $r_2 = \infty$ the effective conductivity is zero $\sigma_e \ p < p_c \ = 0$, and if it r_2 is large, but finite, i.e. $h = r_1 / r_2 << 1$, but not zero, then the current will be able to flow from one end cluster to another. In this case, of course, the conductivity of the grid will be proportional to h $\sigma_e \ p < p_c \ \sim h$ and at $h \to 0$ effective conductivity $\sigma_e \ p < p_c \ \to 0$. Thus, there is no need to introduce the demon of Castellain-Fortuin, his role is played by white resistances with a large but finite resistance. The role of the external field is now played by the relation $h = r_1 / r_2 \ll 1$.

Critical behavior near the percolation threshold is exhibited not only by the density of the infinite cluster $P(p)$, but also by many other important characteristics of the percolation network, for example, the correlation length, which diverges when approaching p_c:

$$\xi \sim (p - p_c)^{-\nu}, \qquad (3.11.12)$$

where ν is the critical index of the correlation length.

The specific effective conductivity also behaves critically p_e. Near the percolation threshold above $(p > p_c)$ and below, $(p < p_c)$ the following occurs:

$$\rho_e = \begin{cases} \rho_1 \tau^{-t}, & p > p_c, \\ \rho_1 h |\tau|^q \equiv \rho_2 |\tau|^q, & p < p_c. \end{cases} \qquad (3.11.13)$$

The analogy between a second-order phase transition and a percolation transition is manifested here in the fact that if the critical temperature – T_c and percolation thresholds – p_c, for each material, or, accordingly, the lattice has its own numerical value, then the critical exponents are universal, depending only on the dimension of the problem, but independent of the lattice type.

Let us consider the question of applying the renormalization group method for calculating critical exponents.

Near the percolation threshold, the structure of connected parts of the percolation network (an infinite cluster, at $p > p_c$ and "lattice animals" at $p < p_c$) has a fractal structure, i.e. are statistically self-similar. Thus, passing from one scale to another and requiring scale invariance, one can obtain an approximate value of the critical exponents. Below we consider several examples of using the renormalization group method to calculate the percolation thresholds and the critical index of the correlation length ν.

Example 1. Triangular Lattice Leak Threshold, Node Problem

For convenience, we will talk in terms of current flow – a conductive node (black) conducts current, a non-conductive node (white) does not conduct, all connections are conductive. Figure 3.11.1 a - a triangular lattice with the designation of conductive (black) and non-conductive (white) nodes, with a triangular cell size equal to b; b – renormalized lattice, triangular cells (indicated in gray) now represent new nodes, the connections between which are indicated by thick black lines. The new nodes form a new (renormalized) triangular lattice with cell size $b' = \sqrt{3} \cdot b$.

The rules for transforming black nodes are as follows: the gray triangle of the lattice goes to the black node of the renormalized one if it 2 or 3 the node has black nodes, otherwise it goes to the white, non-conducting node.

The probability of a black node in a triangular lattice is p, so the probability of meeting a conducting black node in the new renormalized lattice is $p^3 + 3p^2(1-p)$, where the first term "owes" its origin to a gray triangle with three black nodes, and the second to two. At the same time, since the arrangement of black and white nodes in the second case is possible in three different ways, the second term has a factor of 3.

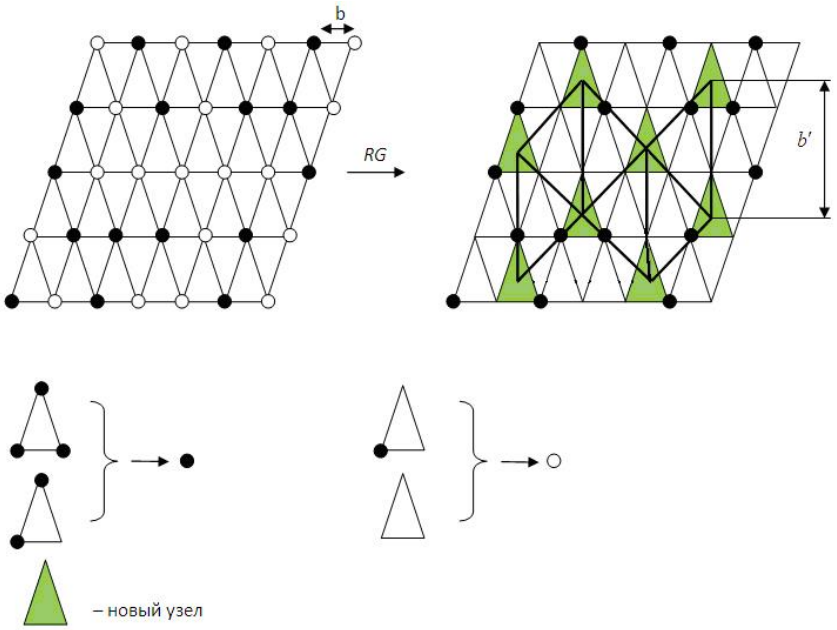


Figure. 3.11.1 – Schematic representation of black (well conducting phase) clusters

Thus, the probability of meeting a black node in the renormalized lattice p' is equal to:

$$p' = p^3 + 3 \cdot p^2 (1 - p). \qquad (3.11.13)$$

A network that is at the percolation threshold under a renormalization group transformation remains at the threshold:

$$p'(p_c) = p_c, \qquad (3.11.14)$$

those. p_c is the fixed point of the transformation. Then from the equation connecting each other p' and p we get:

$$p_c = p_c^3 + 3 \cdot p_c^2 (1 - p_c) \qquad (3.11.15)$$

This equation has three solutions $p_c = 1$, $p_c = 0$, $p_c = 1/2$. The first two of them are trivial – a completely "white" or "black" lattice remains so. Third decision

$$p_c = \frac{1}{2}, \qquad (3.11.16)$$

and is the desired triangular lattice percolation threshold for the knot problem.

In this example, unlike several others, everything works out so well that the expression for the percolation threshold obtained by the renormalization group method coincides with the exact value.

Let us now show how to express the critical index of the correlation length – ν.

Let in some lattice $\xi = \xi_0 \left| p - p_c \right|^{-\nu}$, then in renormalized $\xi' = \xi_0 \left| p' - p_c \right|^{-\nu}$ and $\xi' = \xi / a$, where $a = b' / b$. Thus:

$$a \left| p' - p_c \right|^{-\nu} = \left| p - p_c \right|^{-\nu}, \qquad (3.11.17)$$

whence for the critical index we find:

$$\frac{1}{\nu} = \frac{\ln \dfrac{p' - p_c}{p - p_c}}{\ln a} \equiv \frac{\ln \lambda}{\ln a}, \qquad \lambda = \frac{p' - p_c}{p - p_c}. \qquad (3.11.18)$$

Directing the concentration to the threshold p_c for λ can be written:

$$\lambda = \lim_{p \to p_c} \frac{p' - p_c}{p - p_c} = \left. \frac{dp'}{dp} \right|_{p = p' = p_c}, \qquad (3.11.19)$$

and for the critical index a simple expression is obtained

$$\frac{1}{\nu}=\frac{\ln\left(dp'/dp\big|_{p_c}\right)}{\ln a}. \qquad (3.11.20)$$

Earlier, for a triangular lattice it was obtained $p'=p^3+3\cdot p^2(1-p)$ and $p_c=1/2$, whence:

$$\nu=\frac{\ln a}{\ln b}=\frac{1}{2}\frac{\ln 3}{\ln(3/2)}\approx 1.355. \qquad (3.11.21)$$

The exact value (which can be obtained for a given lattice) $\nu=4/3\approx 1.33$, etc. application of the renormalization group method gives a very good approximation.

