Effective Choice and Ranking of Alternatives in Search and Recommendation Problems

Big Data Analysis and Data Mining
Paris, September, 7-8, 2017

Fuad Aleskerov
(National Research University Higher School of Economics,
Institute of Control Sciences, Russian Academy of Sciences
alesk@hse.ru, alesk@ipu.ru)

Joint work with

Evgeny Mitichkin (University of Mannheim),

Sergey Shvydun (HSE, ICS RAS),

Vyacheslav Yakuba (ICS RAS)

Presentation structure

- The problem
- Selection procedures used
- Choice procedures and their complexity
- Super-threshold procedure
- Superposition of choice procedures
- Application and comparison

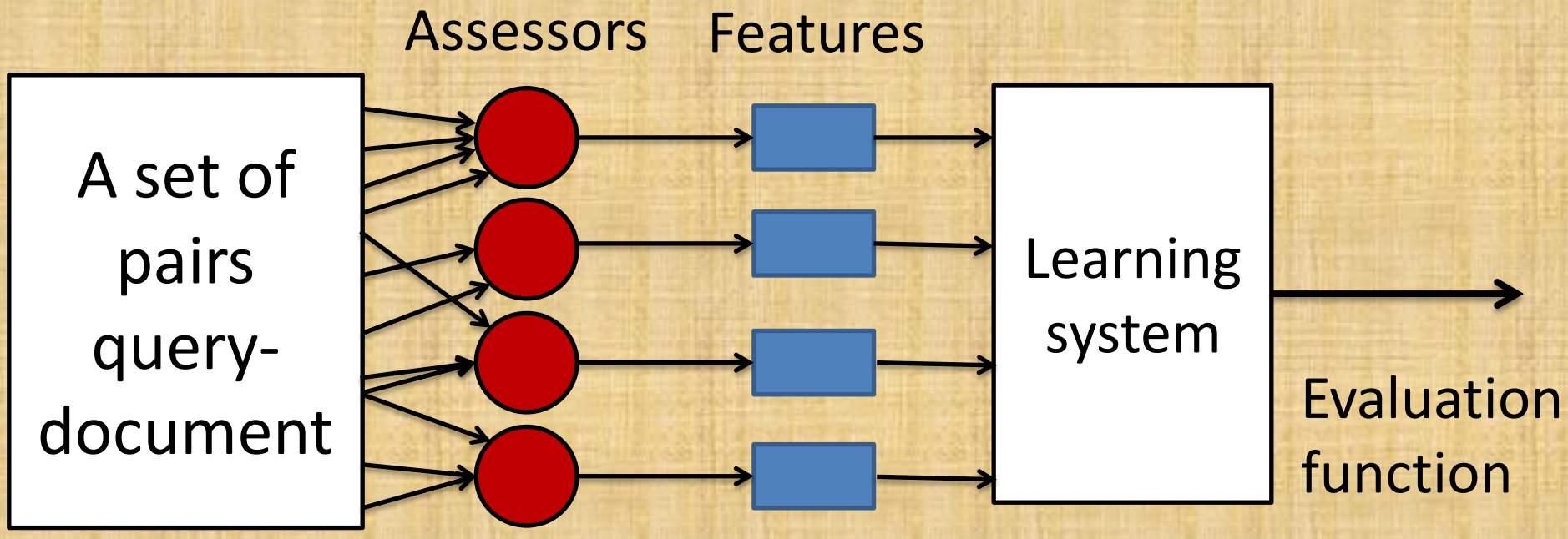
The problem

- Exponential growth of amounts of data and number of nodes in the Internet;
- Algorithms used in most of search engines use obsolete approaches, which can not meet modern requirements of the environment;
- Most of search engines today are human-sensible.

Urgency of the problem

- The problem emerged long ago, but it has not been solved yet;
- Growth of IT-industry and amounts of data to proceed;
- Necessity of transformation of the approaches to the problem of search.

