Ministry of Education and Science of the Russian Federation
Moscow Institute of Physics and Technology (State University)
Department of Control and Applied Mathematics
Chair of Intelligent Systems
Dorodnicyn Computing Centre of RAS

Zhuikov Vladimir Vladimirovich

Shopping recommendation system based on metric analysis of clothing descriptions

010900 – Applied mathematics and physics

Master’s thesis

Research (Thesis) advisors:
Principal Investigator at the
CCAS, DSc
Strijov Vadim Viktorovich

Distinguished Career Professor
Carnegie Mellon University,
School of Computer Science,
Language Technologies Institute
Anatole Gershman

Moscow
2015

Abstract

The thesis presents a shopping recommendation system based on metric analysis of clothing descriptions. The developed system ranks the catalog of clothing and offers corresponding items to the user's request while, at the same time, selecting the most diverse items.

An algorithm for ranking is developed. Based on the request, the recommendation system finds the distance from this request to all documents from the collection of data. The request and the collection of data are sets of features. The system ranks the results in accordance with the following rules: minimizes the distance from the query to the relevant results, maximizes the distance from the query to the irrelevant results and maximizes the distance between the relevant query results. For ranking, Heterogeneous Euclidean-Overlap Metric (HEOM) of clothes catalogue items is used. HEOM metric uses different attribute distance functions to measure distances between objects in mixed scales.

A dataset of clothes catalogue items is collected. The system, in addition to the basic attributes given as text descriptions of clothing, uses attributes based on expert description such as fashion, psychological age and attractiveness. The dataset has features of text, linear and nominal scales.

The computational experiment shows the effectiveness of the proposed algorithm. The importance of features of the collection of data is defined. A software product demonstrating the recommendation system in action is developed.

Thesis Advisor: Vadim V. Strijov, D.Sc.
Title: Principal Investigator at the CCAS

Thesis Advisor: Anatole Gershman, Ph.D.
Title: Distinguished Career Professor, Carnegie Mellon University, School of Computer Science, Language Technologies Institute

List of abbreviations

ACM	– Association for Computing Machinery
CF	– Collaborative Filtering
CSV	– Comma Separated Values
E-commerce	– Electronic commerce
HEOM	– Heterogeneous Euclidean-Overlap Metric
HTML	– HyperText Markup Language
Icam	– Integrated Computer Aided Manufacturing
IDEF0	– Icam DEFinition for Function Modeling
Matlab	– Matrix laboratory
NLPQL	– a Fortran Implementation of a Sequential Quadratic Programming Algorithm with Distributed and Non-Monotone Line Search
NPSOL	– Nonlinear programming of Systems Optimization Laboratory
NUI	– Natural user interface
OMCS	– Open Mind Common Sense
Python	– high-level programming language
SQP	– Sequential quadratic programming
TF.IDF	– Term Frequency–Inverse Document Frequency

Content

Introduction

Research relevance. Online shopping gains increasing popularity [1]. More than half (62%) of US consumers with Internet access now shop online at least once a month, and just 1% say they never shop online, according to a recent report by Walker Sands [2]. One of the most popular shopping items is clothing. From all users buying items online, 63% of them buy clothes [3]. The statistics reveal that women are much more likely to research online and buy offline, with 71% of women doing this, compared to 52% of men [4]. There are many online shopping sites: more than 39 000 in 2013 in Russia [5]. Studies on clothing are receiving increasing interest mainly due to the huge market related to clothing. In China, the potential market is expected to break 20 billion US dollars in 2016 [6]. Such huge market prospects greatly motivate clothing relevant research.

Recommendation systems have a significant impact on the improvement of online shopping services. They use searching technics that suggest desired or similar clothing features from online shopping databases. Current systems rely either on shop statistics (collaborative filtering) or on simple key word matching. These systems do not take into account such important considerations as style, fashion, age and importance of the features for users. In this work, we implemented a recommendation system based on metric analysis of clothing descriptions, including style, fashion, age, text description, pictures.

Purpose. To develop a shopping recommendation system. The system ranks the search results by similarity to the query while, at the same time, selecting maximally diverse features. Determine the importance of features in mixed scales.

Research methods. For extracting features, we use algorithms of clustering. For ranking, we use HEOM (Heterogeneous Euclidean-Overlap Metric), optimization algorithms. We use expert annotations collected from users (22 women) and presented in nominal scales. A part of expert annotations is used for system tuning and another part for testing. We analyze the importance of features in mixed scales. For parsing the collected dataset, we use Scrapy and ntlk library of Python. For the topic modelling tasks we use gensim library of Python. For visualization, we use kivy – cross-platform Python framework for NUI development. For the final algorithms, we use Python and Matlab.

Novelty. Developed the ranking algorithm using HEOM metrics taking into account the features' importance.

Value. Developed shopping recommendation system based on metric analysis of clothing descriptions, that:

- ranks metric objects on requests;
- uses expert evaluations define importance of features;
- visualizes results.

Aspects for the defense

1. The algorithm of the shopping recommendation system is based on metric analysis of clothing descriptions. The system ranks the search results by similarity to the query while, at the same time, selecting maximally diverse features. The algorithm is based on HEOM metric for the mixed scales.
2. Minimax problem. Solve the optimization problem. Analyze the importance of features in mixed scales, explain the results and give recommendations.
3. Software prototype. Represent the working prototype of the shopping recommendation system, IDEF diagrams of the technical system.

Outline

In the first part of this thesis, we present the analysis of types of the recommendation system and the analysis of existing clothing recommendation systems. We include four main types of recommendation systems: collaborative filtering, content-based filtering, hybrid filtering and mobile recommendation systems. We describe clothing recommendation systems based on reasonable computing, concrete attribute, web mining and eleven more technics commonly used in recommendation systems in e-commerce. Based on this analysis, we build the algorithm that makes use of Heterogeneous Euclidean-Overlap Metric and TF.IDF, gradient descent and sequential quadratic programming optimization methods.

In the second part, we present the problem statement, the initial hypothesis, the data structure, where the dataset consists of 1 text data, 20 liner scales and 14 nominal scales.

In the third part, we present the implementation of the algorithm of the shopping recommendation system with a solution to the minimax problem. The most important features for users are “description”, “acrylic”, “cotton”, “subtype”, “color” and “brand”.

The fourth part contains the block diagram and IDEF0 diagrams of the shopping recommendation system. The block-diagram is made up of seven main blocks and three auxiliary blocks. The IDEF0 consists of two levels of decomposition.

In the fifth part, we perform the computational experiment, which defines the quality criteria: P@20, MAP, nDCG.

The prototype of the system has been developed and tested. The system, based on our algorithm, exhibits better results for all quality criteria.

1. Recommendation systems: achievements and problems to solve

1.1. Recommendation System

Recommendation System is a software tool and techniques that provide suggestions for items to be of use to a user [7]. The suggestions relate to various decision-making processes, such as what items to buy, what music to listen, what video to watch, or what online news to read. “Item” is a general term used to mean what the system recommends to users.

The goal of a Recommendation System is to generate meaningful recommendations to a collection of users for items or products that might interest them.

Recommendation systems account for one of the major parts of e-commerce ecosystem. They are a good technique to enable users to sieve through large information and product spaces. Nearly twenty years of research on collaborative filtering have resulted in a mixed set of algorithms and a rich collection of tools for estimating their efficiency. Research in this sphere brings about a better understanding of the way recommendation technology may be applied in specific domains. The differing personalities demonstrated by different recommendation algorithms, indicate that recommendation is an extremely complicated problem. Specific tasks, information needs, and item domains are individual problems for recommenders together with design and evaluation of recommenders are to be performed based on the user tasks to be supported. Efficient deployments should start with deep analysis of prospective users and their objectives. On the basis of this analysis, system designers have a host of options for the choice of algorithm and for its implementing in the surrounding user experience [8].

Recommendation systems vary in the way they analyze these data sources to develop notions of affinity between users and items, which can be used to identify well-matched pairs.

Most recommendation systems use four basic approaches [9]:

– collaborative filtering;
– content-based filtering;
– hybrid techniques;
– mobile recommendation systems.

Collaborative Filtering Recommendation Systems

Collaborative Filtering (CF) refers to a class of techniques used in recommendation systems. These techniques recommend items to users that other users with similar tastes have liked in the past. CF methods are sub-divided into model-based and neighborhood-based approaches. Model-based approaches assume an underlying structure to users’ rating behavior, and induce prognostic models based on the past ratings of all users, in contrast, in neighborhood-based approaches, a subset of users are chosen based on their similarity to the active user, and a weighted combination of their ratings is used to produce predictions for this user [9].

Content-based Filtering Recommendation Systems

Content-based filtering (CB) is popular in information retrieval. In CB, the text and multimedia content of documents is used to select documents, which applicable to a user's query. In the context of recommendation systems, this refers to content-based recommendations that provide recommendations by comparing representations of content telling an item to representations of content that interests a user [9].

Hybrid Recommendation Systems

Hybrid technics are methods that combine content-based and collaborative filtering methods. Claypool [10] uses hybrid technics with an adaptive weighted average. Weight of the collaborative component increases as a number of users retrieving an item increases. Pazzani's approach [11] uses user profile as representation by a vector of weighted words derived from positive training example with the Winnow algorithm. Several hybrid technics are a classification task. They incorporate collaborative elements in this task.

Mobile Recommendation Systems

Smart phones, mobile phones, tablets are becoming a primary platform for information access. Therefore, recommendation systems with these technologies can become key tools for mobile users for leisure and business applications. Recommendation systems for mobile devices increase the usability of mobile systems. It can provide personalized and focused content. For example, a mobile recommendation system is one that offers potentially profitable driving routes for taxi drivers in a city [12].

A recommendation system application has two classes of entities: users and features. Users have preferences for certain features. The data is represented as a *utility matrix* [12]. It gives for each user-features pair. Values of the matrix come from an ordered set. It could be either integers from one to five or integers zero and one, which users gave as a rating for the features (Figure 1.1.1).

	Features				
Users		1	2	3	4
	1	0	1	0	1
	…	1	1	1	1
	u	0	0	?	1
	…	1	1	1	
	n	1		0	

Figure 1.1.1 – User ratings matrix, where each cell $r_{u,i}$ corresponds to the rating of user *u* for item *i*.

A popular technique is to construct a utility matrix with some cells filled with known ratings. The goal is to predict unknown ratings. You can use Matrix Factorization to automatically infer *k* features for both users and items. This technique was used successfully in the Netflix.

The goal of a recommendation system is to predict the blanks in the utility matrix and the overall rating of an item.

Before discussing existing recommendation systems, let me introduce the *long tail* phenomenon that makes recommendation systems necessary [13].

Recommendations in the physical world is simple. It is not possible to adapt the store to each individual customer. For instance, a bookstore will display only the books that are most popular, and a newspaper will print only the articles it believes the most people will be interested in. In the first case, sales figures govern the choices, in the second case, editorial judgement serves [14].

The distinction between the physical and on-line worlds has been called the *long tail* [12] phenomenon. Physical institutions provide only the most popular items to the left of the vertical line, while the corresponding on-line institutions provide the entire range of items: the tail as well as the popular items [13].

The long-tail phenomenon forces on-line institutions to recommend items to individual users. It is not possible to present available items to the user, the way physical institutions can.

CF can perform in domains where there is not much content associated with items, or where the content is difficult for a computer to analyze, such as ideas, opinions. CF system has the ability to provide unexpected recommendations. It can recommend items that are relevant to the user, but do not contain content from the user's profile [9].

Challenges with recommendation systems

Taking advantage of the users, using collaborative filtering, has been made simpler with the data-collection opportunities the web affords. Unfortunately, the massive amounts of data also complicate this opportunity. For instance, although some users' behavior can be modeled, other users do not show typical behavior. These users can skew the results of a recommendation system and decrease its effectiveness. Users can exploit a recommendation system to favor one product over another, based on positive feedback on a product and negative feedback on competitive products. A good recommendation system should understand these issues.

Recommendation systems remain an active area of research, with a dedicated Association for Computing Machinery conference, intersecting several sub-disciplines of statistics, machine learning, data mining, artificial intelligence and information retrievals.

1.2. Review of existing clothing recommendation systems

This chapter discusses several existing research projects and models relate to Recommendation Systems in general and Clothing recommendations in particular.

Clothing recommendation systems based on reasonable computing

Edward Shen and Francis Lam form MIT Media Laboratory [15], created a recommendation system based on commonsense reasoning technology. Their recommendation system is a software agent comprising two sensors: a function sensor and a style sensor. For any input text, the style sensor suggests a suit for a wedding or jeans for a movie. The function sensor recommends a swimsuit for going to the beach. Both of the sensors provide recommendations by performing spreading activation in ConceptNet [16] with the data in OMCS (Open Mind Common Sense) [17], and handcrafted templates that provide function and style information for a set of types, brands, materials, and occasions. The accuracy of OMCS will be an uncertainty.

Clothing recommendation systems based on concrete attributes

Several recommendation systems have been proposed based on concrete attributes (e.g. weather, magazine data, street photography, pictures in web).

DailyDressMe [18] is a website that tells you what to wear based on the weather. It simplifies your daily routine of getting ready by using weather conditions in your region to accordingly suggest suitable outfits.

LookBook.nu [19] is a fashion, youth culture, and community website. It was inspired by street fashion website and blogs such as the The Sartorialist [20] and designed for users to post their own street-fashion photography, featuring themselves and their outfits.

Those systems only recommend some combinations of garments as references to improve fashion sense of people. They give different persons the same suggestion without using their own items. This leads to a problem, in most cases, even if you think the suggestion is great for you, probably, you do not have those items (or the same style items) in your wardrobe. It makes the suggestion not particularly valuable.

Style for Hire [21], a fashion startup co-founded by celebrity stylist Stacy London and Cindy McLaughlin. Aim of the system is to help the masses master the art of their own personal style. It allows users to find an experienced stylist in their city to help them choose clothing for a special event, or more. Users can search for stylists by type of style and more. The stylist’s goal is to help clients think about how to invest more strategically in their wardrobes, considering cost-per-wear, saving and spending habits and ultimately, to develop a wardrobe that is workable and wearable for any age, body type, lifestyle or budget.

Clothing recommendation systems based on Web Mining

Zeng's [22] system utilizes web-mining techniques to trace the customer's shopping behavior and learn his/her up-to-date preferences adaptively. In order to provide decision support for customers, one way to overcome the above problem is to develop intelligent recommendation systems to provide personalized information services. Based on these techniques, the system can trace the customer's shopping behavior and learn his/her up-to-date preferences adaptively.