Example 2. Square lattice, constraint problem

One of the difficulties in applying the renormalization group method is the definition of a renormalizable cell and the percolation requirements for it. In a triangular lattice, this difficulty was not present, the cell was chosen to be triangular, and the percolation was determined by the presence of two or more black nodes. In the case of a square lattice, more accurate reasoning is required. By a percolating cluster, we mean both a cluster that connects "up and down" and a cluster that connects "left and right". Therefore, we will choose the flowing state for the cell, for example, as flowing “from left to right”. Then, as a cell of a square lattice, one can choose the configuration shown in Fig. 3.11.2.

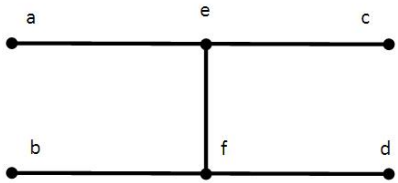


Figure. 3.11.2 – Square lattice cell

Cell of a square lattice, for the study of the flow "left-to-right". a and b are left contacts, c and d are right contacts. Each of the bonds – ae, ec,... is conductive with probability p.

When renormalized, the cell turns into one link:

Accordingly $a = b'/b = 2$.

Below, with indication of the probability of leakage, all cell configurations conducting “from left to right” are shown; 3.11.3.

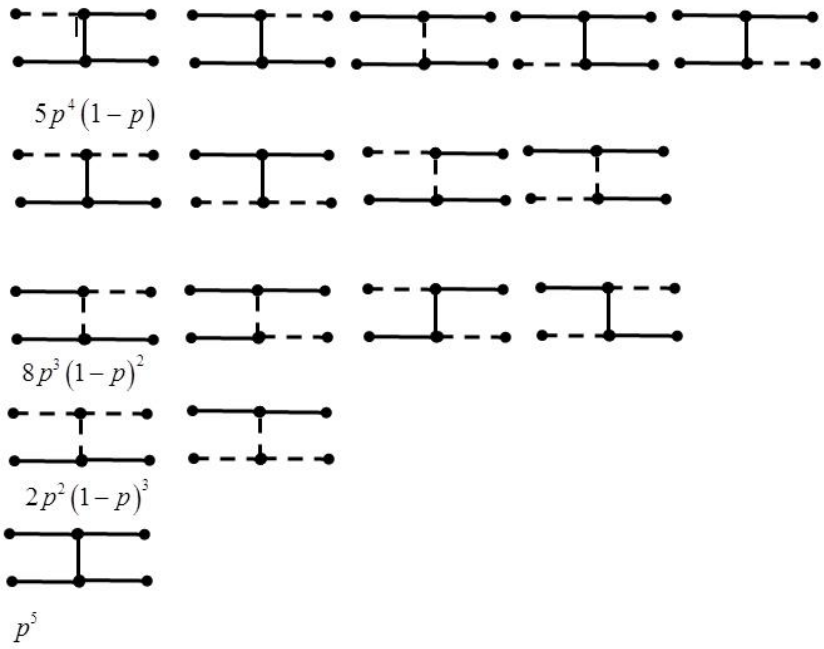


Figure. 3.11.3 – Conductive configurations

Thus, the probability of obtaining a leaky renormalized configuration is:

$$p' = p^5 + 5p^4(1-p) + 8p^3(1-p)^2 + 2p^2(1-p)^3 . \quad (3.11.22)$$

Substituting into the right and left parts, $p' = p = p_c$ it is easy to make sure that the solution of the resulting equation gives:

$$p_c = \frac{1}{2} . \quad (3.11.23)$$

You can immediately calculate the critical index of the correlation length:

$$\nu = \frac{\ln b}{\ln \lambda}, \quad \lambda = \left.\frac{dp'}{dp}\right|_{p=p_c} , \quad (3.11.24)$$

taking the derivative of $p' = p'(p)$, by p and substituting $p = p_c = 1/2$. We find:

$$\left.\frac{dp'}{dp}\right|_{p=p_c} = \left(10p^4 - 20p^3 + 6p^2 + 4p\right)\Big|_{p=p_c=1/2} = \frac{13}{8} . \quad (3.11.25)$$

Thus:

$$\nu = \frac{\ln 2}{\ln\left(13/8\right)} \approx 1.43, \quad (3.11.26)$$

n at the exact value of $\nu = 4/3$.

The renormalization group method does not always give such good results. For clarification, it is necessary to take large sizes of the renormalizing cell, for example, instead of a cell with $b = 2$ take a cell with $b = 5$.

At first glance, this clarification does not represent a significant complication. However, in fact, this is not just a significant, but a fundamentally significant complication, since

instead of 32-x for the case $b=2$ combinations of which 20 leaky, in case $b=5$ the number of combinations is equal to $2^{25} \approx 3 \cdot 10^7$, from which it is first necessary to choose leaky ones, and then also find the probability of their occurrence.

3.12. Correlation and fractal analysis

3.12.1. The concept of a fractal

Mandelbrot B. Fractal geometry of nature. – M.: Institute of Computer Research, 2002. – 656 p.

Mandelbrot B. Fractals, chance and finance. Moscow: Regular and Chaotic Dynamics, 2004. 256 p.

The term fractal was proposed by Benoit Mandelbrot (B. Mandelbrot) in 1975 to denote irregular self-similar mathematical structures. The basic definition of a fractal given by Mandelbrot was: "A fractal is a structure that consists of parts that are in some sense similar to the whole." It should be recognized that this definition, due to its laxity, is not always correct. Many examples of self-similar objects that are not fractals can be cited, for example, a picture of railroad tracks descending to the horizon.

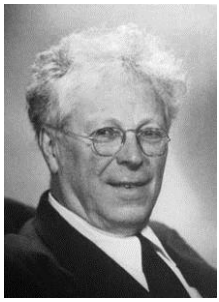
In the simplest case, a small part of the fractal contains information about the entire fractal. A rigorous definition of self-similar sets was given by J. Hutchinson (J. Hutchinson) in 1981. He called a set self-similar if it consists of several components similar to the entire set, i.e. components obtained by affine transformations – rotation, compression and reflection of the original set. Note that such a strict definition is unproductive. In many cases, subsets are studied in which the component of similarity to the entire set is only approximate, and in the Mandelbrot set itself, the components are only similar (and sometimes not similar) to the entire set.

However, self-similarity is a necessary but far from sufficient property of fractals. After all, it is impossible, in fact, to consider a point or a plane drawn by cells as a fractal. The main feature of fractal objects is that the "standard" topological dimension is not enough to describe them d_τ, which, as you know, is equal to 1 for a line ($d_\tau = 1$- a line is a one-dimensional

object), for a surface $d_\tau = 2$, etc. Fractals are characterized by a geometric "indentation".

Therefore, a special concept of fractal dimension introduced by F. Hausdorf (F. Hausdorf) and A.S. Besikovich. As applied to ideal objects of classical Euclidean geometry (lines, planes...), it gave the same numerical values as the topological dimension, but the new dimension had a finer sensitivity to all sorts of imperfections in real objects, making it possible to distinguish and individualize what had previously been faceless and indistinguishable. The dimension of Hausdorff-Besikovich just allows you to measure the degree of "indentation". The dimension of fractal objects is not an integer characteristic of the usual geometric objects. At the same time, in most cases, fractals resemble objects that densely occupy real space, but do not use it completely.

Felix Hausdorff (1868-1942)



Abram Samoylovich Besikovich (1891-1970)

Let there be a set G in a Euclidean space of dimension d_τ. This set is covered with cubes of dimension d_τ, while the length of the edge of any cube does not exceed some value δ, i.e. $\delta_i < \delta$

.