Learning system



Query X	Document Z

y

Evaluation

Y=F(x,z)

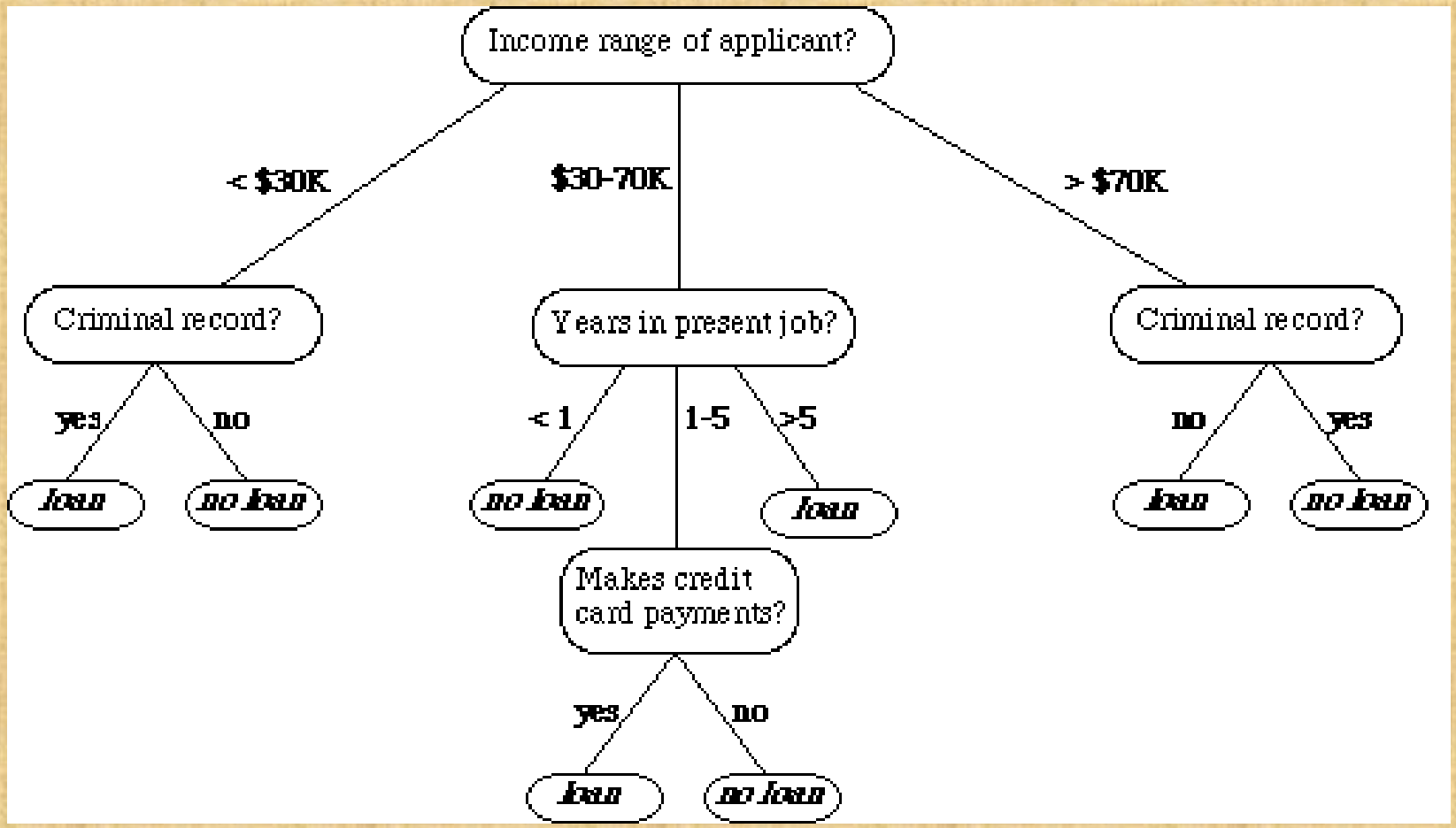
Decision Trees

A decision tree depicts rules for dividing data into groups.