The recommendation process consists of three phases as shown in figure 1.2.1. After the necessary data cleansing and transformation into the form usable in the system, target customer's preferences are mined first in phase 1.

In phase 2, different association rule sets are mined from the customer purchase database, integrated and used for discovering product associations between products. In phase 3, the system uses the match algorithm to match customer preferences and product associations discovered in the previous two phases, so the recommendation products list, comprising the products with the highest scores, are returned to a given target customer [23].

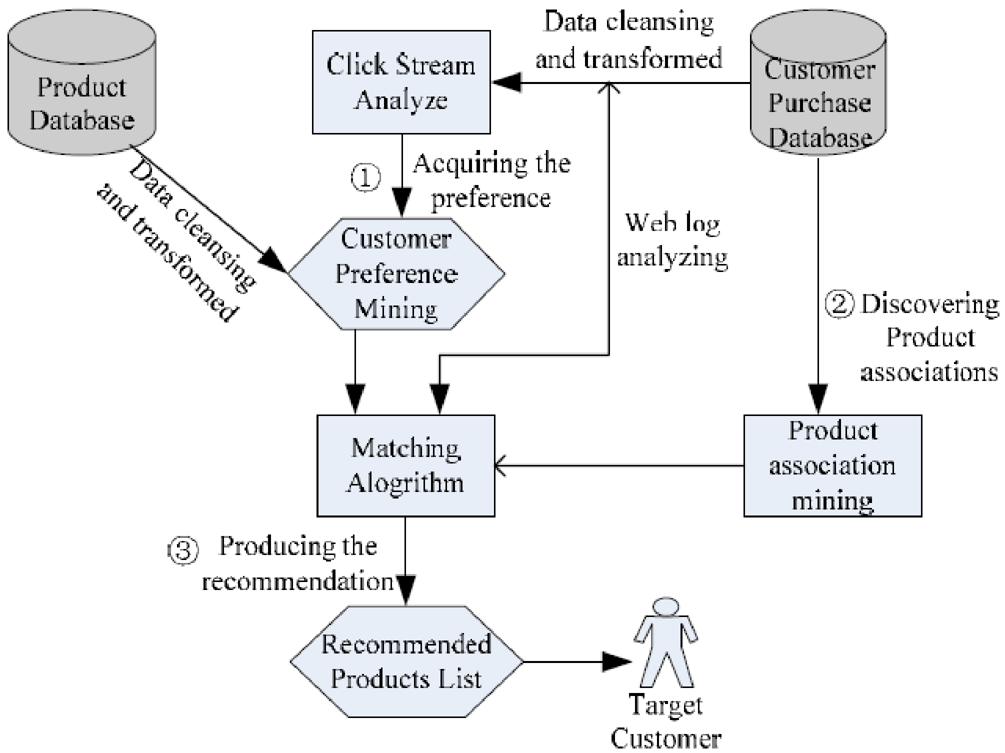


Figure 1.2.1 – Block-diagram of the clothing recommendation systems based on Web Mining

There are others types of clothing recommendation systems based on:

– the “wisdom of crowds” [24],
– web usage mining and decision tree induction [25],
– multimedia mining [26],
– knowledge base of product semantics [27],
– photographs from fashion magazines [28],
– active learning strategy [29],
– middle-level clothing attributes from a picture [30],
– analytical hierarchy process (AHP) [31],
– a modified Bayesian network [32],
– mining visual elements of different fashion styles [33],
– scenarios [34].

In the table 1.2.1, we show the leading recommendation systems of E-commerce [35]. Descriptions of the recommendation technologies and types of finding recommendations are in [35].

Table 1.2.1 – Recommendation systems in E-commerce

Business/Applications	**Recommendation Interface**	**Recommendation Technology**	**Finding Recommendations**
Amazon.com			
Customers who Bought	Top N List	Item to Item Correlation Purchase data	Organic Navigation
Eyes	Email	Attribute Based	Keywords/freeform
Amazon.com Delivers	Email	Attribute Based	Selection options
Book Matcher	Top N List	People to People Correlation Likert	Request List
Customer Comments	Average Rating Text Comments	Aggregated Rating Likert Text	Organic Navigation
CDNOW			
Album Advisor	Similar Item Top N List	Item to Item Correlation Purchase data	Organic Navigation Keywords/freeform

My CDNOW	Top N List	People to People Correlation Likert	Organic Navigation Request List
eBay			
Feedback Profile	Average Rating Text Comments	Aggregated Rating Likert Text	Organic Navigation
Levis			
Style Finder	Top N List	People to People Correlation Likert	Request List
Moviefinder.com			
Match Maker	Similar Item	Item to Item Correlation Editor's choice	Navigate to an item
We Predict	Top N List Ordered Search Results Average Rating	People to People Correlation Aggregated Rating Likert	Keywords/freeform Selection options Organic Navigation
Reel.com			
Movie Matches	Similar Item	Item to Item Correlation Editor's choice	Organic Navigation
Movie Map	Browsing	Attribute Based Editor's choice	Keywords/freeform

To summarize, we have different combinations of clothing recommendation services. Most of them use separately features from pictures, text, expert assessors. In this work, we try to combine all features together and analyze the importance of features based on expert assessors.

1.3. Technology Requirement

For developing shopping recommendation system based on metric analysis of clothing descriptions, we use mathematical algorithms and technologies. In this chapter, we would like to introduce them.

Euclidian Distance Function

$$\mathrm{E}(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{q=1}^{n}\left(x_q - y_q\right)^2}, \tag{1}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are two input vectors, and n is the number of features in the application.

Heterogeneous Euclidean-Overlap Metric (HEOM)

One way to deal with applications with continuous and nominal attributes is to utilize a heterogeneous distance function that makes use of different attribute distance functions on different kinds of features. One approach uses the overlap metric for nominal attributes and normalized Euclidean distance for linear attributes [36]. The following function defines the distance between two values $\boldsymbol{x}$ and $\boldsymbol{y}$ of a given feature i as [36]:

$$\mathrm{d}_i\,(x,y) = \begin{cases} 1, & \text{if } x \text{ or } y \text{ is unknown,} \\ \text{overlap}\,(x,y), & \text{if } i \text{ is nominal,} \\ \mathrm{diff}_i(x,y). & \end{cases} \tag{2}$$

Unknown attribute values are handled by returning an attribute distance of 1 (i.e., a maximal distance) if either one of the attribute values is unknown. The function overlap and the range normalized difference *diff* are defined as [36]:

$$\text{overlap}\,(x,y) = \begin{cases} 0, & \text{if } x = y, \\ 1, & \text{otherwise,} \end{cases} \tag{3}$$

$$\mathrm{diff}_i(x,y) = \frac{|x-y|}{\max_i - \min_i}. \tag{4}$$

where max_i and min_i are the maximum and minimum values, respectively, observed in the training set for attribute i. This means that it is possible for a new input vector to have a value outside this range and produce a difference value greater than one. However, such cases are rare, and when they do occur, a large difference may be acceptable anyway. The normalization serves to scale the attribute down to the point where differences are usually less than one [36].

The above definition for d_i returns a value which is typically in the range 0..1, whether the attribute is nominal or linear. The overall distance between two possibly heterogeneous input vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by the HEOM $(\boldsymbol{x}, \boldsymbol{y})$ [36]:

$$\mathrm{HEOM}(x, y) = \sqrt{\sum_{i=1}^{n} d_i(x_i, y_i)^2}. \tag{5}$$

"This distance function removes the effects of the random ordering of nominal values, but its overly simplistic approach to handling nominal attributes fails to make use of additional information provided by nominal attribute values that can aid in generalization" [36].

TF.IDF

Frequency–Inverse Document Frequency (TF.IDF) is a mathematical statistic that is meant to show how important a word is to a document in a collection or corpus. Many researchers use it as a weighting factor in information retrieval and text mining. The TF.IDF value grows with the number of times a word appears in the document, but is offset by the frequency of the word in the corpus. It means that the frequency helps to deal with the fact that some words appear more than one times [37].

TF.IDF is the product of two statistics, term frequency and inverse document frequency.

TF stands for term frequency, the number of times that term *t* occurs in document *d*.

IDF stands for inverse document frequency, which means the amount of information the word provides, whether the term is common or rare across all documents. It is the logarithmically scaled fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient [38].

$$\mathrm{TF.IDF}(t, d, D) = \mathrm{TF}(t, d) \cdot \mathrm{IDF}(t, D). \tag{6}$$

$$\mathrm{TF}(t, d) = \frac{n_i}{\sum_k n_k}, \qquad \mathrm{IDF}(t, D) = \log \frac{|D|}{|(d_i \supset t_i)|}. \tag{7}$$

where

$|D|$ is a total number of documents in the corpus;

$|(d_i \supset t_i)|$ - number of documents where the term t_i appears (when $n_i \neq 0$)

Cosine similarity measure

Let's $\boldsymbol{x}$ and $\boldsymbol{y}$ – n-dimension vectors. Then cosine similarity measure:

$$\mathrm{CosSim}\,(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_i x_i \cdot y_i}{\sqrt{\sum_i x_i^2} \cdot \sqrt{\sum_i y_i^2}} = \frac{< \boldsymbol{x}, \boldsymbol{y} >}{||\boldsymbol{x}|| \cdot ||\boldsymbol{y}||}. \tag{8}$$

Gradient descent

"Gradient descent is an optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent" [39]:

$$f(x)\colon \mathbb{R}^n \to \mathbb{R}, \tag{9}$$

$$f(x) \to \min_{x \in \mathbb{R}^n} f(x). \tag{10}$$

If *f(x)* has gradient, then we can use gradient descent to define a minimum of the function (table 1.3.1).

Table 1.3.1 – Gradient descent algorithm

input: $f(x)\colon \mathbb{R}^n \to \mathbb{R}$

output: optimum *x*

1) repeat: $x^{[k+1]} = x^{[k]} - \lambda^{[k]} \cdot \nabla f\left(x^{[k]}\right)$, where $\lambda^{[k]}$:
   - const ($f(x)$ – differentiable, bounded above or strongly convex with const Λ),
   - decreases with fractional step (when *const* method does not work),
   - $\lambda^{[k]} = \operatorname*{argmin}_{\lambda} f\left(x^{[k]} - \lambda \cdot \nabla f\left(x^{[k]}\right)\right)$ (steepest descent method).
2) if the stopping criterion holds, then output = $x^{[k+1]}$

Stopping criterion:

1) $\left\|x^{[k+1]} - x^{[k]}\right\| \le \epsilon$,
2) $\left\|f(x^{[k+1]}) - f(x^{[k]})\right\| \le \epsilon$, ϵ – *const.*

Sequential quadratic programming

Sequential Quadratic Programming (SQP) is an effective method for the numerical solution of constrained nonlinear optimization problems. It is based on a deep theoretical foundation and provides influential algorithmic tools for the solution of large-scale technologically related problems. The SQP method is defined as an overview of Newton's method for unconstrained optimization. A number of software packages (NPSOL, NLPQL, OPSYC, OPTIMA, MATLAB, and SQP) are based on this approach" [40].

We consider the application of the SQP methodology to nonlinear optimization problems of the form:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{over } x \in \mathbb{R}^n \\ &\text{subject to } h(x) = 0 \\ &\qquad\qquad\quad g(x) \leq 0 \end{aligned}$$

where $f: \mathbb{R}^n \to \mathbb{R}$ is the objective functional, the functions $h: \mathbb{R}^n \to \mathbb{R}^m$ and $g: \mathbb{R}^n \to \mathbb{R}^p$ describe the likeness and difference constraints.

The nonlinear optimization problems consist of linear and quadratic programming problems, when f is linear or quadratic and the constraint functions h and g are affine.

Using these algorithms and technics we implement shopping recommendation system based on metric analysis of clothing descriptions.

2. Problem statement, the initial hypothesis, the input data

2.1. Goal setting and initial hypothesis

The goal of this research work is to develop shopping recommendation system based on metric analysis of clothing descriptions. The recommendation system finds clothes similar to the queries. Similarity is defined as a minimum distance between a query and responses. We also try to maximize the distance between relevant responses. The maximum distance between relevant responses allows users to find relevant clothes with maximum difference within a minimum distance to query.

We propose the hypothesis that clothing descriptions (features) have different importance (different weight). It is not necessary to take into account all features to recommend users similar clothes. Most probably people look at several features and, based on them, make their decisions to choose clothes.

In fig. 2.1.1, we show a block-diagram of the shopping recommendation system based on metric analysis of clothing descriptions.

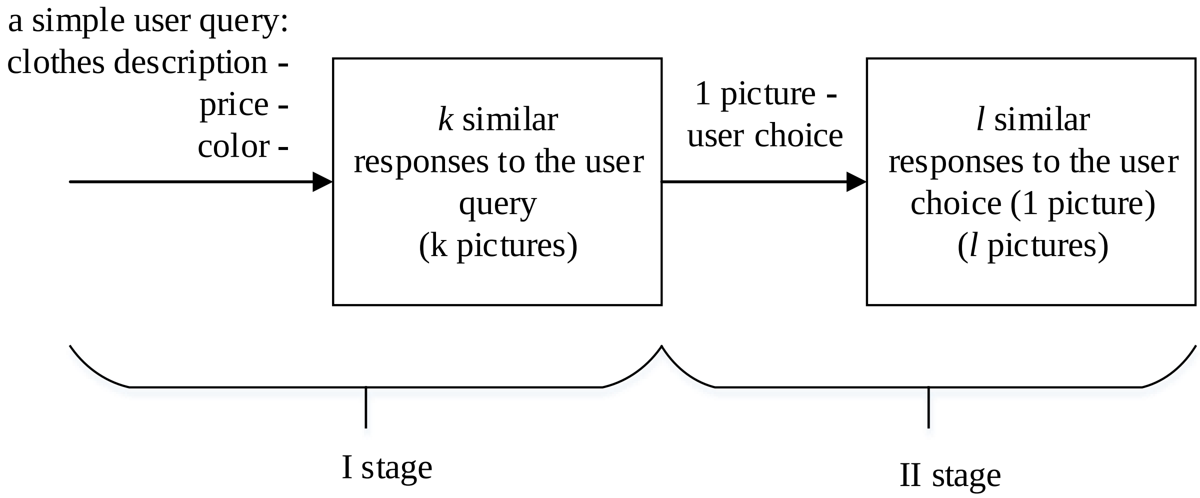


Figure 2.1.1 – The block-diagram of the shopping recommendation system based on metric analysis of clothing descriptions

I Stage

1. A user writes a garment description. For example: "I would like to buy a winter midi-dress with blue buttons from Russia."
2. The user chooses price and color from combo box. For example: "from 5690", "up to 12870" and "red" respectively.
3. The system finds *k* similar responses to the user query (the system extracts features from the text of the user query and compares the features: "Price" and "Color", with features in the Dataset).
4. The system shows *k* pictures to the user.