, depending on some parameter d, and δ over all elements of the coverage is introduced:

$$l_d(\delta) = \sum_i \delta_i^d \text{ . (3.12.1)}$$

Let's define the lower bound of this sum:

$$L_d(\delta) = \inf \sum_{i,\delta_i<\delta} \delta_i^d \text{ . (3.12.2)}$$

When decreasing the maximum length δ, if the parameter d is large enough, it will obviously run:

$$\lim_{\delta \to 0} L_d(\delta) \to 0. \quad (3.12.3)$$

For some sufficiently small value of the parameter, the following d will be performed:

$$\lim_{\delta \to 0} L_d(\delta) \to \infty. \quad (3.12.4)$$

Intermediate, critical value d_x, for which:

$$\lim_{\delta \to 0} L_d(\delta) = \begin{cases} 0, & d > d_x, \\ \infty, & d < d_x, \end{cases} \quad (3.12.5)$$

and is called the Hausdorff-Besikovich dimension (or fractal dimension). For simple geometric objects, the Hausdorff-Besikovich dimension coincides with the topological one (for a segment d_x = 1, for a square d_x = 2, for a cube d_x = 3, etc.)

Despite the fact that the Hausdorff-Besikovich dimension is perfectly defined from a theoretical point of view, for real fractal objects, the calculation of this dimension is very difficult. Therefore, a somewhat simplified indicator is introduced – capacitive dimension d_c.

Grinchenko V.T., Matsypura V.T., Snarskii A. A. Introduction to nonlinear dynamics. Chaos and fractals. – M.: LKI, 2007. – 264 p.

When determining this dimension, cubes (squares, segments...) with faces of the same size are used. In this case, of course, it is true:

$$L_d(\delta) = N(\delta)\delta^{d_c}, \quad (3.12.6)$$

where $N(\delta)$ is the number of cubes covering the area G. By taking the logarithm and passing to the limit with a decrease in the face of the cube ($\delta \to 0$), we obtain:

$$d_c = -\lim_{\delta \to 0} \frac{\log N(\delta)}{\log \delta}, \qquad (3.12.7)$$

if this limit exists. It should be noted that in most numerical methods for determining the fractal dimension, exactly is used d_c, while it must be taken into account that the condition is always true: $d_x \le d_c$. For regular self-similar fractals, the capacitive dimension and the Hausdorff-Besikovich dimension coincide; therefore, terminologically they are often not distinguished and they simply speak of the fractal dimension of an object.

When carrying out practical calculations of the fractal dimension for real objects, the following methodological technique is used. Suppose that at some stage of the fractal coverage it was necessary to use $N(\delta)$ cubes with faces of size δ, and at another stage, $N(\delta')$ elements with faces of size δ'. In view of the expected power dependence, it is true:

$$N(\delta) \sim \frac{1}{\delta^{d_i}}, \quad N(\delta') \sim \frac{1}{\delta'^{d_i}}, \qquad (3.12.8)$$

from where the value d_c can be evaluated as:

$$d_c = -\frac{\log\left(N(\delta)/N(\delta')\right)}{\log\left(\delta/\delta'\right)}. \qquad (3.12.9)$$

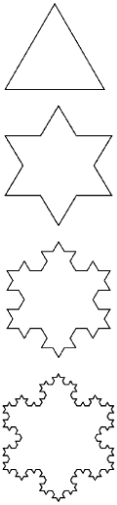
3.1 2.2. Abstract fractals

Let us consider the principles of formation of several abstract fractal objects that have pronounced self-similarity properties.

The construction of the fractal set, snowflake Helge von Koch (H. Von Koch), begins with a regular triangle, the side length of which is equal to 1. The side of the triangle is considered the base link. Further, at any iteration step, each link is replaced by a generating element – a broken line, which consists along the edges of the original link of segments 1/3 of

the length of this link, between which two sides of a regular triangle are placed with a side also 1/3 of the length of the link.

Niels Fabian Helge von Koch (1870-1924)



The first 4

generations of the Koch snowflake

All segments – sides of the resulting polyline are considered basic links for the next iteration. The curve obtained as a result of the nth iteration for any finite n is called a prefractal, and only when n tends to infinity does the Koch curve become a fractal. The fractal set obtained as a result of the iterative process is a line of infinite length, which limits the finite area. Indeed, at each step, the number of sides of the resulting polygon increases by a factor of 4, while the length of each side decreases only by a factor of 3, i.e. the length of the polygon at the nth iteration is equal to $3\times\left(4/3\right)^n$ and tends to infinity as n. The area under the curve, if we take the area of the first generating triangle as a unit, is equal to:

$$S = 1 + \frac{1}{3}\sum_{k=0}^{\infty}\left(\frac{4}{9}\right)^k = \frac{8}{5} = 1,6. \quad (3.12.10)$$

Let's calculate the fractal dimension of the Koch snowflake. Let the side length of the original triangle be equal to one. In this case, the role of the cubes covering the figure in question is played by line segments. Then at the zero step we have: $\delta = 1,\ N(\delta) = 3$. For the second step it is true: $\delta' = 1/3,\ N(\delta') = 12$. These data are sufficient to estimate the fractal dimension:

$$d_c = -\frac{\log\left(N(\delta)/N(\delta')\right)}{\log\left(\delta/\delta'\right)} = -\frac{\log\left(3/12\right)}{\log\left(3\right)} = \frac{\log 4}{\log 3} \sim 1,26 \quad (3.12.11)$$

Vaslav Francis Sierpinski

(1882-1969)

The self-similar fractal proposed in 1915 Vaslav Serpinski, is formed according to the following rules. The initial set corresponding to the zero step is an equilateral triangle. Then it is divided into four regions by connecting the seridines of the sides of the original triangle with line segments. Then the interior of the central region of the original triangle is removed – a small internal "inverted triangle". Then, in the next iteration step, this process is repeated for each of the three remaining triangles. Continuing the described procedure to infinity, a set is formed, called the Sierpinski napkin.

Obviously, the fractal dimension of Sierpinski's napkin is:

$$d_c = -\frac{\log\left(N(\delta)/N(\delta')\right)}{\log(\delta/\delta')} = \frac{\log 3}{\log 2} \approx 1{,}58. \qquad (3.12.12)$$

This fractal is interesting because the area it occupies is zero. To substantiate this fact, we calculate the total area of the parts excluded during construction. At the first step, a quarter of the area of the original triangle is discarded, at the second step, a quarter of the area is removed from each of the three triangles, and so on. Thus, the total remote area is calculated as the sum of the series (the area of the original triangle is taken equal to one):

$$\begin{aligned} S &= \frac{1}{4} + \frac{3}{4}\cdot\frac{1}{4} + \frac{3}{4}\cdot\frac{3}{4}\cdot\frac{1}{4} + \ldots = \\ &= \frac{1}{4}\cdot\left[1 + \left(\frac{3}{4}\right) + \left(\frac{3}{4}\right)^2 + \left(\frac{3}{4}\right)^3 + \ldots\right] = \\ &= \frac{1}{4}\cdot\frac{1}{1-3/4} = 1. \end{aligned} \qquad (3.12.13)$$

Thus, the excluded area is the same as the area of the original triangle.