The first rule splits the entire data set into some number of pieces, and then another rule may be applied to a piece, forming a second generation of pieces. In general, a piece may be either split or left alone to form a final group.

Decision Tree Example

- *Should we offer a loan to a person?*



Well-Known Algorithms

- **AID, SEARCH**

AID finds binary splits on ordinal and nominal inputs that most reduce the sum of squares of an interval target from its mean. The program stops when the reduction in sum of squares is less than some constant times the overall sum of squares (the constant is 0.006 by default).

- **CHAID (Chi-Squared Automatic Interaction Detection)**

CHAID recursively partitions the data with a nominal target using multiway splits on nominal and ordinal inputs. A split must achieve a threshold level of significance in a chi-square test of independence between the nominal target values and the branches, or else the node is not split. The search ends when no more merges or re-splits are significant.

Well-Known Algorithms

- **Classification and Regression Trees**
  - The program creates binary splits on nominal or interval inputs for a nominal, ordinal, or interval target.
  - An exhaustive search is made for the split that maximizes the splitting measure:
    - *For an interval target* - reduction in square error or mean absolute deviation from the median.
    - *For an ordinal target* - "ordered twoing" (modification of towing index).
    - *For a nominal target* - reduction in the Gini index and "twoing."
  - *Stopping rule* - cost-complexity pruning: a large tree is created, then a subtree is found, then another sub-tree within the first, and so on forming a sequence of nested sub-trees, the smallest consisting only of the root node. A subtree in the sequence has the smallest overall cost among all sub-trees with the same number of leaves. The final pruned tree is selected from subtrees in the sequence using the costs estimated from an independent validation data set or cross validation.

Well-Known Algorithms

- **ID3, C4.5, C5**
  - The split is chosen that maximizes the gain ratio:

$$Gain\ ratio = \frac{reduction\ in\ entropy}{entropy\ of\ split}$$

  - *Stopping rule* - "pessimistic" pruning. In each node, an upper confidence limit of the number of misclassified data is estimated assuming a binomial distribution around the observed number misclassified. The confidence limit serves as an estimate of the error rate on future data. The pruned tree minimizes the sum over leaves of upper confidences.
- **QUEST**
  - This algorithm selects an input variable to split on before searching for a split, thereby eliminating the time required for searching on most inputs and eliminating the bias towards nominal inputs inherent when relying on candidate splitting rules to select the input variable.

Well-Known Algorithms

- **OC1 (Oblique Classifier 1)**

Choose an initial linear combination at random. Apply a heuristic hill-climbing algorithm to find a local optimum. Make a random change or a random restart and climb again.

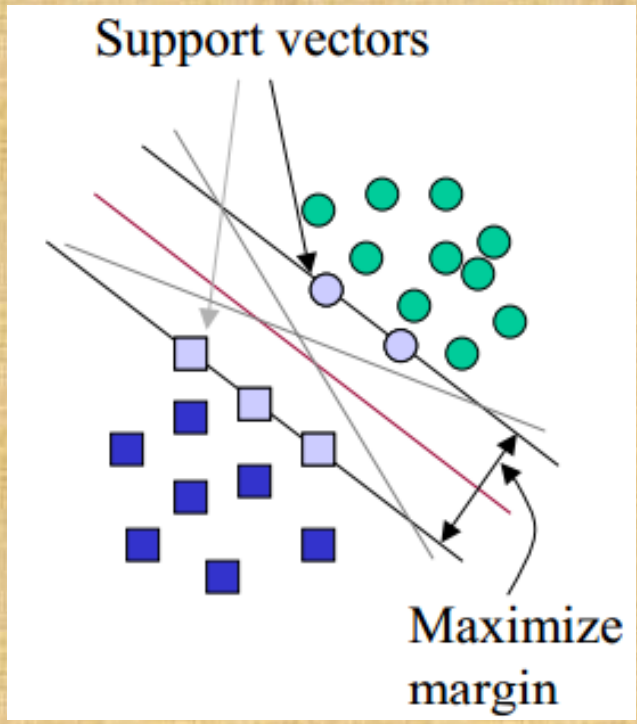
- **SAS algorithms**
  - Splits can be evaluated as a reduction in impurity (least squares, Gini index, or entropy) or as a test of significance (chi-square test or F test).
  - The user specifies the maximum number of branches from a split, thus allowing binary trees, bushy trees, or anything in between.
  - The limited search on an input begins with a split into many branches and then proceeds in steps of merging pairs of branches.
  - The heuristic search also considers switching input values between branches after every merge, thereby examining many more candidates than just merging.