II Stage

Most often, Stage I does not provide sufficient information and it is necessary to clarify the query.

1. The user chooses one of the *k* pictures from I Stage.
2. The system finds *l*, similar to the picture, results (the system compares the document with the chosen picture with all documents in the dataset. Comparison goes on 35 features).
3. The system shows *l* final pictures to the user.

2.2. Data structure

In this section, we describe the structure of the dataset for the shopping recommendation system. The dataset items use 37 features, including 20 linear scales, 16 nominal scales and 1 text scale (table 2.2.1).

Scale is an algebraic structure with a given set of operations and relations that satisfies a fixed set of axioms. In statistics and quantitative research methodology, various attempts have been made to classify variables (or types of data) and thereby develop a taxonomy of levels of measurement or scales of measure [41].

Nominal scale is a finite list of symbols.

Linear scale is ordinal scale with operations: addition and subtraction.

Table 2.2.1 – Dataset feature description of the shopping recommendation system

#	Feature	Scale	Example
1	Type	nominal, $\mathbb{C}$	Платья
2	Subtype	nominal, $\mathbb{C}$	Вязаные платья
3	Price	linear, $\mathbb{W}$	[0…+∞]
4	Color	nominal, $\mathbb{C}$	бежевый
5	Description	text, $\mathbb{T}$	Платье Savage голубого цвета с контрастными рукавами. Модель выполнена из мягкого трикотажа. Детали: приталенный крой, круглый вырез.
6	Photo	picture, $\mathbb{P}$	SA004EWCJH70.jpg
7	Brand	nominal, $\mathbb{C}$	Finn Flare
8	Season	nominal, $\mathbb{C}$	Демисезон
9	Collection	nominal, $\mathbb{C}$	Весна-лето
10	Country	nominal, $\mathbb{C}$	Россия
11	The length of a back	linear, $\mathbb{W}$	[0…200]
12	Sleeve items	linear, $\mathbb{W}$	[0…200]
13	Clothing items	nominal, $\mathbb{C}$	прозрачность
14-30	17 materials	linear, $\mathbb{W}$	[0…100]
31	Article	nominal, $\mathbb{C}$	SA004EWCJH70
32	Evening dress	nominal, $\mathbb{C}$	{0,1}

33	Everyday dress	nominal, ℂ	{0,1}
34	Modest dress	nominal, ℂ	{0,1}
35	Catchy dress	nominal, ℂ	{0,1}
36	Adult dress	nominal, ℂ	{0,1}
37	Youth dress	nominal, ℂ	{0,1}

To get the items dataset we used several technics. First, we used the crawler to look through the Web pages of on-line merchants. Second, we extract all relevant information (such as separate parameters and their values, photos and short text description for every product). We get the raw database for further analysis.

Crawler. The program uses the Scrapy library for Python and allows to look through all of the relevant website links starting from the start url. When it opens the link with the dress description, it starts looking through the HTML-code, extracting information by rules written in XPath language (fig. 2.2.1). This information is automatically being saved in CSV-file as the database (the fragment is shown in fig. 2.2.2).

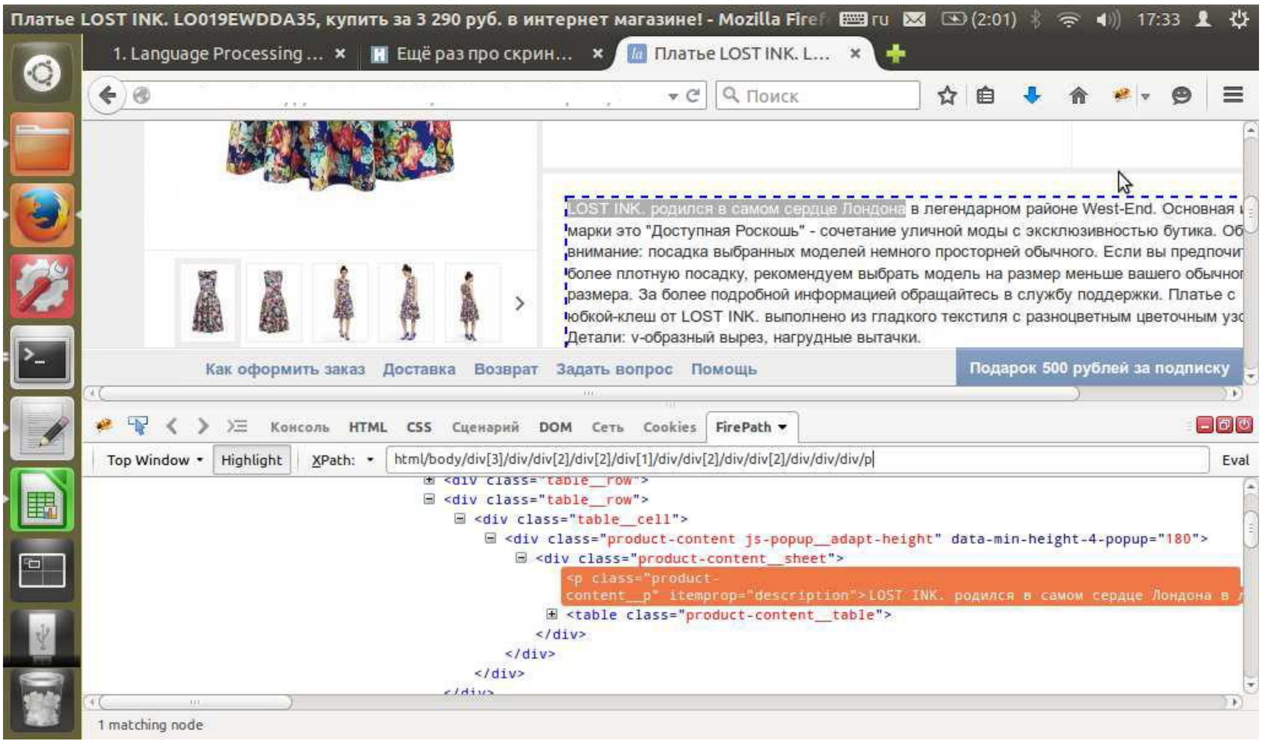


Figure 2.2.1 – HTML-code analysis for XPath usage

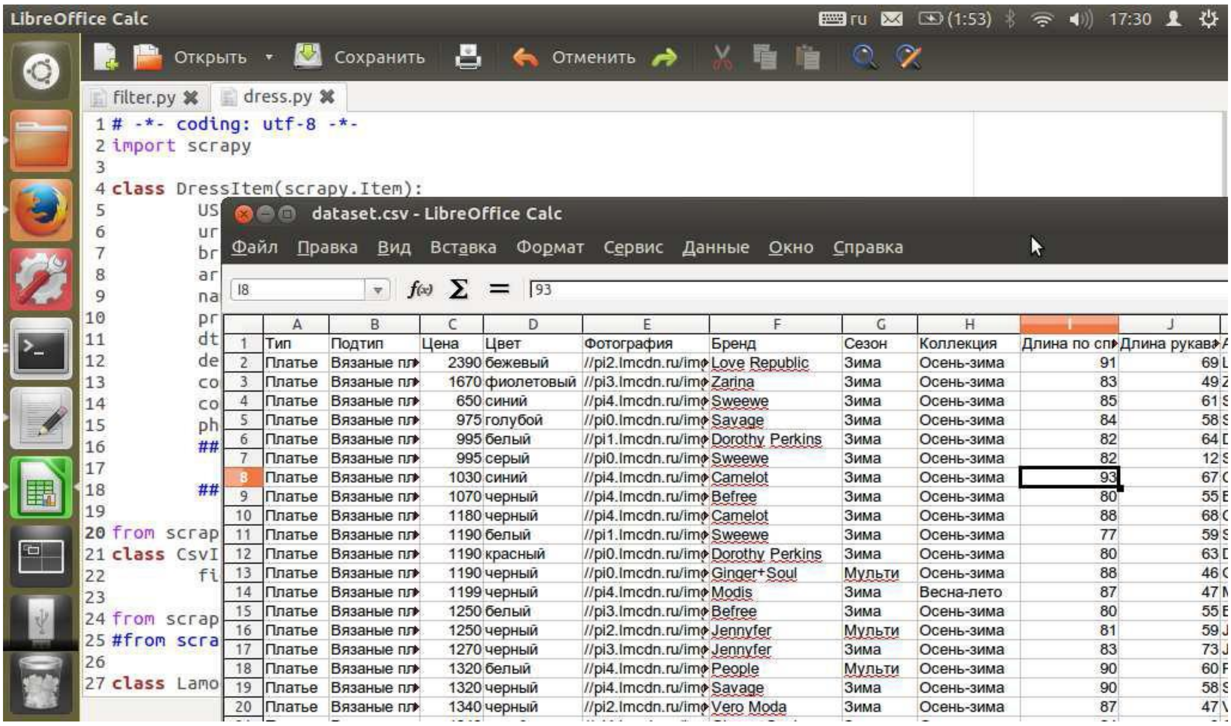


Figure 2.2.2 – The fragment of database extracted by the crawler

In order to make the database more convenient and ready to use we had to work on its structure and change some parameters (to divide the chart of parameters from website into separate ones with defined values).

Some of the extracted parameters can not be used before processing. For the second step we write additional code to process all data. It divides complex parameters into parameter-value pairs and extracts useful information from the text description (we use frequency analysis for this purpose).

After the clean-up changes, we are provided, we get a well-composed, ready to use database. The database consists of 4435 items; each item contains 31 features (table 2.2.1).

Then, we use expert annotations collected from 22 women and presented in nominal scales:

- Evening dress/Everyday dress,
- Modest dress/Catchy dress,
- Adult dress/ Youth dress.

Each woman looked at each item and marked “0” or “1” in the nominal scales, mentioned before. “0” means positive response, “1” – negative response.

Finally, the dataset consist of 37 features and 4435 items.

For database representation, we have used CSV (Comma Separated Values) format that is the most common import and export format for spreadsheets and databases (database.csv).

2.3. Problem statement

In this section, we present a formal definition of the problem statement of the shopping recommendation system based on metric analysis of clothing descriptions.

The initial dataset contains the set of pairs of mixed-scale data:

$$\mathfrak{D} = \{(\boldsymbol{x_i}, \boldsymbol{y_i}): i \in \mathcal{I}\}, \text{the object index } i \in \mathcal{I}\{1, \dots, m\},$$

$\boldsymbol{x}_i \in \mathbf{X}, \mathbf{X}$ – the object-feature matrix for the dataset, $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^T$, $\boldsymbol{y}_i \in \mathbb{X}^{\boldsymbol{k}}$ – vector objects relevant, where $\mathbb{X} = \mathbb{L}_{\mathbf{1}} \times \dots \times \mathbb{L}_{\boldsymbol{n}}$ – an object space.

Objects are in mixed scales with a metric *d:*

$$d: \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}_+, \tag{11}$$

$$d(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sqrt{\sum_{q=1}^{n} \alpha_q \cdot r(x_{iq}, x_{jq})^2}, \tag{12}$$

where *n* – a number of features, α_q – coefficients (weights) of the features q. The higher α_q, the more important feature q.

Statement 1

As d is a metric, it follows $d \geq 0$. As $r(x_{iq}, x_{jq})^2 \geq 0$, follows $\alpha_q \geq 0$. Let's normalize $\boldsymbol{a}: \sum_{q=1}^{n} a_q = 1$.

$$\alpha_q \geq 0, \tag{13}$$

$$\sum_{q=1}^{n} a_q = 1, \tag{14}$$

where $r(x_{iq}, x_{jq})$ is a distance between vectors $\boldsymbol{x_i}$, $\boldsymbol{x_j}$ on the feature q.

We have three types of scales: nominal, linear and text. For the defining $r(x_{iq}, x_{jq})$, we use Heterogeneous Euclidean-Overlap Metric (HEOM):

$$r(x_{iq}, x_{jq}) = \begin{cases} \text{diff}\,(x_{iq}, x_{jq}), if\ \mathbb{L}_q - \text{linear scale}, \\ \text{overlap}(x_{iq}, x_{jq}), if\ \mathbb{L}_q - \text{nominal scale}, \\ \text{sim}\,(x_{iq}, x_{jq}), \quad if\ \mathbb{L}_q - \text{text scale}, \end{cases} \tag{15}$$

$$\text{diff}(x_{iq}, x_{jq}) = \frac{|x_{iq} - x_{jq}|}{max_{\mathbb{L}_q} - min_{\mathbb{L}_q}}, \tag{16}$$

$$\text{overlap}(x_{iq}, x_{jq}) = \begin{cases} 1, \text{whenever } x_{iq} \neq x_{jq}, \\ 0, \qquad\qquad \text{otherwise}, \end{cases} \tag{17}$$

$$\text{sim}(x_{iq}, x_{jq}) = \arccos \frac{< x_{iq}, x_{jq} >}{||x_{iq}|| \cdot ||x_{jq}||}. \tag{18}$$

The function diff (x_{iq}, x_{jq}) is determined by normalized difference between two values of feature q.

The range of the resulting function d is less than or equal to the square root of the feature number.

$$d_i(x_i, y) \in [0, \sqrt{n}]. \quad (19)$$

We need to find relevant responses from a query with maximum distances between responses.

The problem statement is to define vector $\boldsymbol{\alpha}$ - vector of weighs, with following conditions:

1) to minimize sum of distances between a query and relevant responses;
2) to maximize sum of distances between a query and irrelevant responses;
3) to maximize sum of distances between relevant responses.

$$\boldsymbol{\alpha} = (a_1, \dots, a_n) = \begin{cases} \operatorname{argmin} \sum_{i=1}^{m} \sum_{y \in \mathbb{Y}_r} d(x_i, y), \\ \operatorname{argmax} \sum_{i=1}^{m} \sum_{y \in \mathbb{Y}_{nr}} d(x_i, y), \\ \operatorname{argmax} \sum_{i=1}^{m} \sum_{y', y'' \in \mathbb{Y}_r} d(y', y''), \end{cases} \quad (20)$$

where m – a number of objects in the dataset, $\mathbb{Y}_r$ is a space of relevant responses, $\mathbb{Y}_{nr}$ is a space of irrelevant responses; y, $y^{'}$, $y^{''}$ are responses.