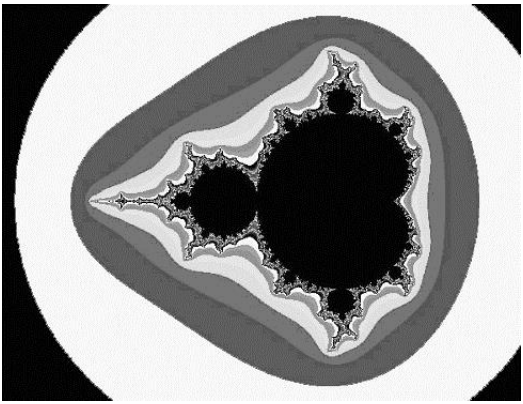
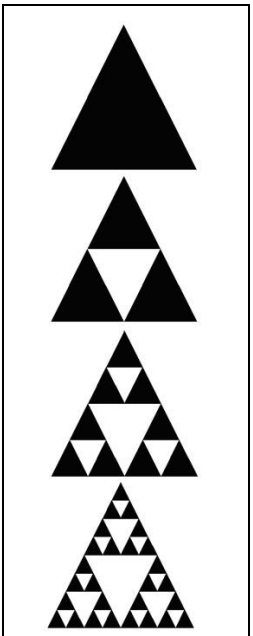


Figure. 3.12.1 – Mandelbrot set



Construction of the Sierpinski Napkin

The algorithm for constructing the Mandelbrot set (Fig. 29) is based on an iterative calculation according to the formula:

$$Z_{i+1} = Z_i^2 + C, \quad i = 0,\ 1,\ 2,\ ..., \qquad (3.12.14)$$

where Z_{i+1}, Z_i and C are complex variables.

Iterations are performed for each starting point C of a rectangular or square region – a subset of the complex plane. The iteration process continues until it Z_i goes beyond the circle of a given radius, the center of which lies at the point (0,0), or after a sufficiently large number of iterations. Depending on the number of iterations during which Z_i stays inside the circle, point colors are set C (Fig. 3.12.1). If Z_i remains inside the circle for a sufficiently large number of iterations, then this pixel is painted black. The Mandelbrot set contains precisely those points that do not go to infinity during an infinite number of iterations. Since the number of iterations corresponds to the color number, the points that are closer to the Mandelbrot set have a brighter color.

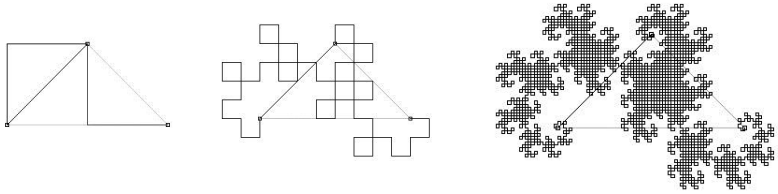
In the 80s of the last century, the method of "Systems of Iterative Functions" (Iterated Functions System – IFS), which is a simple means of obtaining fractal structures. IFS is a system of functions that map one multidimensional set to another. The

simplest implementation of IFS is an affine plane transformation:

$$X' = AX + BY + C,$$
$$Y' = DX + EY + F, \qquad (3.12.14)$$

where X, Y are the previous values of the coordinates, X', Y' are the new values, A, B, C, D, E and F are the coefficients.

As an example of the use of IFS to build fractal structures, we can cite the "dragon" of Harter-Hateway (3.12.2), formed using the Java applet provided on the Internet at http://www.fractals.nsu.ru/fractals.chat.ru/ifs2.htm (Fig. 3.12.1). Using IFS to compress conventional images, such as photographs, is based on local self-similarity detection (unlike fractals where global self-similarity is observed).



a b c

Figure. 3.12.2 – "Dragon" Harter-Heituey: a) – the second step of the iteration; b) – the sixth step; c) the twelfth step

In the 80s, M. Barnsley and A. Sloane proposed the idea of compressing and storing graphic information, based on considerations from the theory of fractals and dynamic systems. Based on this idea, an algorithm for fractal information compression was created, which allows you to compress some samples of graphic information by 500-1000 times. Each image is encoded with several simple affine transformations.

According to the Barnsley algorithm, pairs of areas are selected in the image, the smaller of which is similar to the larger one, and several coefficients are stored that encode the transformation that converts the larger area into a smaller one. In this case, it is required that a plurality of such areas cover the entire image.

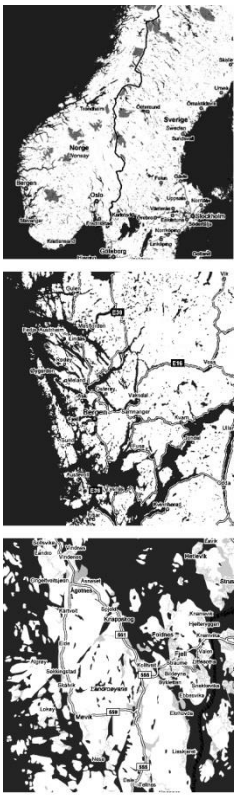
One of the best examples of fractals in nature is the structure of coastlines. Indeed, on a kilometer stretch, the coast looks as cut as on a hundred-kilometer one. Experience shows that the length of the coastline L depends on the scale l, at which measurements are taken, and increases with the decrease of the latter according to a power law $L = \Lambda l^{1-\alpha}$, $\Lambda = const.$. For example, the fractal dimension of the UK coastline (Fig. 31) is $\alpha \approx 1.52$.

3.12.3. Information flows and fractals

The application of the theory of fractals in the analysis of information flows allows us to look at the patterns that form the basis of computer science from a general position. It is known that many information retrieval systems that include elements of cluster analysis allow you to automatically detect new classes and distribute documents among these classes. Accordingly, it is shown that thematic information arrays are self-similar developing structures, however, their self-similarity is valid only at the statistical level (for example, the distribution of thematic clusters of documents by size).

What determines the nature of the fractal properties of information flows generated by such cluster structures? On the one hand, the parameters of rank distributions, and on the other hand, the mechanism for the development of information clusters. The appearance of new publications increases the size of already existing clusters and causes the formation of new ones.

Fractal properties are also characteristic of clusters of informational websites on which documents are published that correspond to certain topics.



Coast of Norway at different scales

(according to maps.google.com)

The volumes of messages in thematic information flows form time series, the study of which is increasingly using the theory of fractals.

Studying the characteristics of time series generated by information flows, the messages of which reflect the processes taking place in the real world, makes it possible to predict their dynamics, reveal hidden correlations, cycles, etc.

The results of real numerical experiments are given as illustrations. As a basis for the study of the fractal properties of the series, reflecting the intensity of publications of thematic information flows, the system of content monitoring of news from Internet websites InfoStream was used. The subject of the studied information flow was determined by a request to this system. To illustrate, an analysis was made of online media reports – an array of 1,4,069 documents published since January 1, 2006. December 31, 2007, on the subject of computer virology, satisfying the request:

"computer virus" OR "virus attack" OR (antivirus AND (program OR utility OR Windows OR linux))

determined by daily discreteness (Fig. 3.12.3).

Let us dwell in more detail on some methods for analyzing this type of time series, generated, in particular, by information flows.

3.12.4. DFA Method

One of the universal approaches to identifying self-similarity is based on the DFA method (Detrended Fluctuation Analysis) is a universal method for processing series of measurements.

DFA method is a variant of the analysis of variance that allows you to explore the effects of long-term correlations in non-stationary series. In this case, the root-mean-square error of the linear approximation is analyzed depending on the size of the approximation segment.