Disadvantages of Decision Trees

- *Over-fitting:* They may fit the data well but then predict new data worse than having no model at all.
- *Instability*: They may fit the data well, predict well, and convey a good story, but then, if some of the original data are replaced with a fresh sample, a completely different tree may emerge using completely different inputs in the splitting rules and consequently conveying a completely different story.
- The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts. Consequently, practical decision-tree learning algorithms are based on heuristic algorithms which cannot guarantee to return the globally optimal decision tree.

Support vector machine (SVM)

- SVM is a discriminative classifier formally defined by a separating hyperplane
- SVM *maximize the margin* around the separating hyperplane.
- The decision function is fully specified by a subset of training samples, the support vectors.



Support vector machine (SVM)

Advantages:

- *Effective* in high dimensional spaces.
- Still effective in cases where the number of dimensions is greater than the number of alternatives.
- Uses a subset of training points in the decision function, so it is also *memory efficient*.
- *Versatile*: different functions can be specified for the decision function.

Disadvantages:

- If the number of features is much greater than the number of alternatives, the method is likely to give *poor performances*.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

Search Problem

Consider a finite set A of alternatives evaluated by *n* criteria, i.e., the vector of criterial values (x_1,…,x_n) is assigned to each alternative *x* from A

$$x \in A \rightarrow (x_1, \ldots, x_n).$$

The problem lies in constructing a transformation φ — the rule of aggregation over A — such that φ: $\mathrm{A} \rightarrow \mathrm{R}^1$.

Choice (aggregation) procedures

- We know 5 groups of choice procedures
  - a) Scoring (positional) rules;
  - b) Rules, using majority relation;
  - c) Rules, using value function;
  - d) Rules, using tournament matrix;
  - e) q-Paretian rules.
- In total, about 30 procedures

An example

- **Borda rule**

$$x \in A, P_i \in \vec{P}, r_i(x, \vec{P}) = card(L_i(x))$$

$$a \in C(\vec{P}) \Leftrightarrow \left[\forall b \in A, r(a, \vec{P}) \geq r(b, \vec{P}) \right], r(a, \vec{P}) = \sum_{i=1}^{n} r_i(a, P_i)$$

However, this rule does not satisfy non-compensation principle.

- ***Simpson's Procedure (Maxmin Procedure)***

$$\forall a, b \in X, S^+ = \left(n(a,b)\right)$$

$$n(a,b) = card\left\{ i \in N \mid aP_i b \right\}, \quad n(a,a) = +\infty$$

$$x \in C(\vec{P}) \Leftrightarrow x = \arg\max_{a \in A} \min_{b \in A} (n(a,b))$$

Complexity

- ***Borda rule***

$$O\left(n \cdot (k+1)\right),$$

where n – number of alternatives, k – number of criteria.

- ***Simpson's Procedure (Maxmin Procedure)***

$$O\left(n^2 \cdot (\frac{k}{2}+1)+n\right),$$

where n – number of alternatives, k – number of criteria.