Let

$$A = \frac{\sum_{i=1}^{m} \frac{\sum_{y \in \mathbb{Y}_r} d_i(x_i, y)}{\sqrt{n} \cdot |\mathbb{Y}_r|}}{m}, \quad (21)$$

$$B = \frac{\sum_{i=1}^{m} \frac{\sum_{y \in \mathbb{Y}_{nr}} d_i(x_i, y)}{\sqrt{n} \cdot |\mathbb{Y}_{nr}|}}{m}, \quad (22)$$

$$C = \frac{\sum_{i=1}^{m} \frac{2 \cdot \sum_{y', y'' \in \mathbb{Y}_r} d_i(y', y'')}{\sqrt{n} \cdot |\mathbb{Y}_r \cdot (|\mathbb{Y}_r| - 1)|}}{m}. \quad (23)$$

The values of $A, B, C \in [0,1]$, therefore, we can rewrite:

$$\boldsymbol{\alpha} = (a_1, \dots, a_n) = \begin{cases} \operatorname{argmin} A(\boldsymbol{\alpha}), \\ \operatorname{argmax} B(\boldsymbol{\alpha}), \\ \operatorname{argmax} C(\boldsymbol{\alpha}). \end{cases} \quad (24)$$

Suppose that $\lambda = \text{const}$,

$$B(\boldsymbol{\alpha}) \cdot \big(1 - A(\boldsymbol{\alpha})\big) + \lambda \cdot C(\boldsymbol{\alpha}) \to \max. \quad (25)$$

Thereby, the task is to solve optimization problem, to find a maximum of the function *f(**α**):*

$$f(\boldsymbol{\alpha}) = B(\boldsymbol{\alpha}) \cdot \big(1 - A(\boldsymbol{\alpha})\big) + \lambda \cdot C(\boldsymbol{\alpha}) \to \max. \quad (26)$$

2.4. Data preprocessing

The next step is to prepare data and a filter that can sort the positions in the database according to the user's request and choose every position that satisfies the user's demand.

Stopwords

We have used Wikipedia [42] to collect function words. We have interjections, particles, prepositions, pronouns, question words and unions (table 2.4.1). Every type of function words saved as txt-file.

Table 2.4.1 – Stopwords

Function words	Number
Interjections	302
Particles	151
Prepositions	189
Pronouns	38
Question words	16
Unions	141

A dictionary and a corpus

As was mentioned in table 2.2.1, feature #5 typically consists of several sentences. For instance: "Платье Savage голубого цвета с контрастными рукавами. Модель выполнена из мягкого трикотажа. Детали: приталенный крой, круглый вырез."

It is useful to compare descriptions from database with the queries. For this reason, we create a dictionary. We offer the following solution for this case. The *gensim* library for Python is a tool to realize unsupervised semantic modelling from plain text. It allows us to build a frequency distribution of words for any plain text.

There are 4435 documents (items) in the dataset. Therefore, we have a corpus of 4435 documents. First, tokenize the documents, remove punctuation, spaces, stemming each word in the document. For the stemming process, use Python library – snowballstemmer.

Next, remove common words, using the stoplist from stopwords, mentioned before, as well as words that appear once in the corpus. To convert documents to vectors, use a document representation called bag-of-words. In this representation, each document is represented by one vector where each vector element represents a word-repetition number of word in the document pair. The mapping between a word and its unique ids is called a dictionary. Here we assigned a unique integer id to all words appearing in the corpus from the gensim library. This sweeps across the texts, collecting word counts and relevant statistics.

In the end, we get 1424 distinct words in the processed corpus, which means, 1424 numbers (1424-D vector) represent each document.

There is a dictionary: {' выполн': 6, ' удлинен': 297, ' paris': 927, ' ожерел': 705, ' пол': 746, ' с-образн': 227, 'мероприят': 854, …}.

There is a corpus for the first document: [(0, 1.0), (1, 1.0), (2, 1.0), (3, 1.0), (4, 1.0), (5, 2.0), (6, 1.0), (7, 2.0), (8, 1.0), (9, 1.0), (10, 1.0), (11, 1.0), (12, 1.0), (13, 2.0), (14, 1.0), (15, 3.0), (16, 1.0), (17, 1.0), (18, 1.0), (19, 1.0), (20, 1.0), (21, 1.0), (22, 1.0), (23, 1.0)]

Summarize, we have the dictionary – dictionary.dict, and the corpus for each document – dictionary.mm.

3. The implementation of the algorithm of the shopping recommendation system

3.1. Expert evaluation

In the thesis, we have twenty two experts (women) to evaluate six nominal features: Evening dress, Everyday dress, Modest dress, Catchy dress, Adult dress, Youth dress. Twenty one expert evaluate 200 documents and one expert evaluate 235 documents. Total evaluation: 4435 documents.

Information about experts:

- women;
- 20-27 years old;
- students of engineering science and humanities;
- Russian.

The task is to read 31 features of each documents and evaluate six more ones based on information from 31 features.

In the table 3.1.1, we show the results of the evaluations.

Table 3.1.1 – Results of the evaluations

	Evening dress	Everyday dress	Modest dress	Catchy dress	Adult dress	Youth dress
##	1458	2977	2467	1968	1802	2633
sum	4435		4435		4435	

There are 1458 evening and 2977 everyday dresses, 2467 modest and 1968 catchy dresses, 1802 adult and 2633 youth dresses. The total number of dresses is 4435.

3.2. Heterogeneous Euclidean-Overlap Metric

In this section, we calculate the distances between request and responses in mixed scales, using HEOM. There are three different types of scales in the dataset: text, nominal, linear. Let start with Stage II (Figure 2.1.1). The request in Stage II has 35 features: 1 – in a text scale, 20 – in linear scales and 14 – in nominal scales.

Using HEOM metric, we calculate a distance between a query and all documents in the dataset. Then, using the ranking, we sort the results. The responses with minimum distances is the most similar with the query in Stage II.

As was mentioned in (12), a metric *d* for our case

$$d(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sqrt{\sum_{q=1}^{35} \alpha_q \cdot r(x_{iq}, x_{jq})^2}. \tag{27}$$

Let $\alpha_q = 1$ for any $q \in [1,35]$.

As you can see, we did not use two features: “Photo” and “Article” for the HEOM metric.

Text

First, tokenize a request – text column, remove punctuation, spaces, stemming each word in the document. Use library – snowballstemmer in Python. Next, we remove common words, using the stopwords. Finally, use a module – similarities – from gensim library to find cos distance between the request and each document in the dataset. Use an equation (18). For each request, get a vector. A vector have a dimension: 4435x1. Each value is from 0 to 1. “0” means that the request and a document in the dataset are very similar in terms of the text, “1” – very different.

Linear scales

For 20 linear features of a request, calculate a linear distances between corresponding features of the request and each document of the dataset. Use an equation (16).

In a result, have a vector of distances from the request to each of the document in the dataset. A vector has dimension: 4435x20. Each value is from 0 to 1. “0” means that the request and a document in the dataset are very similar in terms of each linear feature, “1” – very different.

Nominal scales

For 14 nominal features of a request, calculate a nominal distances between corresponding features of the request and each document of the dataset. Use the equation (17).

In a result, have a vector of distances from the request to each of the document in the dataset. A vector has dimension: 4435x14. Each value is 0 or 1. “0” means that the request and a document in the dataset are very similar in terms of each nominal feature, “1” – very different.

Using (27), calculate a metric *d*.

3.3. Parameters of optimization models

This section describes models of optimization models and finds parameters.

We can calculate a metric d. I would like to remind you that we assumed that $\alpha_q = 1$ for any $q \in [1,35]$.

In accordance with the Problem statement in 2.3, we need to solve minimax problem and define vector $\boldsymbol{\alpha}$ as mentioned in (20) and (26).

According to (26), we have to find a maximum of the function $f(\boldsymbol{\alpha})$.

It is an optimization problem. For solving the problem, we use several optimization algorithms: a gradient descent algorithm and a sequential quadratic programming method (SQP) for nonlinear optimization.

For a gradient descent algorithm, we need to find partial derivatives for each α_q.

First, we find a function $f(\boldsymbol{\alpha})$. We have:

$$f(\boldsymbol{\alpha}) = B(\boldsymbol{\alpha}) \cdot \big(1 - A(\boldsymbol{\alpha})\big) + \lambda \cdot C(\boldsymbol{\alpha}) \tag{28}$$

$$f'(\boldsymbol{\alpha}) = B'(\boldsymbol{\alpha}) \cdot \big(1 - A(\boldsymbol{\alpha})\big) - B(\boldsymbol{\alpha}) \cdot A(\boldsymbol{\alpha})' + \lambda \cdot C'(\boldsymbol{\alpha}) \tag{29}$$

Using (21), (22), (23), (28) and suppose, that λ =1:

$$f(\boldsymbol{\alpha}) = B(\boldsymbol{\alpha}) \cdot \big(1 - A(\boldsymbol{\alpha})\big) + C(\boldsymbol{\alpha}) \tag{30}$$

$$f'(\boldsymbol{\alpha}) = B'(\boldsymbol{\alpha}) \cdot \big(1 - A(\boldsymbol{\alpha})\big) - B(\boldsymbol{\alpha}) \cdot A(\boldsymbol{\alpha})' + C'(\boldsymbol{\alpha}) \tag{31}$$

$$\begin{aligned} f(\boldsymbol{\alpha}) = \frac{1}{m} \cdot \sum_{i=1}^{m} \Bigg[\frac{1}{\sqrt{n} \cdot |\mathbb{Y}_{nr}|} \cdot \Bigg(\sum_{y \in \mathbb{Y}_{nr}} \sqrt{\sum_{q=1}^{n} \alpha_q \cdot r_q\big(x_{iq}, y_q\big)^2} \Bigg) \cdot \\ \cdot \Bigg(1 - \sum_{i=1}^{m} \frac{1}{m \cdot \sqrt{n} \cdot |\mathbb{Y}_r|} \cdot \sum_{y \in \mathbb{Y}_r} \sqrt{\sum_{q=1}^{n} \alpha_q \cdot r_q\big(x_{iq}, y_q\big)^2} \Bigg) + \lambda \cdot \\ \cdot \frac{2}{\sqrt{n} \cdot \big(|\mathbb{Y}_r| \cdot (|\mathbb{Y}_r| - 1)\big)} \cdot \sum_{y', y'' \in \mathbb{Y}_r} \sqrt{\sum_{q=1}^{n} \alpha_q \cdot r_q(y', y'')^2} \Bigg] \end{aligned} \tag{32}$$

$$\frac{\delta f}{\delta \alpha_q} = \frac{1}{m}\cdot\sum_{i=1}^{m}\left[\frac{1}{\sqrt{n}\cdot|\mathbb{Y}_{nr}|}\cdot\left(\sum_{y\in\mathbb{Y}_{nr}}\frac{r_q\left(x_{iq},y_q\right)^2}{2\cdot\sqrt{\sum_{j=1}^{n}\alpha_j\cdot r_j\left(x_{ij},y_j\right)^2}}\right)\cdot\right.$$

$$\cdot\left(1-\sum_{i=1}^{m}\frac{1}{m\cdot\sqrt{n}\cdot|\mathbb{Y}_r|}\cdot\sum_{y\in\mathbb{Y}_r}\sqrt{\sum_{j=1}^{n}\alpha_j\cdot r_j\left(x_{ij},y_j\right)^2}\right)-\frac{1}{\sqrt{n}\cdot|\mathbb{Y}_{nr}|}\cdot$$

$$\cdot\left(\sum_{y\in\mathbb{Y}_{nr}}\sqrt{\sum_{j=1}^{n}\alpha_j\cdot r_j\left(x_{ij},y_j\right)^2}\right)\cdot \tag{33}$$

$$\cdot\sum_{i=1}^{m}\frac{1}{m\cdot\sqrt{n}\cdot|\mathbb{Y}_r|}\cdot\sum_{y\in\mathbb{Y}_r}\frac{r_q\left(x_{iq},y_q\right)^2}{2\cdot\sqrt{\sum_{j=1}^{n}\alpha_j\cdot r_j\left(x_{ij},y_j\right)^2}}+\lambda\cdot$$

$$\left.\cdot\frac{2}{n\cdot\left(|\mathbb{Y}_r|\cdot(|\mathbb{Y}_r|-1)\right)}\cdot\sum_{y',y''\in\mathbb{Y}_r}\frac{r_q(y',y'')^2}{2\cdot\sqrt{\sum_{j=1}^{n}\alpha_j\cdot r_j(y',y'')^2}}\right]$$

for any $q \in [1,35]$,

where *m* is a number of objects in the dataset, *n* is number of features, $\mathbb{Y}_r$ is a space of relevant responses, $\mathbb{Y}_{nr}$ is a space of irrelevant responses; *y*, y', y'' are responses.

For defining relevant and irrelevant responses, used an expert assessment. An expert assessed relevant and irrelevant responses, when $\alpha_q = 1$ for any $q \in [1,35]$. There are valuations of 200 queries with 20 relevant and irrelevant responses. Therefore, there are 200x20=4000 – power sampling. Split the queries into two different groups: 100 queries – for a learning set and 100 queries – for testing and evaluating. Split the queries randomly and repeat it 10 times.

To find the maximum, we take into account (13) and (14):

$$\alpha_q \geq 0,\ \sum_{q=1}^{n} a_q = 1$$

In the result, get $\max(f(\boldsymbol{\alpha}))$ for 10 learning sets (table 3.3.1, 3.3.2, figure 3.3.1).