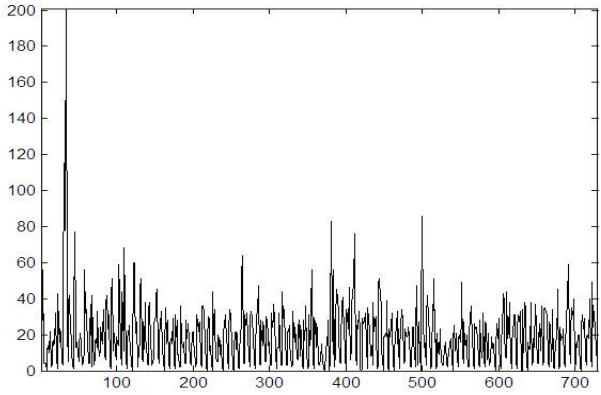


Figure 3.12.3 – The number of thematic publications (y-axis) in the context of dates – the ordinal number of the day (x-axis)

Peng C.-K., Havlin S., Stanley HE, Goldberger AL Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series. Chaos. – Vol 5. – 1995. – P. 82.

One of the universal approaches to identifying self-similarity is based on the DFA method (Detrended Fluctuation Analysis) is a universal method for processing series of measurements.

DFA method is a variant of the analysis of variance that allows you to explore the effects of long-term correlations in non-stationary series. In this case, the root-mean-square error of the linear approximation is analyzed depending on the size of the approximation segment. Within the framework of this method, the data is first reduced to zero mean (subtraction of the mean value $\langle F \rangle$ from the time series F_n, $n = 1, ..., N$):

$$y(k) = \sum_{i=1}^{k} \left[F_i - \langle F \rangle_N \right]. \qquad (3.12..16)$$

Then a series of $y(k)$ values $k = 1,...,N$ split into non-overlapping segments of length *n*, within each of which the least squares method determines the equation of a straight line approximating the sequence $y(k)$. The found approximation $y_n(k)$ ($y_n(k) = a_n k + b_n$) is considered as a local trend.

Next, the root-mean-square error of the linear approximation $D(n)$ is calculated over a wide range of n *values* :

$$D\ n\ =\sqrt{\frac{1}{N}\sum_{k=1}^{N}\ y(k)-y_n(k)^{\ 2}}\ .\quad (3.12.17)$$

In the case when the dependence $D(n)$ has a power-law character $D\ n\ \sim n^{\alpha}$, i.e. the presence of a linear section at a double logarithmic scale $\ln D \sim \alpha \ln n$, we can talk about the existence of scaling.

As can be seen from fig. 3.12.4, the values $D(n)$ for the selected information flow depend in a power-law manner on n, i.e. on a double logarithmic scale, this dependence is close to linear.

5.12.5. Correlation analysis

If denoted by Y_t member of the number of publications series (the number of emails received, for example, per day t, $t=1,...,N$), then the autocorrelation function for this series Y is defined as:

$$m=\frac{1}{N}\sum_{t=1}^{N}Y_t$$

$$F(k)=\frac{1}{(N-k)}\sum_{t=1}^{N-k}(Y_{t+k}-m)(Y_t-m)= \\ =\frac{1}{N-k}\sum_{t=1}^{N-k}X_{t+k}X_t,\quad (3.12.18)$$

where m is the mean value of the series Y .

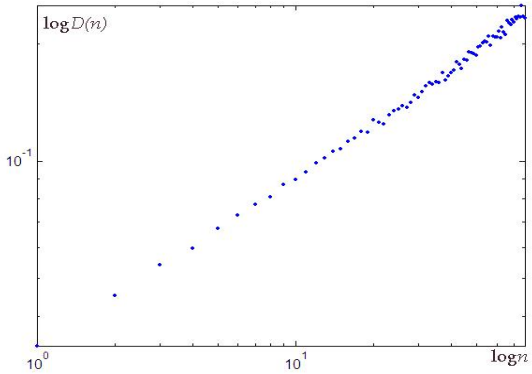


Figure 3.12.4 – Dependence D (n) of a series of observations (ordinate axis) on the length of the approximation segment n (abscissa axis) in a logarithmic scale (α =0.7)

It is assumed that the series Y may contain a hidden periodic component.

It is known that the autocorrelation function has the property that if a hidden periodic component exists, then its value asymptotically approaches the square of the mean value of the original series.

Brillouin L. Science and information theory. – M.: State. ed. Phys.-Math. lit., 1960.

There is a well-known theorem that if the considered series is periodic, i.e. can be presented as:

$$X_t = u_0 \cos(\omega t + \varphi), \qquad (3.12.19)$$

then its autocorrelation function will be:

$$F(k) = \frac{1}{2} u_0^2 \cos(\omega k), \qquad (3.12.20)$$

i.e. _ the autocorrelation function of the periodic series is also periodic, having the same frequency, but without the phase angle φ.

Consider a number series X, which is the sum of some meaningful component N and a sinusoidal signal S:

$$X_t = N_t + S_t. \qquad (3.12.21)$$

Let's find the autocorrelation function for this series (the values are reduced to the average $m = 0$ and divided by standard deviations):

$$F(k) = \frac{1}{N-k}\sum_{t=1}^{N-k} X_{k+t}X_t =$$

$$= \frac{1}{N-k}\sum_{t=1}^{N-k}(N_{k+t} + S_{k+t})(N_t + S_t) =$$

$$= \frac{1}{N-k}\sum_{t=1}^{N-k} N_{k+t}N_t + \frac{1}{N-k}\sum_{t=1}^{N-k} S_{k+t}S_t + \frac{1}{N-k}\sum_{t=1}^{N-k} N_{k+t}S_t + \frac{1}{N-k}\sum_{t=1}^{N-k} S_{k+t}N_t.$$

(3.12.22)

Obviously, the first term is a non-periodic function, asymptotically tending to zero. Since there is no mutual correlation between N and S, the third and fourth terms also tend to zero. Thus, the non-zero contribution is the second term – signal autocorrelation S. Those. the autocorrelation function of the series X remains periodic.

As an illustration, consider an information flow model that considers a time series corresponding to the number of new messages in the network. It is assumed that the daily number of messages in the network grows exponentially (with a very small exponential value), and this number is superimposed by fluctuations associated with the weekly cyclicity in the work of information sources. A certain element of randomness is also taken into account, expressed by the corresponding deviations.

To obtain the corresponding time series, the values of the function were considered:

$$y = ae^{0.001x} + \sin(\pi x/7 + a), \qquad (3.12.23)$$

which implements the simplest information flow model – the exponent is responsible for the increase in the number of publications over time (general trend), the sine is for the weekly frequency, the parameter a is for random deviations. The number of publications y cannot be a negative number. On fig. 3.12.5 shows the graph of the model.

The original series was processed: reduced to zero mean and normalized (each term divided by the mean). After that, the correlation coefficients were calculated, which for series of measurements X with length N are calculated by the formula:

$$\sigma^2 = \frac{1}{N}\sum_{t=1}^{N}(Y_t - m)^2$$

$$R(k) = \frac{F(k)}{\sigma^2}, \qquad (3.12.24)$$

where $F(k)$ is the autocorrelation function; σ^2 - dispersion.

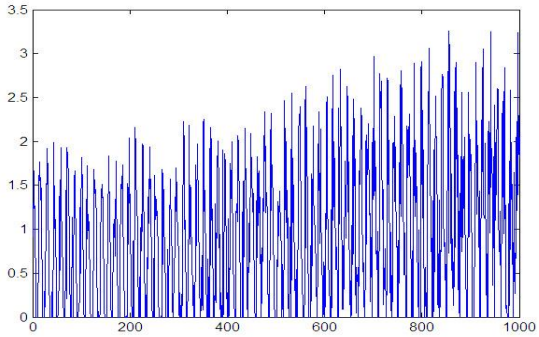


Figure. 3.12.5 – Flow model with exponential growth (abscissa – variable x - day, ordinate – variable y - number of publications)

On fig. 3.12.6 shows a graph of the values of the correlation coefficients (abscissa axis – variable *k*, ordinate axis – correlation coefficient $R(k)$.