Super-threshold choice model

a formalization of H.Simon's idea

Aizerman M., Aleskerov F. Theory of Choice, Elsevier, North--Holland, 1995, ISBN 0 444 822100, 314 pp.

Super–threshold choice rule:

$$\pi_{st}: x \epsilon \mathrm{C}(\mathrm{X}) \Leftrightarrow (x \epsilon \mathrm{X} \;\&\; x_i \geq V(X)).$$

The main idea of the super-threshold superposition approach is to consider the consistent application of super-threshold procedures to different criteria in order to find the best alternatives.

How to define V(X)

- Mean value:

$$V(X) = \frac{1}{|X|} \cdot \sum_{x \in X} u(x)$$

Otherwise speaking, everything above average is acceptable.

There are many other ways to define V(X).

Complexity of super-threshold rule

- Obviously, it is linear
- So, we can process huge amount of information
- But, in the search problem the alternatives are evaluated via a set of criteria
- We need one more step

‘Classic’ Model

$$C(X) = \{y \in X \mid \bar{\exists} x : u(x) - u(y) > \varepsilon(X)\}$$

$$xP_X y \Leftrightarrow u(x) - u(y) > \varepsilon(X)$$

Equivalent Models

- Theorem 1. Choice function is rationalizable as (1) iff WARP holds

- Theorem 2. Super-threshold choice is equivalent to (1)

Superposition of Choice Procedures

- The very idea of superposition of functions goes back to the 13th Hilbert's problem on superposition of real-valued functions
- Answer was given by A.N.Kolmogorov and V.I.Arnold in 1956
- I studied the problem in 1988-89 from different point of view – that of choice functions (Aizerman M., Aleskerov F. Theory of Choice, Elsevier, North--Holland, 1995, ISBN 0 444 822100, 314 pp.)
- Next attempt was made in 2008 by Y.Cinar and myself (Aleskerov F., Cinar Y. 'q-Pareto- scalar' Two-stage Exstremization Model and its Reducibility to One-stage Model,' Theory and Decision, 65, 2008, 291-304)

Formal statement of the problem of superposition of choice functions

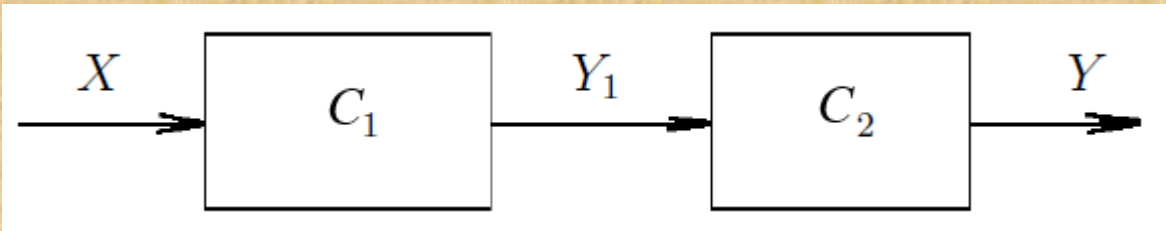
Initial criteria: $u_1(\cdot), u_2(\cdot), \ldots, u_n(\cdot)$

Choice functions: $C_1(\cdot), \ldots, C_n(\cdot)$

Superposition of choice functions:

$$C(\cdot) = C_1(C_2(\ldots C_n(\cdot)))$$

Two–Step Choice Mechanism



Depending on whether C_1 and C_2 are single–criterion or multi–criteria extremizational mechanisms, the following four cases occur:

1. scalar–scalar mechanism;
2. scalar–vector mechanism;
3. vector–scalar mechanism;
4. vector-vector mechanism.