Table 3.3.1 – Maximum values of $f(\boldsymbol{\alpha})$ for 10 learning sets (Gradient descent)

##	$f(\boldsymbol{\alpha})$	##	$f(\boldsymbol{\alpha})$	##	$f(\boldsymbol{\alpha})$	##	$f(\boldsymbol{\alpha})$	##	$f(\boldsymbol{\alpha})$
1.	0.2762	3.	0.2754	5.	0.2908	7.	0.2676	9.	0.2809
2.	0.2684	4.	0.2744	6.	0.2768	8.	0.2948	10.	0.2948

Table 3.3.1 – Maximum values of $f(\boldsymbol{\alpha})$ for 10 learning sets (SQP)

##	$f(\boldsymbol{\alpha})$	##	$f(\boldsymbol{\alpha})$	##	$f(\boldsymbol{\alpha})$	##	$f(\boldsymbol{\alpha})$	##	$f(\boldsymbol{\alpha})$
1.	0.2815	3.	0.2850	5.	0.2814	7.	0.2849	9.	0.2856
2.	0.2814	4.	0.2826	6.	0.2835	8.	0.2886	10.	0.2826

Maximum value of $f(\boldsymbol{\alpha})$:

Optimization, using Gradient descent method: $\max f(\boldsymbol{\alpha}) = \overline{\max f(\boldsymbol{\alpha})} \pm \delta = 0.2800 \pm 0.0101$

Optimization, using PQS method: $\max f(\boldsymbol{\alpha}) = \overline{\max f(\boldsymbol{\alpha})} \pm \delta = 0.2837 \pm 0.0023$

Mean value of two optimization methods: $\max f(\boldsymbol{\alpha}) = \overline{\max f(\boldsymbol{\alpha})} \pm \delta = 0.2819 \pm 0.0074$

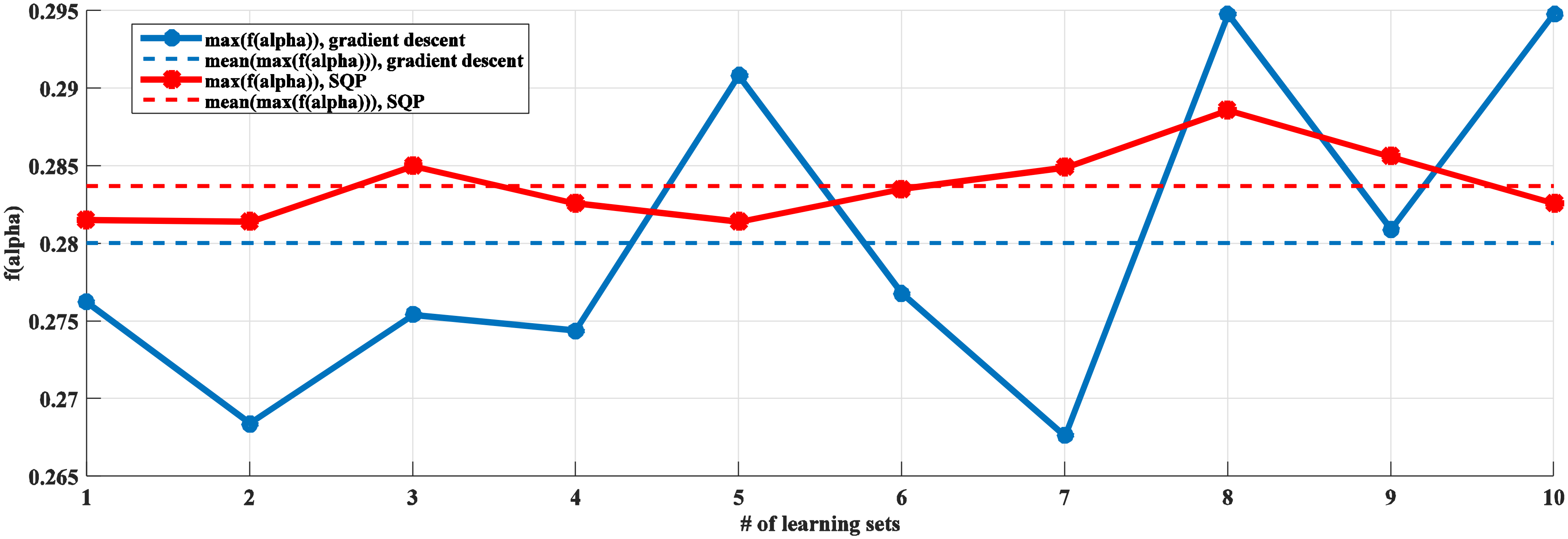


Figure 3.3.1 – Maximum values of $f(\boldsymbol{\alpha})$ with Gradient descent and SQP optimization methods

As you can see, SQP optimization method is more stable than gradient descent method for different learning sets.

3.4. The importance of features in mixed scales

We solved the optimization problem and found the $\boldsymbol{\alpha}$. We can draw graphs of a coefficient a_q is a value. The higher the value of the coefficient a_q is the more important the appropriate feature.

In table 3.4.1, you can see order features for the graphs.

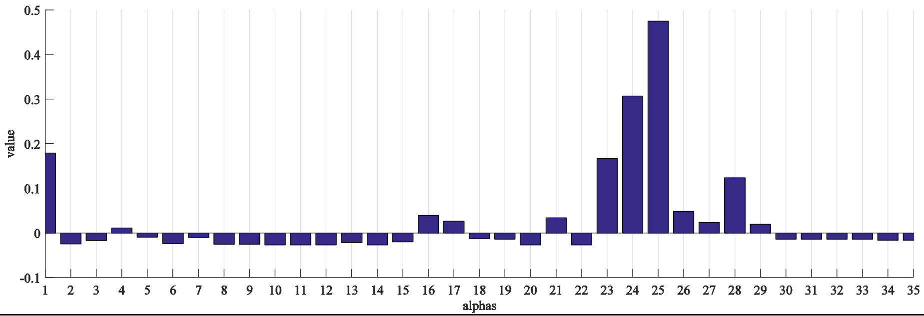
Table 3.4.1 – Features of the shopping recommendation system

#	Feature	Scale	Example
1	Description	text, $\mathbb{T}$	Платье Savage голубого цвета с контрастными рукавами. Модель выполнена из мягкого трикотажа. Детали: приталенный крой, круглый вырез.
2	Price	linear, $\mathbb{W}$	[0…+∞]
3	The length of a back	linear, $\mathbb{W}$	[0…200]
4	Sleeve items	linear, $\mathbb{W}$	[0…200]
5-21	17 materials	linear, $\mathbb{W}$	[0…100]
22	Type	nominal, $\mathbb{C}$	Платья
23	Subtype	nominal, $\mathbb{C}$	Вязаные платья
24	Color	nominal, $\mathbb{C}$	бежевый
25	Brand	nominal, $\mathbb{C}$	Finn Flare
26	Season	nominal, $\mathbb{C}$	Демисезон
27	Collection	nominal, $\mathbb{C}$	Весна-лето
28	Country	nominal, $\mathbb{C}$	Россия
29	Clothing items	nominal, $\mathbb{C}$	прозрачность
30	Evening dress	nominal, $\mathbb{C}$	{0,1}
31	Everyday dress	nominal, $\mathbb{C}$	{0,1}
32	Modest dress	nominal, $\mathbb{C}$	{0,1}
33	Catchy dress	nominal, $\mathbb{C}$	{0,1}
34	Adult dress	nominal, $\mathbb{C}$	{0,1}
35	Youth dress	nominal, $\mathbb{C}$	{0,1}

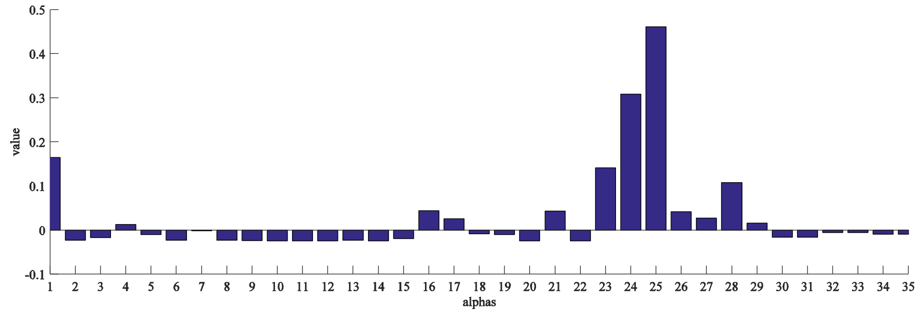
In fig. 3.4.1 – 3.4.4 you can see results of different optimization methods. On fig. 3.4.1 we used a gradient descent optimization method for ten random learning sets (1 set is 100 queries). On fig. 3.4.2 we used an Sequential quadratic programming method for the ten random learning sets (1 set is 100 queries). Fig. 3.4.3 and 3.4.4 shows standard deviations of the ten learning sets of the two optimization methods.