A graphical representation of the correlation coefficient for a series of observations corresponding to the dynamics of the real information flow of web publications indicates the invariance of the correlation properties by day of the week (Fig. 3.12.7). At the same time, the correlation coefficients of a series of observations averaged over weeks are approximated by a hyperbolic function that characterizes the long-term dependence of the original series members (Fig. 3.12.8).

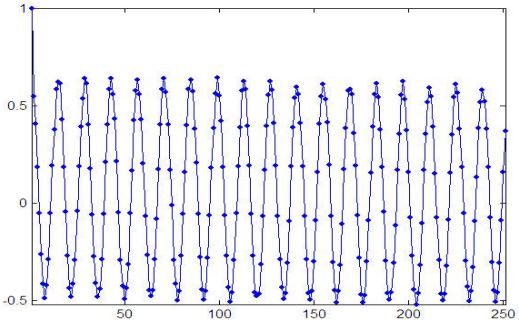


Figure. 3.12.6 – Values of model correlation coefficients

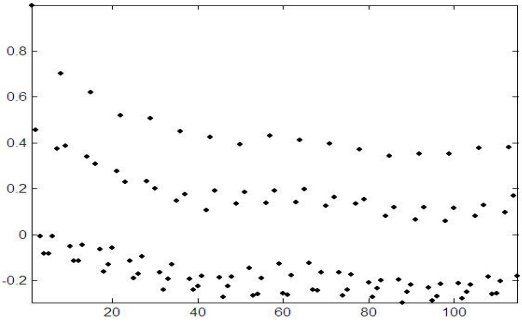


Figure. 3.12.7 – Correlation coefficients of a series of observations $R(k)$ (ordinate axis) depending on k (abscissa axis)

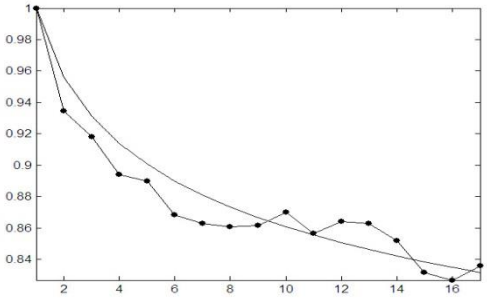


Figure. 3.12.8 – Correlation coefficients of a series of observations $R(k)$ (y-axis) averaged over weeks depending on k (abscissa)

3.12.6. Fano factor

To study the behavior of processes, it is customary to use another indicator – the dispersion dispersion index (IDC), the so-called Fano factor. This value is defined as the ratio of the variance of the number of events $\sigma^2(k)$ on a given observation window k to the corresponding mathematical expectation $m(k)$:

$$\Phi(k) = \sigma^2(k) / m(k). \qquad (3.12.25)$$

For self-similar processes, the following relation holds:

$$\Phi(k) \approx 1 + Ck^{2H-1}, \qquad (3.12.26)$$

where C and H are constants.

Hugo Fano (1912-2001)

Fano U. Ionization field of radiations. II. The fluctuations of the number of ions. Phys. Rev., 1947. – No. 72.

3.12.7. R/S analysis. Hurst exponent

Hurst exponent (H. E. Hurst) $-H$ is related to the coefficient of the normalized range R/S, where R is the "range" of the corresponding time series calculated in a certain way, and S is the standard deviation. Hurst experimentally found that for many time series it is true: $R/S = (N/2)^H$.

It is proved that the Hurst exponent is related to the traditional "cellular" fractal dimension D by a simple relationship:

$$D = 2 - H. \qquad (3.12.27)$$

The condition under which the Hurst exponent is associated with the fractal "cell" dimension in accordance with the above formula is defined by E. Feder as follows: therefore, the relationship is valid when the structure of the curve describing the fractal function is examined with high resolution, i.e. in the local limit. Another important condition is the self-affinity of the function.

The Hurst exponent characterizes the persistence – the propensity of the process to trends (in contrast to the usual Brownian motion). The value $H > ½$ means that the dynamics of the process directed in a certain direction in the past, most likely, will entail the continuation of movement in the same direction. If $H < ½$, then the process is predicted to change direction. $H = ½$ means uncertainty – Brownian motion.

Harold Edwin Hurst (*1880-1978)*

Hurst H. Long Term Storage Capacity of Reservoirs. Transactions of the American Society of Civil Engineers, 1951. No. 116.

In particular, to study the fractal characteristics of thematic information flows for time series $F(n)$, $n = 1,...,N$, composed of the number of messages published over a period of time from $n-1$ to n, the value of the Hurst exponent was studied, determined from the ratio:

$$R/S = \left(N/2\right)^H, \quad N >> 1. \tag{3.12.28}$$

Here S is the standard deviation:

$$S = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(F(n) - \left\langle F\right\rangle_N\right)^2}, \tag{3.13.29}$$

$$\left\langle F\right\rangle_N = \frac{1}{N}\sum_{n=1}^{N}F(n), \tag{3.13.30}$$

and R - the so-called scope:

$$R(N) = \max_{1\le n\le N} X(n,N) - \min_{1\le n\le N} X(n,N), \tag{3.12.31}$$

Where

$$X(n,N) = \sum_{i=1}^{n}\left(F(i) - \left\langle F\right\rangle_N\right). \tag{3.12.32}$$

Studies of the fractal properties of measurement series obtained as a result of monitoring thematic information arrays from the Internet indicate that the indicator H takes values in the range of 0.65 ÷0.75, i.e. much more than ½. Therefore, it can be argued that in this case persistence is found (the existence of long-term correlations that can be associated with the manifestation of deterministic chaos). It turns out that the series $F(n)$ has a fractal dimension D equal to $D = 2 - H \approx 1.35 \div 1.25$.

Studies of thematic information flows confirm the assumption of self-similarity and iteration of processes in the web space. Publications, citations, direct references, etc. generate self-similarity, which manifests itself in stable statistical distributions and well-known empirical laws.

3.13. Pareto-Zipf laws

Wilfredo Pareto

(1848 – 1923)

Analyzing social processes, V. Pareto considered the social environment as a pyramid, at the top of which are some people representing the elite. As a result of research, he mathematically formulated the relationship between the amount of income and the number of people who receive it. Pareto in 1906 found that about 80% of the land in Italy belongs to only 20% of its inhabitants. He came to the conclusion that the parameters of the distribution obtained by him are approximately the same and do not differ fundamentally in different countries and at different times. Exactly the same Pareto pattern is observed in the distribution of income between people.

Pareto income distribution is described by the equation $N \approx A/X^{p}$, where X *is* the amount of income, N is the number of people with income equal to or greater than X, A and p *are* the distribution parameters. In mathematical statistics, this distribution was named Pareto, while natural restrictions on the parameters are assumed: $X \geq 1$, $p > 1$.

The Pareto distribution has the stability property, i.e. the sum of two random variables that have a Pareto distribution will also follow this distribution. The observed rule, called "Pareto's law" or "the 80/20 principle", is applicable in very many areas. For example, in an information search, it is enough to identify 20% of the most important keywords to find 80% of the required documents, and then expand the search or use the "find similar" option to complete the task. Another example: 80% of the visits to a website come from only 20% of its web pages.

When building queuing systems, including information retrieval systems, it is necessary to take into account the fact that the most complex functionality of the system, the implementation of which takes 80 or more percent of labor costs, will be used by no more than 20 percent of users of this system.