Axioms

- *Heritage condition* (Condition **H**):

$$\forall X, X' \epsilon 2^A,\ X' \subseteq X \Rightarrow C(X') \supseteq C(X) \cap X';$$

- *Concordance* (Condition **C**):

$$\forall X', X'' \epsilon 2^A \rightarrow C(X' \cup X'') \supseteq C(X') \cap C(X'');$$

- *Independence of Outcast of options* (Condition **O**):

$$\forall X, X' \epsilon 2^A,\ X' \subseteq X \backslash C(X) \Rightarrow C\left(X \backslash X'\right) = C(X).$$

- *Monotonicity condition*

$$\forall X \epsilon 2^A, x \epsilon \mathrm{C}\left(\vec{\mathrm{P}}\right), \forall \vec{P}, \vec{P}':$$

$$(\forall a, b \in X, aP_i b \Leftrightarrow aP_i' b \ \&\ \exists y \in X, yP_i x \Rightarrow xP_i' y)$$

$$\Rightarrow x \epsilon \mathrm{C}\left(\vec{\mathrm{P}}'\right).$$

Examples Of Two–Step Choice Mechanisms

1. **Maxmin Procedure – Borda Procedure**

The rule does not satisfy Conditions **H**, **C**, **O** and *Monotonicity condition*.

2. **Borda Procedure – Maxmin Procedure**

The rule does not satisfy Conditions **H**, **C**, **O** and satisfies *Monotonicity condition*.

Application: Types of search problem

1. Information about the utility of alternatives **is not specified**. In this case, the threshold values and the order of elimination are normally defined manually.
2. Information about the utility of **some alternatives is specified**. The task is to choose items in a new, unseen list of alternatives. In this case the threshold level as well as the order of procedures should be defined automatically.

Super-threshold superposition procedures

Two methods of defining the order of elimination and threshold values are proposed:

1. Super-threshold etalon-based procedure,
2. Super-threshold procedure based on the distribution function.

1. Super-threshold etalon-based procedure

The procedure is divided into two steps:

1) Selection of the etalon criterion:

$$\varphi^* = \{\varphi_i \epsilon \varphi \Big| \forall j, j \neq i: \max_{x \in A^*} \varphi_i(x) - \min_{x \in A^*} \varphi_i(x) \leq \max_{x \in A^*} \varphi_j(x) - \min_{x \in A^*} \varphi_j(x)\},$$

where φ is the set of criteria, A^* is the set of the high-utility alternatives, $A^* \subset A$.

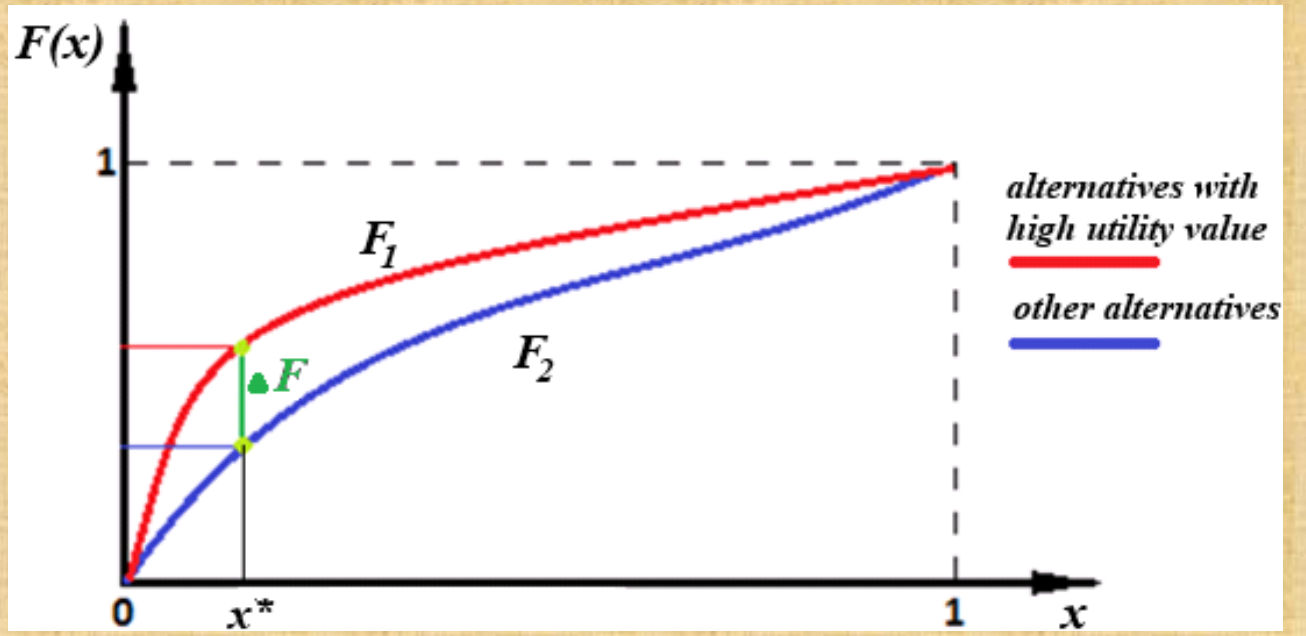
2) Construction of the order of the eliminations