1) Gradient descent

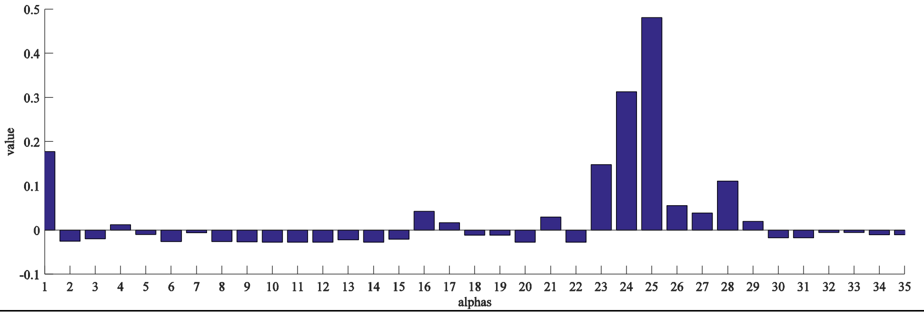
Learning test 1



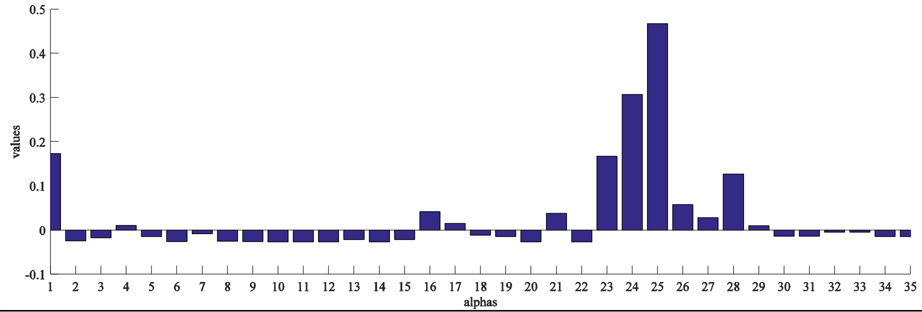
Learning test 2



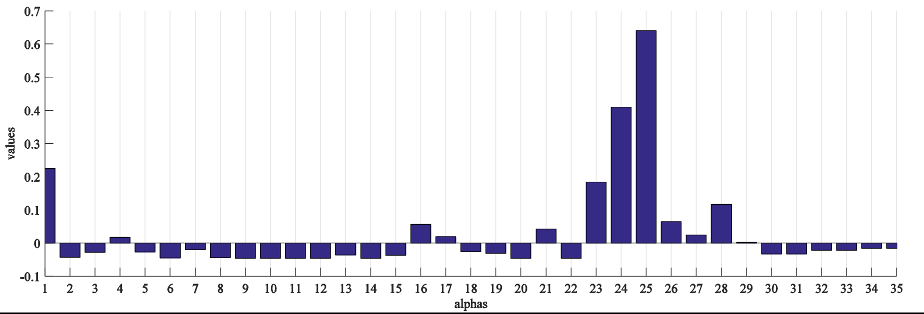
Learning test 3



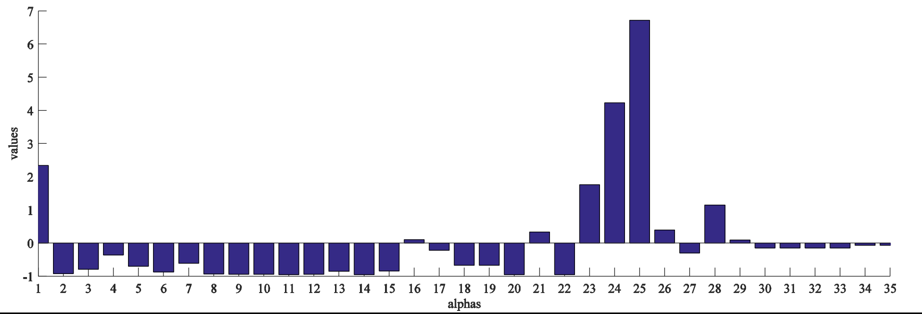
Learning test 4



Learning test 5



Learning test 6



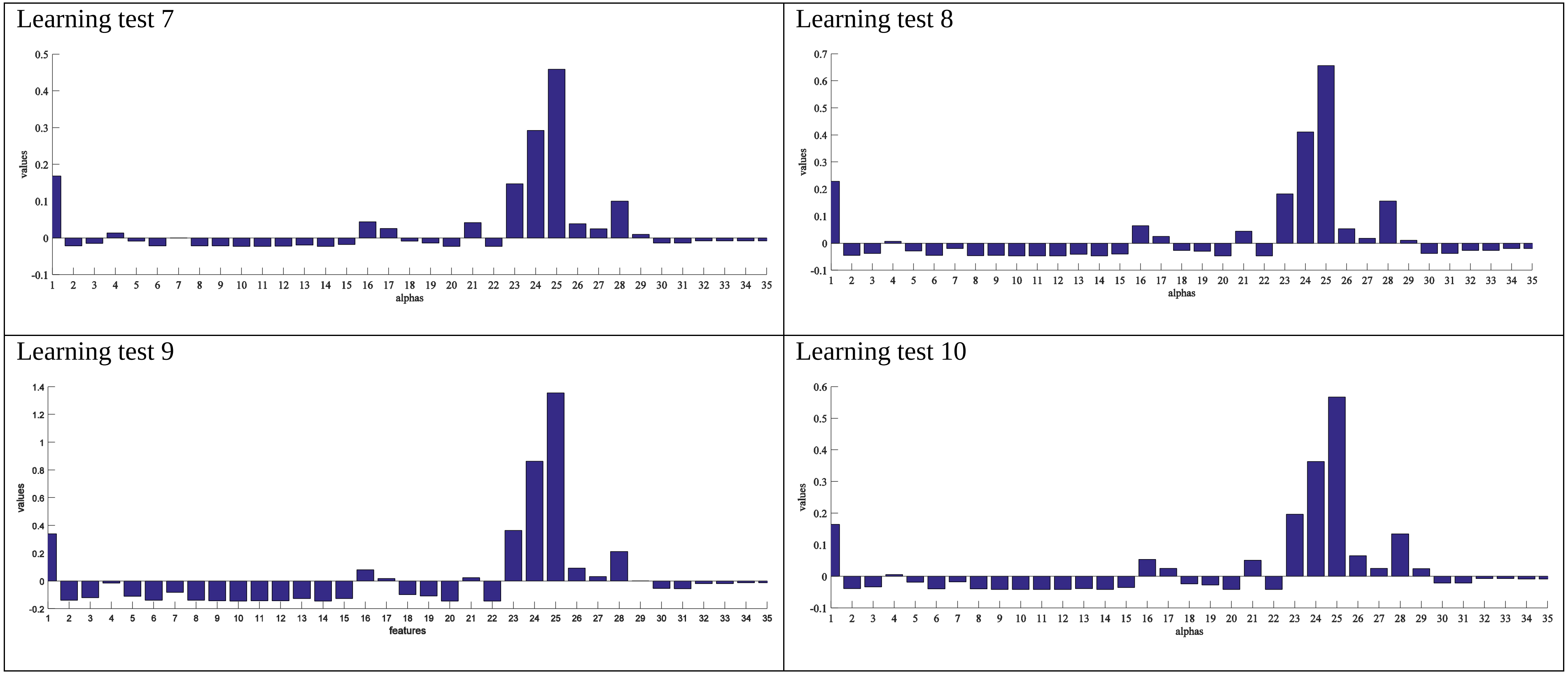
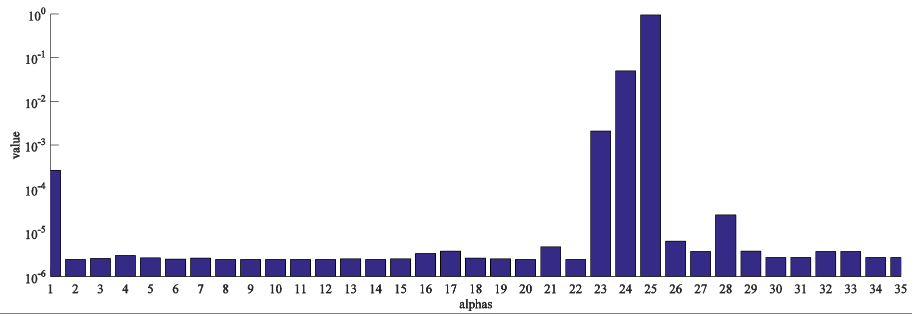


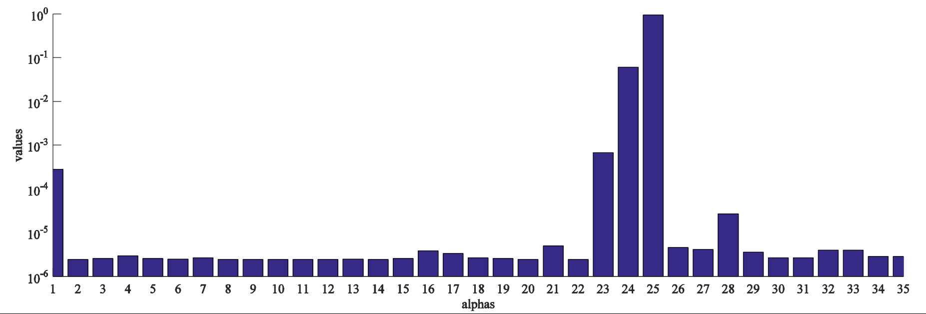
Figure 3.4.1 – The importance of features in mixed scales (Gradient descent)

2) Sequential quadratic programming

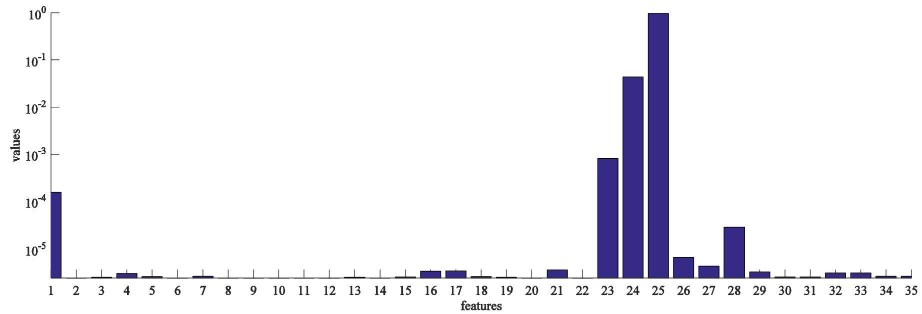
Learning test 1



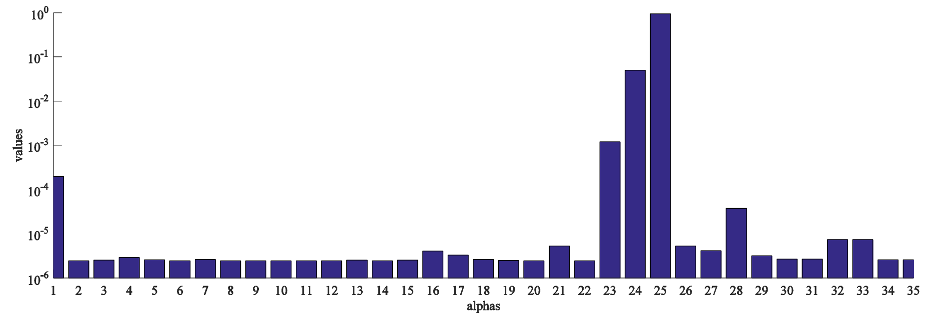
Learning test 2



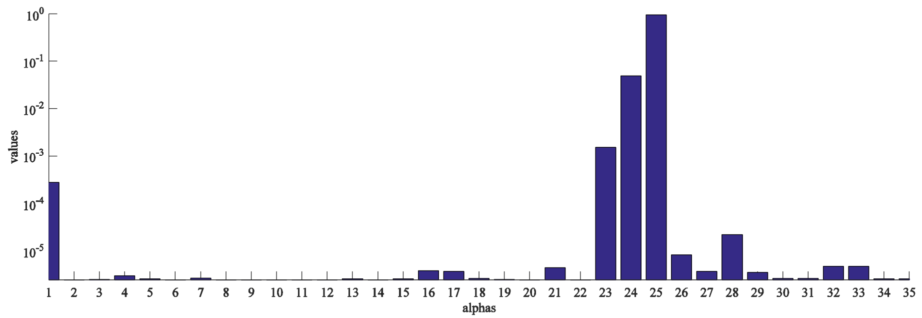
Learning test 3



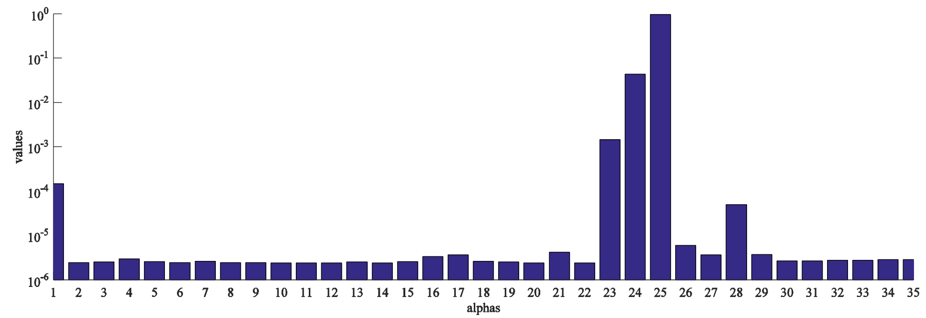
Learning test 4



Learning test 5



Learning test 6



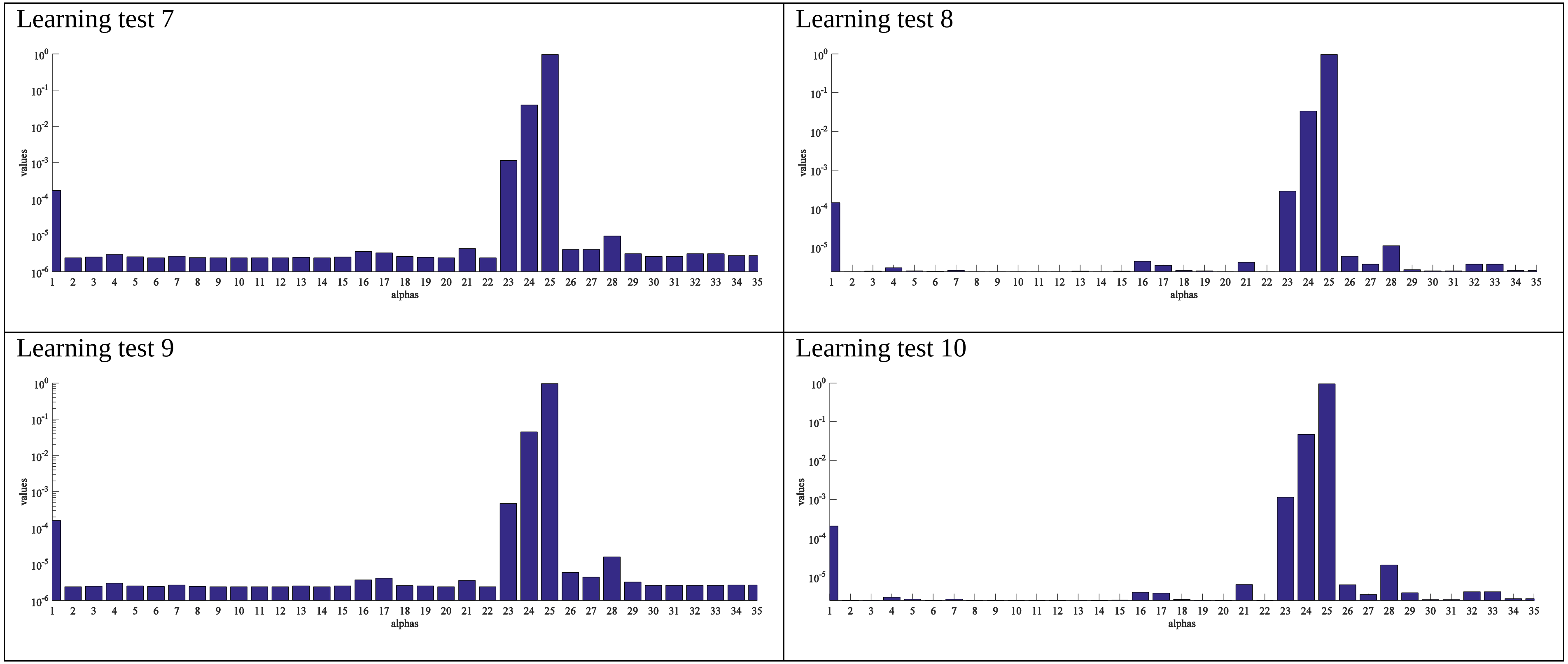


Figure 3.4.2 – The importance of features in mixed scales (Sequential quadratic programming)

Standard deviation of 10 learning sets

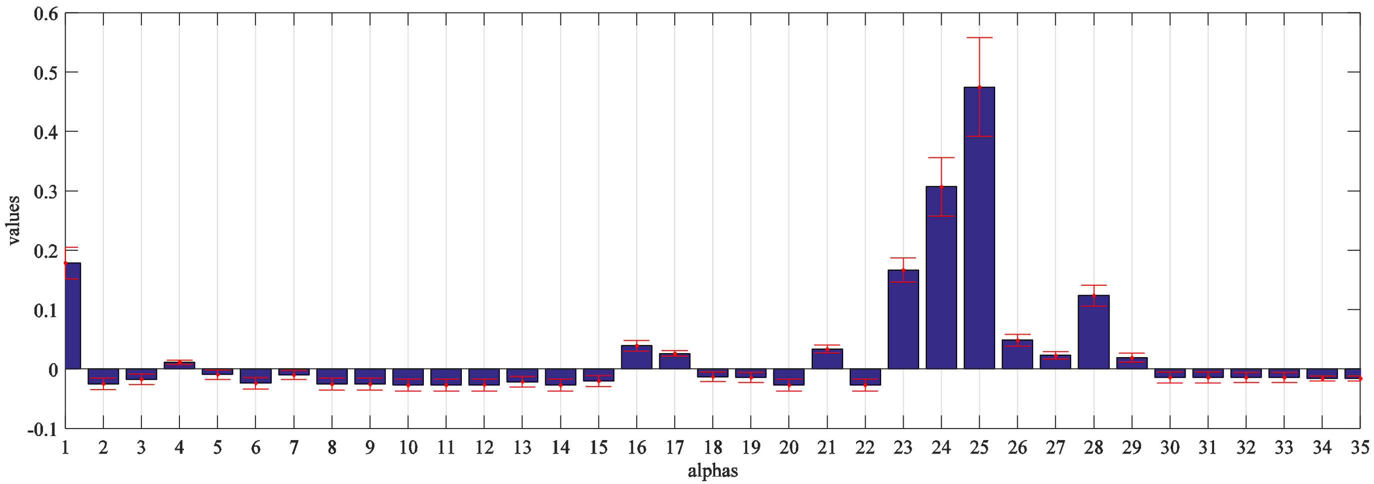


Figure 3.4.3 – The importance of features in mixed scales (Gradient descent – standard deviation)

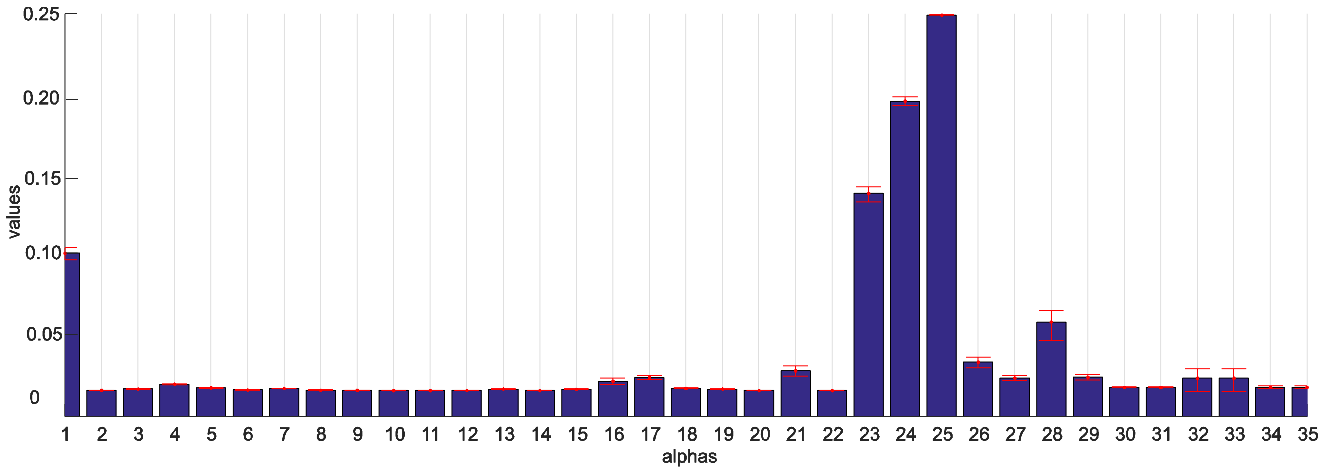


Figure 3.4.4 – The importance of features in mixed scales (Sequential quadratic programming – standard deviation)