In a strict formulation, this effect is called the Pareto principle. Assume that the sequence $x_1, x_2, ...x_n, ...$ corresponds to the incomes of individual people. After ranking this sequence in descending order, a new sequence is obtained $x_1, x_2, ..., x_r, ...$ (elements x_r are arranged in descending order).

Suppose that N is the total number of people whose income is at least $x_{(r)}$, i.e. $N = r$. Then the Pareto rule can be rewritten as follows:

$$r = \frac{A}{x_{(r)}{}^{p}}. \quad (3.13.1)$$

Where:

$$x_{(r)} = \left(\frac{A}{r}\right)^{\frac{1}{p}}. \quad (3.13.2)$$

The sum of the first n ($n = 1, 2, ..., N$) values of the quantity is considered x_r, i.e. The total income of the richest people is $m\ n$:

$$m\ n = \sum_{r=1}^{n} x_r = \sum_{r=1}^{n} \left(\frac{A}{r}\right)^{1/p} = \sum_{r=1}^{n} \frac{C}{r^{\gamma}}, \quad (3.13.3)$$

Where $\gamma = 1 - 1/p;\ C = A^{1/p}$.

Passing from discrete values to continuous ones (assuming that $n >> 1$), we have:

$$m\ n \approx \int_1^n \frac{C}{r^\gamma} dr \approx \frac{C}{1-\gamma} n^{1-\gamma}. \qquad (3.13.4)$$

In dimensionless variables $\mu = m\ n\ / m\ N$ - and $\nu = n/N$ the last equality has the form (see Fig. 23):

$$\mu = \nu^{1-\gamma}. \quad (3.13.5)$$

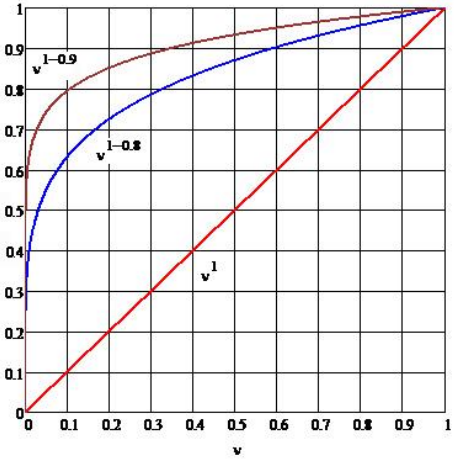


Figure 3.13.1 – Pareto distribution for different parameter values: dependence $\mu = \nu^{1-\gamma}$ for three cases: $\gamma = 0$, $\gamma = 0.8$, $\gamma = 0.9$

The value μ - in our example – the relative amount of income received by the first in rank n people, whose share (relative to all people) is ν.

J. Zipf studied the use of the statistical properties of language in text documents and identified several empirical laws, which he presented as empirical evidence of his "principle of least effort". He experimentally showed that the distribution of words in a natural language obeys a law that is often cited as Zipf's first law relating to the distribution of word frequency in a text. This law can be formulated as follows. If for some rather large text we make a list of all the words that are found in it, and then rank these words in descending order of the frequency of their occurrence in the text, then for any word the product of its rank and frequency of occurrence will be a constant value:, where *is* $f \times r = c$ the f frequency occurrence of the word in the text; r is the rank of the word in the list; c is an empirical constant value (Zipf coefficient). For Slavic languages in particular, the Zipf coefficient is approximately 0.06-0.07.

George Zipf (1902 – 1950)

Manning CD, Schütze H. Foundations of Statistical Natural Language Processing – Cambridge, Massachusetts: The MIT Press , 1999.

The above dependence reflects the fact that there is a small vocabulary that makes up the bulk of the tokens of the text. These are mostly official words. For example, given in the monograph by K.D. Manning and G. Schutze's analysis of the novel "Tom Sawyer" made it possible to identify 11,000 English words. At the same time, twelve words were found (the, and, etc.), each of which covers more than 1% of lexemes in the novel.

Zipf explained the hyperbolic distribution by the "principle of least effort", assuming that when creating a text, less effort is spent on repeating some words than on using new ones, i.e. to appeal to "working memory, not long-term memory."

Zipf formulated another pattern, which is that the frequency and number of words that enter the text with a given frequency are also related by a similar relationship, namely:

$$N(f) = \frac{B}{f^{\beta}},$$

(3.13.6)

where $N(f)$ is the number of different words, each of which is used in the text f times, B is some normalization constant.

A lexeme is a word as an abstract unit of morphological analysis. Different word forms of one word are combined into one lexeme. For example, ***dictionary, dictionary, dictionary*** *– these are forms of the same lexeme, by convention written as* ***a dictionary.***

There is a simple quantitative model for determining the dependence of frequency on rank. Let's assume that random text is generated by a monkey on a typewriter. With probability p a space is generated, and with probability $(1-p)$ other characters are generated, each of which has an equal probability. It is shown that the text obtained in this way will give results similar in form to the Zipf distribution.

A more complex model for generating a random text that satisfies Zipf's second law was proposed by G.A. Simon in 1955. According to this model, if the text has reached n word length, then what $(n+1)$ the th word of the text will be is determined by two assumptions:

1. Let $N(f,n)$ be the number of different words, each of which was used f once among the first n words of the text. Then the probability that $(n+1)$ the -th will be a word that was previously used f once is proportional $f \cdot N(f,n)$ to – the total number of occurrences of all words, each of which was previously used f once.

2. With probability δ $(n+1)$ The th word will be the new word.

Assumption 1 implies:

$$N(f,n+1) - N(f,n) = K(n)\ (f-1) \cdot N(f-1,n) - f \cdot N(f,n)\ , \quad (3.13.6)$$

where $K(n)$ is the coefficient of proportionality.

Similarly, assumption 2 leads to the equation:

$$N(1,n+1)-N(1,n)=\delta-K(n)\cdot N(1,n). \quad (3.13.7)$$

From the condition that the probability of generating $(n+1)$ the th word is 1, we have:

$$\sum_f K(n)\cdot f\cdot N(f,n)+\delta=K(n)\sum_f f\cdot N(f,n)+\delta=1\,. \quad (3.13.8)$$

Given what $\sum_f f\cdot N(f,n)+\delta=n$, we have:

$$K(n)=\frac{1-\delta}{n}. \quad (3.13.9)$$

In addition, one more assumption is introduced, which consists in the fact that for all f the following is true:

$$\frac{N(f,n+1)}{N(f,n)}=\frac{n+1}{n}. \quad (3.13.10)$$

It follows from the last assumption that

$$\frac{N(f,n)}{N(f-1,n)}=\frac{N(f,n+1)}{N(f-1,n+1)}=\varphi(f). \quad (3.13.11)$$

In this case, $\varphi(f)$ it does not depend on n and, taking into account the previous equations:

$$\varphi(f)=\frac{(1-\delta)(f-1)}{1+(1-\delta)f)}. \quad (3.13.12)$$

Passing to the function $N(f)=N(f,n)/n$ for $\varphi(f)$, we have:

$$\frac{N(f)}{N(f-1)}=\varphi(f). \quad (3.13.13)$$

Using the last equation $f-1$ once we get:

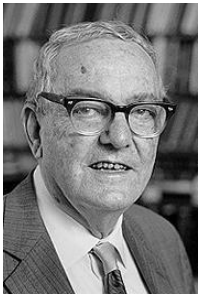
$$\begin{aligned} &N(f)=\varphi(f)N(f-1)= \\ &=\varphi(f)\varphi(f-1)\cdots\varphi(2)N(1). \end{aligned} \quad (3.13.14)$$