Computational complexity: $O(n \cdot k \cdot \log(k))$, where n – the number of alternatives, k – the number of criteria.

2. Super-threshold procedure based on the distribution function

Two groups – alternatives with high utility value and all other alternatives.

Each group is presented by a set of distribution functions based on all criteria.



Threshold value ***x**:***

$$\Delta F = \max_{x \in A} |F_1(x) - F_2(x)|,$$

where $F_1(x), F_2(x)$ are the values of distribution functions depending on *x*.

Computational complexity: $O(k \cdot (n + k))$, where n – the number of alternatives, k – the number of criteria.

Application of super-threshold procedures to the search problem

- LETOR 4.0 collection was chosen (1175000 documents, 136 factors in the dataset);
- There were performed some procedures of data analysis and clustering;
- The first method showed better result on small dataset, the second demonstrated higher results in the case with different choice rules.

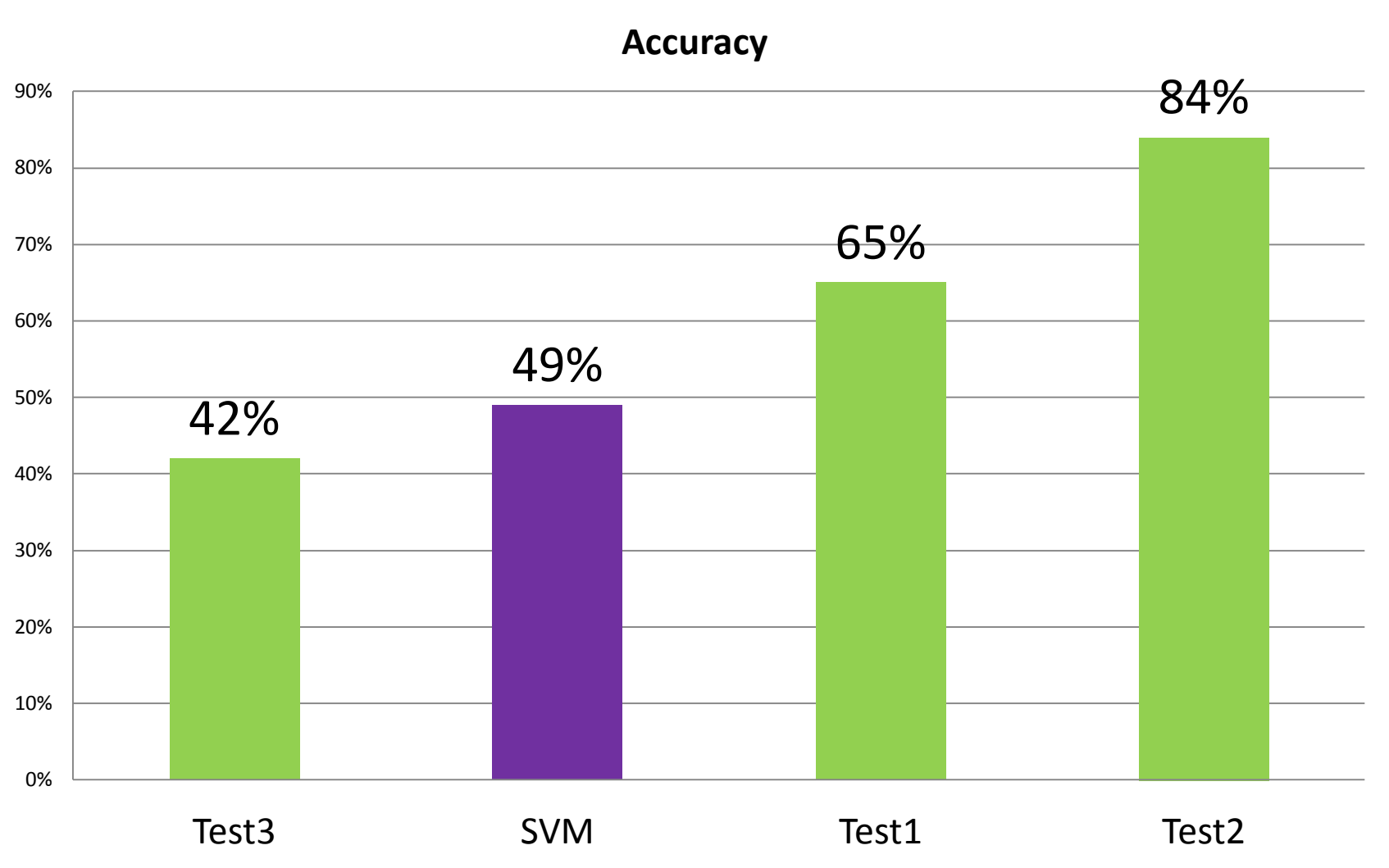
Comparison with Support vector machine (SVM)

***Test 1*:** Super-threshold etalon-based procedure distinguished relevant (4 and 3) documents from irrelevant

***Test 2*:** Super-threshold etalon-based procedure distinguished documents with a relevance value of 4 and other objects.

Test 3: Super-threshold procedure based on the distribution function distinguished relevant (4 and 3) documents from irrelevant.

Comparison with Support Vector Machine (SVM)



Conclusion