Analyze Figure 3.4.1 – 3.4.4, we have numbers of importance features (table 3.4.2): 1, 4, 16, 17, 21, 23, 24, 25, 26, 27, 28, 32, 33

Table 3.4.2 – The important features

#	Feature	Scale	Example
1	Description	text, $\mathbb{T}$	Платье Savage голубого цвета с контрастными рукавами. Модель выполнена из мягкого трикотажа. Детали: приталенный крой, круглый вырез.
4	Sleeve items	linear, $\mathbb{W}$	[0…200]
16	Acrylic	linear, $\mathbb{W}$	[0…100]%
17	Cotton	linear, $\mathbb{W}$	[0…100]%
21	Polyester	linear, $\mathbb{W}$	[0…100]%
23	Subtype	nominal, $\mathbb{C}$	Вязаные платья
24	Color	nominal, $\mathbb{C}$	бежевый
25	Brand	nominal, $\mathbb{C}$	Finn Flare
26	Season	nominal, $\mathbb{C}$	Демисезон
27	Collection	nominal, $\mathbb{C}$	Весна-лето
28	Country	nominal, $\mathbb{C}$	Россия
32	Modest dress	nominal, $\mathbb{C}$	{0,1}
33	Catchy dress	nominal, $\mathbb{C}$	{0,1}

It was predictable that text description was the important feature. However, it is interesting that users look at acrylic, cotton and polyester materials. Color and brand, season and collection, country and style are important to people. You can see that price, other materials and length of a sleeve are not important factors for user request.

The most important features (table 3.4.3):

1, 16, 17, 23, 24, 25

Table 3.4.3 – The most important features

#	Feature	Scale	Example
1	Description	text, $\mathbb{T}$	Платье Savage голубого цвета с контрастными рукавами. Модель выполнена из мягкого трикотажа. Детали: приталенный крой, круглый вырез.
16	Acrylic	linear, $\mathbb{W}$	[0…100]%
17	Cotton	linear, $\mathbb{W}$	[0…100]%
23	Subtype	nominal, $\mathbb{C}$	Вязаные платья
24	Color	nominal, $\mathbb{C}$	бежевый
25	Brand	nominal, $\mathbb{C}$	Finn Flare

In the result, the most important features for users are description of the clothes, proportions of acrylic, cotton; subtype of the clothes, color and brand. It means that users do not need look at all features, they can decide based on several features.

4. Block diagram of the shopping recommendation system

In this part, we develop a functional diagram of the shopping recommendation system based on metric analysis of clothing descriptions.

4.1. The block-diagram of the shopping recommendation system

For the development of the shopping recommendation system, it is necessary to develop a block-diagram. The block diagram gives you understanding of the algorithm and structure of the whole system.

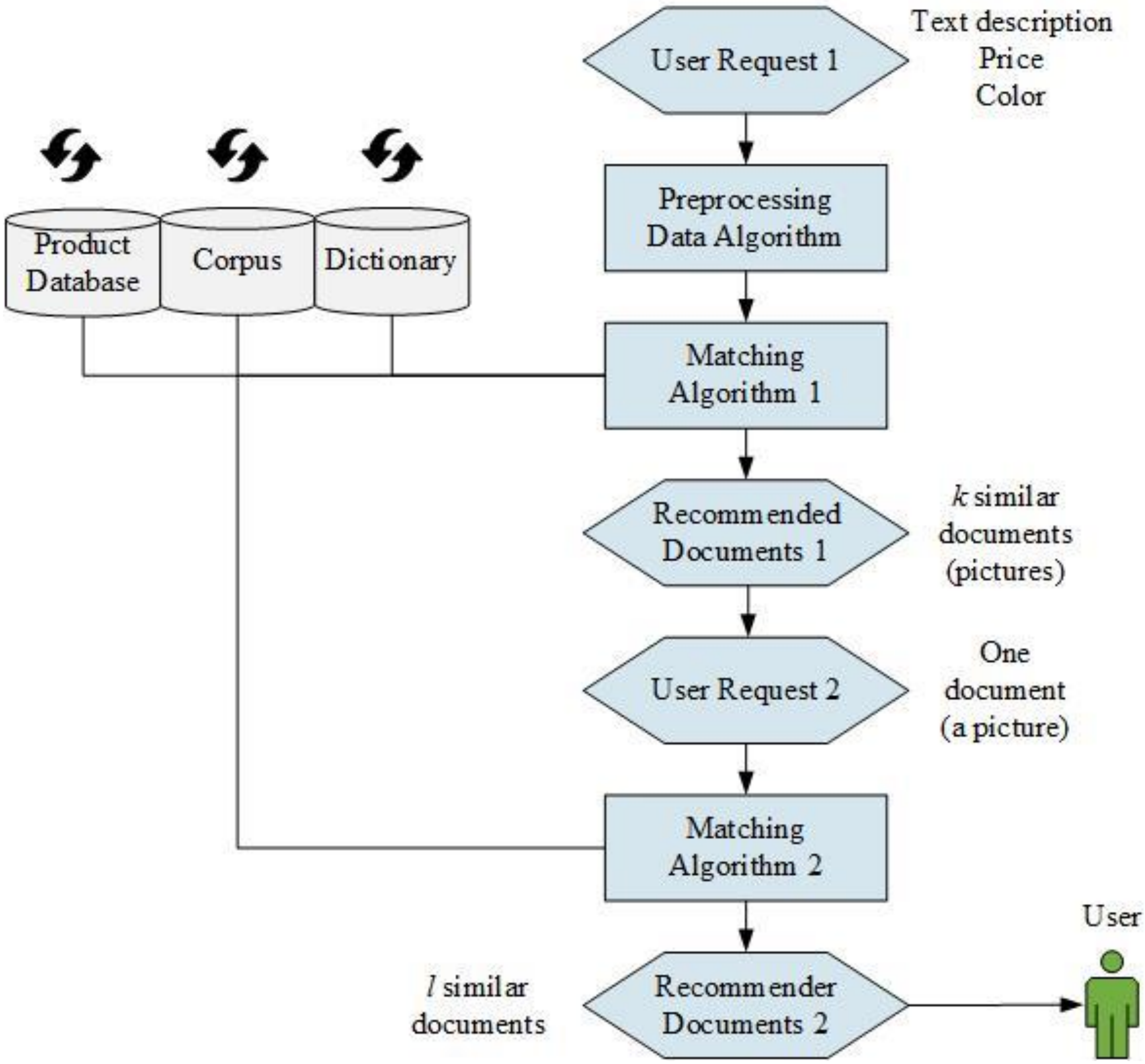


Figure 4.1.1 – The block-diagram of the shopping recommendation system

The block-diagram consists of seven main blocks and three auxiliary blocks (table 4.1.1).

Table 4.1.1 – Description of the block-diagram

User Request 1	A user writes a request: text description, choose price and color
Preprocessing Data Algorithm	The algorithm processes a query of a user
Matching Algorithm 1	The algorithm compares the query and documents in a dataset (3 features)
Recommended Documents 1	Results – *k* ranking similar documents
User Requests 2	The user chooses one similar document
Matching Algorithm 2	The algorithm compares the query (the one document) and documents in a dataset (35 features)
Recommended Documents 2	Results – *l* ranking similar documents
Product Database	dataset of documents
Corpus	corpus of dataset
Dictionary	dictionary of the system

Firstly, a user writes a request. The user writes text description of the desired clothing, chooses price and color. Next, the algorithm processes the user query and compares it with documents in a dataset. The comparison goes for three features. In the result, the system recommends the user *k* ranking similar documents – *k* similar clothes. Due to incomplete data (3 features vs. 35 features), we need to refine ranking results. The user chooses one similar clothes from the results. The algorithm compares the new query (35 features) and documents (35 features) in the dataset. Finally, the system recommends the user *l* ranking similar documents – *l* similar clothes.

4.2. The diagram IDEF0 of the shopping recommendation system

In this part, it is developed the diagram of the shopping recommendation system in IDEF0 notation. You can see the first level of the decomposition (figure 4.2.1).

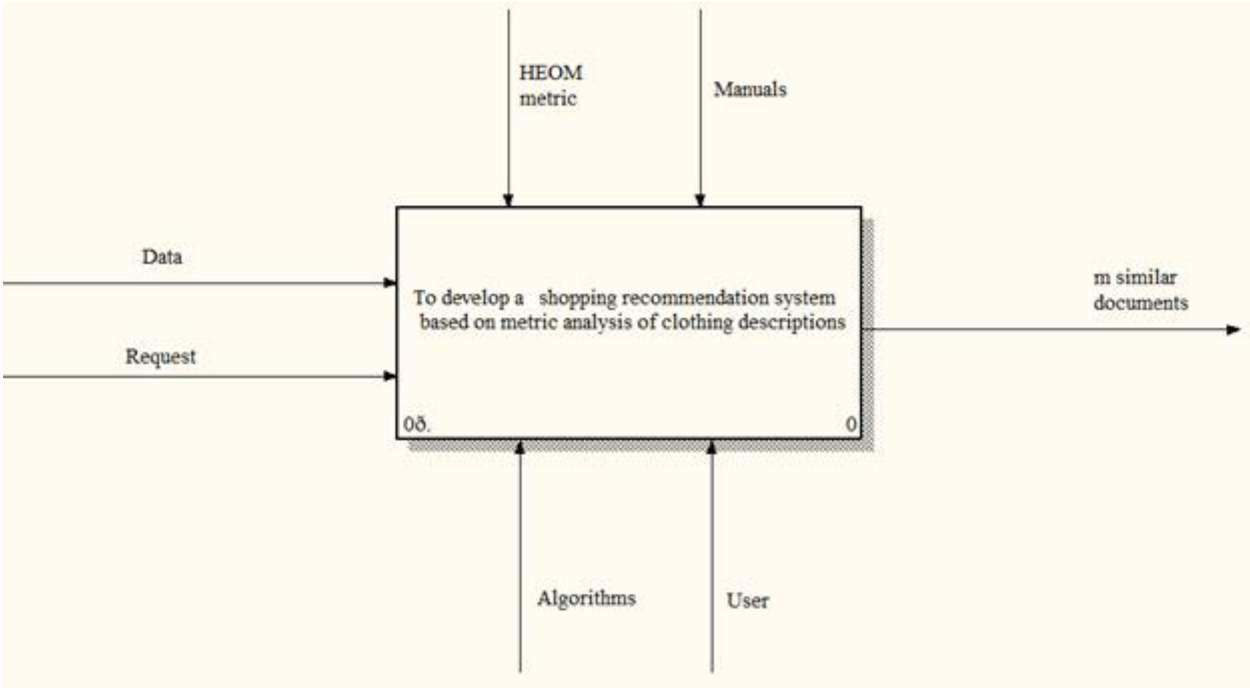


Figure 4.2.1 – The contextual diagram "A0 To develop a shopping recommendation system based on metric analysis of clothing descriptions"

On the top level of the decomposition, it is the model of the shopping recommendation system based on metric analysis of clothing descriptions as IDEF0. The IDEF0 shows an interaction the recommendation system and an external environment. Inputs the system are data and user requests. Data is an information for creating the dataset, dictionary, stopwords. Outputs the system are *m*, similar to a request, documents. The process of development a shopping recommendation system is controlled by user and programmer manuals and HEOM metrics. The system is managed by a user and algorithms (table 4.2.1).

Table 4.2.1 – “To develop a shopping recommendation system based on metric analysis of clothing descriptions”

Name	To develop a shopping recommendation system based on metric analysis of clothing descriptions
Number	A0
Input Arrow(s) of "To develop a shopping recommendation system based on metric analysis of clothing descriptions" Activity	
Definition	**Name**
Information data for the dataset, dictionary, stopwords, corpus	Data
A request from a user – description of the clothing: text description, price, color	Request
Output Arrow(s) of "To develop a shopping recommendation system based on metric analysis of clothing descriptions"	
Definition	**Name**
m documents, similar to a one user-selected document	*m* similar documents
Control Arrow(s) of "To develop a shopping recommendation system based on metric analysis of clothing descriptions" Activity	
Definition	**Name**
HEOM metric measures the distance between a request and responses, between two documents	HEOM metrics
Software user manual for users and programmers, manuals for the system development	Manuals
Mechanism Arrow(s) of "To develop a shopping recommendation system based on metric analysis of clothing descriptions" Activity	
Definition	**Name**
Users who use the recommendation system	User
Algorithms are used in the recommendation system	Algorithms

4.3. Functional decomposition of the diagram IDEF0

There are the functional decomposition of the diagram IDEF0. The decomposition shows you the main stages of the system. You can see the decompositions of the shopping recommendation system: “preprocessing data” and “choose *m* similar documents”.

The first level of the decomposition consists of three related works: “preprocessing data”, “choose *k* similar documents”, “choose *m* similar documents”. The works performed by algorithms and a user using manuals and HEOM metric.