By introducing the notation, $\rho = 1/(1-\delta)$ the last equation can be rewritten:

$$N(f) = \frac{(f-1)(f-2)\cdots 2\cdot 1}{(f+\rho)(f+\rho-1)\cdots(\rho+2)} N(1) = \frac{\Gamma(f)\Gamma(f+2)}{\Gamma(f+\rho+1)} N(1). \quad (3.13.15)$$

Considering that $\frac{\Gamma(f)}{\Gamma(f+\rho)} \sim f^{-\rho}$ at $f \to \infty$ and denoting $\beta = 1+\rho$, we finally have:

$$N(f) \sim f^{-\beta}. \quad (3.13.16)$$



G.A. Simon (Herbert A. Simon, 1916-2001)

The Zipf distribution is often distorted in practice due to insufficient volumes of text corpora, which leads to the problem of estimating the parameters of statistical models. On the other hand, the relationship between rank and frequency was taken by Salton in 1975 [116] as a starting point for choosing terms for indexing. Next, he considered the idea of sorting words according to their frequency in the corpus. As a second step, high frequency words can be eliminated because they are not good distinguishing features for collection documents. In the third step, terms with a low frequency defined by some threshold (for example, words that occur only once or twice) are removed because they occur so infrequently that they are rarely used in user queries. Using this approach, you can significantly reduce the size of the search engine index. A more principled approach to the selection of index terms – taking into account their weight values. In weight models, mid-frequency terms turn out to be the most significant, since they are the most significant in the selection of a particular document (the most frequent words occur simultaneously in a large number of documents, and low-frequency terms may not be included in documents that are of interest to the user).

Another empirical law formulated by Zipf is that the number of meanings of a word correlates with the square root of

its frequency. It was assumed that infrequently used words were less ambiguous, which confirms that high-frequency words are not suitable for indexing information retrieval systems.

Zipf also determined that the length of a word is inversely proportional to its frequency, which can be easily verified by simply analyzing the list of function words. The last law really exemplifies the principle of economy of effort: shorter words require less effort to reproduce, and thus are used more frequently. This "law" can be confirmed by considering the above model of word generation by a monkey. It is easy to see that the probability of generating a word decreases with length, the probability of a word from n non-whitespace characters is:

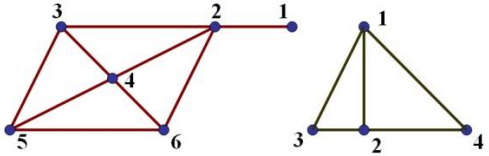
$$(1-p)^n \cdot p, \qquad (3.13.17)$$

where p is the probability of generating a gap.

Although Zipf's law gives interesting general characteristics of words in corpora, in general some limitations of its applicability in obtaining the statistical characteristics of collections of documents consisting of many independent documents of different authors have been noticed. Zipf's laws are satisfied not only by words from one text, but by many objects of the modern information space.

Questions to control

1. Give examples of graphs that (a) can and (b) cannot be bypassed without passing through any connection twice.
2. Formulate the concept of a social network.
3. List the main characteristics of complex networks.
4. What characteristics of Erdős-Rényi networks do you know?
5. What is the average minimum path (SP)?
6. How does the average minimum path (SP) for a) a square grid, b) an Erdős-Rényi network, c) a small Watts-Strogatz world depend on the size of the network?
7. What is the clustering coefficient of this node? What property of the network does it characterize?
8. Find the clustering coefficient for the nodes of the depicted graphs



9. In which case is Efficiency a better characterization of network properties than average shortest path (SP)? Give a specific example.
10. How is the average shortest path (SP) calculated for a network with link weights?
11. How is the number of nearest m -th neighbors related z_m to z_1 and z_2 ?
12. Formulate the Molloy-Read criterion.
13. Write down the node degree distribution function for the Erdős-Rényi network in the case of a large number of nodes.
14. Write down the node degree distribution function for a scale-invariant (SF) network in the case of a large number of nodes. What is the peculiarity of the normalization of this function?
15. Give an example of a deterministic scale-invariant network.
16. Draw several construction steps (u, v) – a flower for (1,2), (1,3), (1,4), (2,2), (2,3), (3,3).

17. What is the difference between algorithms for building a small world (small world) in the Watts-Strogatz and Watts-Strogatz-Newman models?

18. Algorithm for constructing a percolation network.

19. What is the percolation threshold?

20. What is the percolation threshold for a one-dimensional lattice? For infinitely dimensional?

21. What is common and different in the properties of an infinite (giant) cluster in a percolation and complex network?

22. Find Padé approximants for the following functions:

$ctg(x), \quad x \approx 0$

$1/\cos(x), \quad x \approx \pi/2$

$1/\sin(x), \quad x \approx 0, \quad x \approx \pi$

23. Derive the Cauchy distribution – (3.4.5).

24. Show that Planck's formula for thermal radiation (3.5.12) satisfies the scaling relation (Wien's law) (3.5.14).

25. What is the order parameter?

26. Formulate the main provisions of the phenomenological theory of phase transitions of the second kind.

27. Define an infinite cluster.

28. Draw (qualitatively) the dependence of the density of an infinite cluster on the concentration of bonds (nodes).

29. What are the grounds for asserting that the percolation transition is a second-order phase transition?

30. Fundamentals of cluster analysis. Explain the algorithm of hierarchical grouping-union.

31. Fundamentals of cluster analysis. *k-means* method. What clustering quality function is maximized by this algorithm?

32. Explain the basic concept of cellular automata. Give an algorithm for the information diffusion model.

33. Give the definition and algorithm for calculating the Hurst exponent. How does the fractal dimension of a self-similar time series correlate with the Hurst exponent ?

34. Define a linear classification problem. Write and explain Rocchio's formula for calculating the vector-profile of a category C_i.

35. Give the formula and algorithm for calculating the autocorrelation function.

36. Give the main distance metrics between documents used in cluster analysis.

37. Boolean i model by isca. Give a definition of DNF and bring the query to this form – a logical expression : $q = a \wedge (b \vee \neg c)$. How many disjunction operations will be used in the resulting DNF?

38. Write a formula for calculating the proximity measure of a document and a query in accordance with the vector-spatial search model.

39. Give a Bayesian criterion for a document to belong to a certain category.

40. Derive a formula for calculating the search status in accordance with the probabilistic search model.

41. Support Vector Machine (SVM). The dividing strip is given by the system of inequalities:

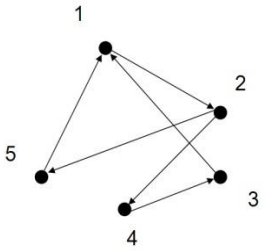
$$3x - 4y > 4;$$
$$3x - 4y < 6.$$

What is the width of this band?

42. Search/classification quality indicators. Write formulas for calculating the completeness, accuracy and F – measures of information retrieval. Calculate the F – measure if the recall value c is 0.8 and the precision value is 0.6.

43. Latent Semantic Indexing Method (LSA / LSI). What is the singular decomposition of a matrix s? Application in the LSA / LSI method.

44. Write a system of equations for recursive counting authorship and mediation coefficients _ to algorithm e HITS and also carry out the calculation (4 and iterations) for the network :



45. Lead simulation and simulation model and iterative formula for calculating the parameter a PageRank.

46. Classic definition $TF \cdot IDF$. Term w_i meets in document e $d^{(j)}$ with a normalized frequency equal to 0.5. Number of documents w_i in the data array containing the term is 20 and the total number of documents in the array is 1000. What is the value of $TF \cdot IDF$?