- Superposition of super-threshold procedures are proposed to be used for the search problem with large number of alternatives characterized by a set of criteria.
- Two methods of defining the threshold and the elimination order in learning to rank problem are proposed.
- Both procedures were tested on the LETOR 4.0 dataset. The experiments showed that the goal to develop self-learning choice procedures for the choice of alternatives is achieved.

Patents, certificates

- Copyright certificate for software #2013618228 ‘Choice and ranking of alternatives using superposition of positional rules’, Russian Federation, 04.09.2013 (with E.Mitichkin, S,Shvydun, V.Yakuba)
- Certificate Aleskerov F., Mitichkin E., Chistyakov V., Shvydun S., Iakuba V. Method for selecting valid variants in search and recommendation systems (variants), World Intellectual Property Organization, Patent Scope, Publication Number WO/2014/148948, International Application Number PCT/RU2013/001180, Publication Date 25.09.2014, International Filing Date 27.12.2013,
- https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2014148948
- Certificate Aleskerov F., Kalugina E., Shvydun S. for the computer software “Efficient selection of the personnel using the procedure of the threshold aggregation”, Federal Organization of Intellectual Rights, Russian Federation, # RU 2014612447, 20.03.2014
- Aleskerov F.T., Mitichkin E.O., Chistyakov V.V., Shvydun S.V., Iakuba V.I. Patent #2543315 Method for selecting valid variants in search and recommendation systems (variants), Priority 22.03.2013, Registered in the State Register of Russian Federation 27.01.2015

Thank you!