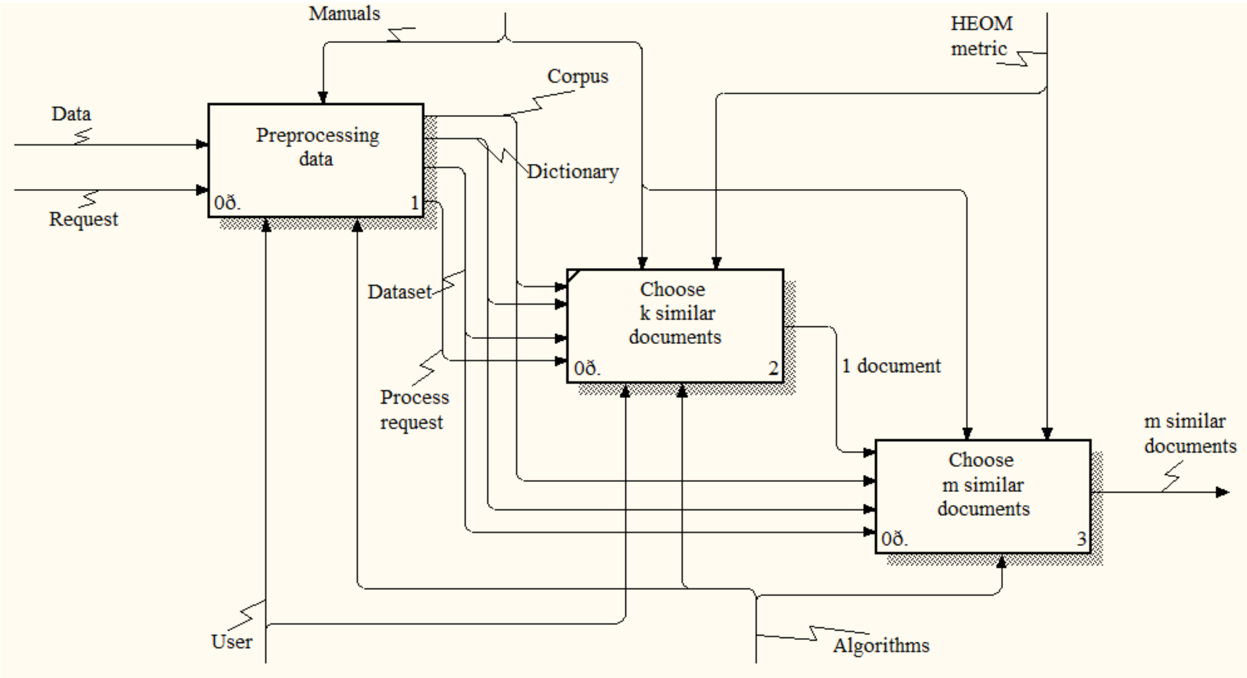


Figure 4.3.1 – The first level of the decomposition “To develop a shopping recommendation system based on metric analysis of clothing descriptions”

The work “Preprocessing data” transforms input data and a request into a corpus, a dictionary, a dataset and a process request. Manuals control the work. A user and algorithms manage it (table 4.3.1).

The work “Choose k similar documents” uses input process request, a corpus, a dictionary and a dataset to get similar documents for the request, using HEOM metric. A user chooses one similar document. It is an output of the work. Manuals control the work. Algorithms manage it (table 4.3.2).

The work “Choose *m* similar documents” uses one input user-selected document, a corpus, a dictionary and a dataset to get *m* similar documents for the one user-selected document, using HEOM metric. It is an output. Manuals control the work. Algorithms manage it (table 4.3.3).

Table 4.3.1 – The first level of the decomposition “Preprocessing data”

Name	Preprocessing data
Number	A1
Definition	The work “Preprocessing data” transforms input data and a request into a corpus, a dictionary, a dataset and a process request. Manuals control the work. A user and algorithms manage it.
Input Arrow(s) of "Preprocessing data" Activity	
Definition	**Name**
Information data for the dataset, dictionary, stopwords, corpus	Data
A request from a user – description of the clothing: text description, price, color	Request
Output Arrow(s) of "Preprocessing data" Activity	
Definition	**Name**
a structured set of texts from a dataset	Corpus
a collection of words from a dataset and requests	Dictionary
a collection of the documents – descriptions of clothes. 37 features: 1 text, 20 linear, 16 nominal values	Dataset
a process user request	Process request
Control Arrow(s) of "Preprocessing data" Activity	
Definition	**Name**
Software user manual for users and programmers, manuals for the system development	Manuals
Mechanism Arrow(s) of "Preprocessing data" Activity	
Definition	**Name**
Users who use the recommendation system	User
Algorithms are used in the recommendation system	Algorithms

Table 4.3.2 – The first level of the decomposition “Choose *k* similar documents”

Name	Choose *k* similar documents
Number	A2
Definition	The work “Choose *k* similar documents” uses input process request, a corpus, a dictionary and a dataset to get similar documents for the request, using HEOM metric. A user chooses one similar document. It is an output of the work. Manuals control the work. Algorithms manage it.
Input Arrow(s) of "Choose *k* similar documents" Activity	
Definition	**Name**
a structured set of texts from a dataset	Corpus
a collection of words from a dataset and requests	Dictionary
a collection of the documents – descriptions of clothes. 37 features: 1 text, 20 linear, 16 nominal values	Dataset
a process user request	Process request
Output Arrow(s) of "Choose *k* similar documents" Activity	
Definition	**Name**
a document of a user choice – one user-selected clothes	1 document
Control Arrow(s) of "Choose *k* similar documents" Activity	
Definition	**Name**
Software user manual for users and programmers, manuals for the system development	Manuals
HEOM metric measures the distance between a request and responses, between two documents	HEOM metric
Mechanism Arrow(s) of " Choose *k* similar documents " Activity	
Definition	**Name**
Users who use the recommendation system	User
Algorithms are used in the recommendation system	Algorithms

Table 4.3.3 – The first level of the decomposition “Choose *m* similar documents”

<table>
<tr><td colspan="2">Name</td><td colspan="2">Choose m similar documents</td></tr>
<tr><td colspan="2">Number</td><td colspan="2">A3</td></tr>
<tr><td colspan="2">Definition</td><td colspan="2">The work “Choose m similar documents” uses one input user-selected document, a corpus, a dictionary and a dataset to get m similar documents for the one user-selected document, using HEOM metric. It is an output. Manuals control the work. Algorithms manage it.</td></tr>
<tr><td colspan="4">Input Arrow(s) of "Choose m similar documents" Activity</td></tr>
<tr><td colspan="3">Definition</td><td>Name</td></tr>
<tr><td colspan="3">a document of a user choice – one user-selected clothes</td><td>1 document</td></tr>
<tr><td colspan="3">a structured set of texts from a dataset</td><td>Corpus</td></tr>
<tr><td colspan="3">a collection of words from a dataset and requests</td><td>Dictionary</td></tr>
<tr><td colspan="3">a collection of the documents – descriptions of clothes. 37 features: 1 text, 20 linear, 16 nominal values</td><td>Dataset</td></tr>
<tr><td colspan="4">Output Arrow(s) of "Choose m similar documents" Activity</td></tr>
<tr><td colspan="3">Definition</td><td>Name</td></tr>
<tr><td colspan="3">m documents, similar to a one user-selected document</td><td>m similar documents</td></tr>
<tr><td colspan="4">Control Arrow(s) of "Choose m similar documents" Activity</td></tr>
<tr><td colspan="3">Definition</td><td>Name</td></tr>
<tr><td colspan="3">Software user manual for users and programmers, manuals for the system development</td><td>Manuals</td></tr>
<tr><td colspan="3">HEOM metric measures the distance between a request and responses, between two documents</td><td>HEOM metric</td></tr>
<tr><td colspan="4">Mechanism Arrow(s) of "Choose m similar documents" Activity</td></tr>
<tr><td colspan="3">Definition</td><td>Name</td></tr>
<tr><td colspan="3">Algorithms are used in the recommendation system</td><td>Algorithms</td></tr>
</table>

The second level of the decomposition “Preprocessing data” consists of five related works: “create stopwords”, “create dataset”, “create dictionary”, “create corpus”, “to process request”. The works performed by algorithms and a user using manuals (figure 4.3.2).

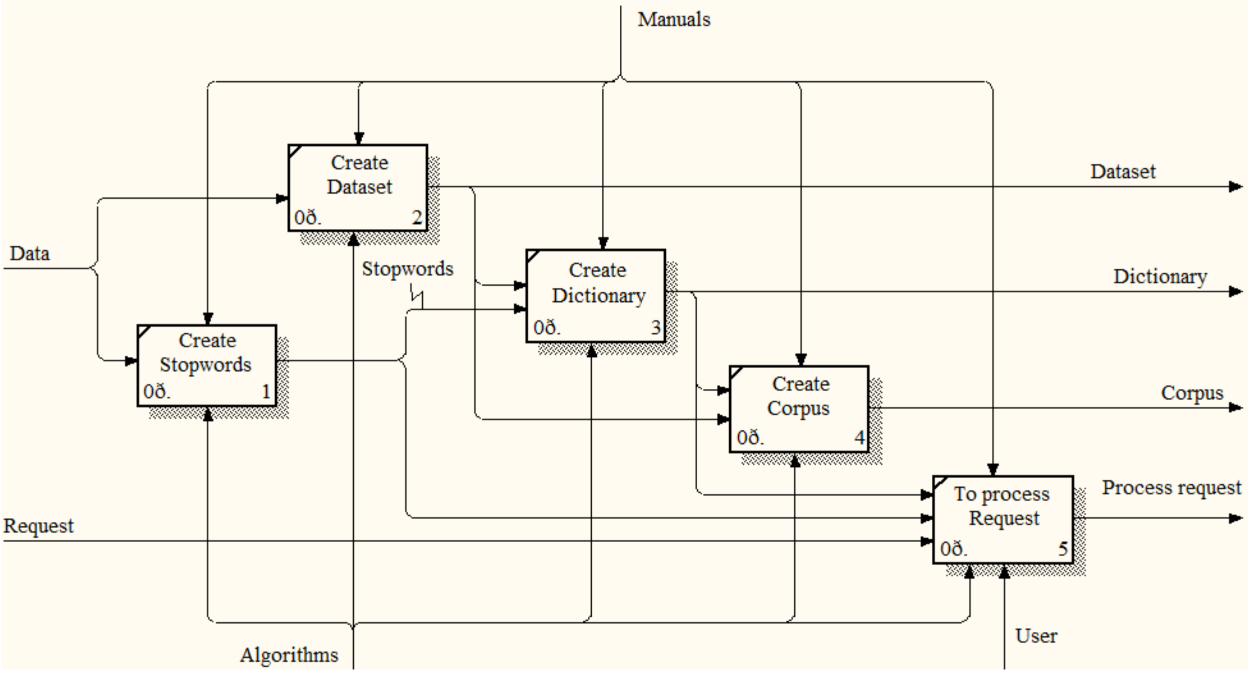


Figure 4.3.2 – The second level of the decomposition “Preprocessing data”

The work “Create Stopwords” uses input data to transform it into stopwords. The stopwords are an output. Manuals control the work. Algorithms manage it (table 4.3.4).

The work “Create Dataset” uses input data to transform it into a dataset. The dataset is an output. Manuals control the work. Algorithms manage it (table 4.3.5).

The work “Create Dictionary” uses a dataset and stopwords to create a dictionary. The dictionary is an output. Manuals control the work. Algorithms manage it (table 4.3.6).

The work “Create Corpus” uses a dataset and a dictionary to create a corpus. The corpus is an output. Manuals control the work. Algorithms manage it (table 4.3.7).

The work “To process Request” uses a dictionary, stopwords and a request to create a process request. The process request is an output. Manuals control the work. Algorithms manage it. A user manages a request (table 4.3.8).

Table 4.3.4 – The second level of the decomposition “Create Stopwords”

Name	Create Stopwords
Number	A11
Definition	The work “Create Stopwords” uses input data to transform it into stopwords. The stopwords are an output. Manuals control the work. Algorithms manage it.
Input Arrow(s) of "Create Stopwords" Activity	
Definition	**Name**
Information data for the dataset, dictionary, stopwords, corpus	Data
Output Arrow(s) of "Create Stopwords" Activity	
Definition	**Name**
words which are filtered out before processing data. Stopwords are interjections, particles, prepositions, pronouns, question words and unions	Stopwords
Control Arrow(s) of "Create Stopwords" Activity	
Definition	**Name**
Software user manual for users and programmers, manuals for the system development	Manuals
Mechanism Arrow(s) of "Create Stopwords" Activity	
Definition	**Name**
Algorithms are used in the recommendation system	Algorithms

Table 4.3.5 – The second level of the decomposition “Create Dataset”

Name	Create Dataset
Number	A12
Definition	The work “Create Dataset” uses input data to transform it into a dataset. The dataset is an output. Manuals control the work. Algorithms manage it.
Input Arrow(s) of "Create Dataset" Activity	
Definition	**Name**
Information data for the dataset, dictionary, stopwords, corpus	Data
Output Arrow(s) of "Create Dataset" Activity	
Definition	**Name**
a collection of the documents – descriptions of clothes. 37 features: 1 text, 20 linear, 16 nominal values	Dataset
Control Arrow(s) of "Create Dataset" Activity	
Definition	**Name**
Software user manual for users and programmers, manuals for the system development	Manuals
Mechanism Arrow(s) of "Create Dataset" Activity	
Definition	**Name**
Algorithms are used in the recommendation system	Algorithms

Table 4.3.6 – The second level of the decomposition “Create Dictionary”

Name	Create Dictionary	
Number	A13	
Definition	The work “Create Dictionary” uses a dataset and stopwords to create a dictionary. The dictionary is an output. Manuals control the work. Algorithms manage it.	
Input Arrow(s) of "Create Dictionary" Activity		
Definition		**Name**
a collection of the documents – descriptions of clothes. 37 features: 1 text, 20 linear, 16 nominal values		Dataset
words which are filtered out before processing data. Stopwords are interjections, particles, prepositions, pronouns, question words and unions		Stopwords
Output Arrow(s) of "Create Dictionary" Activity		
Definition		**Name**
a collection of words from a dataset and requests		Dictionary
Control Arrow(s) of "Create Dictionary" Activity		
Definition		**Name**
Software user manual for users and programmers, manuals for the system development		Manuals
Mechanism Arrow(s) of "Create Dictionary" Activity		
Definition		**Name**
Algorithms are used in the recommendation system		Algorithms

Table 4.3.7 – The second level of the decomposition “Create Corpus”

Name	Create Corpus
Number	A14
Definition	The work “Create Corpus” uses a dataset and a dictionary to create a corpus. The corpus is an output. Manuals control the work. Algorithms manage it.
Input Arrow(s) of "Create Corpus" Activity	
Definition	**Name**
a collection of words from a dataset and requests	Dictionary
a collection of the documents – descriptions of clothes. 37 features: 1 text, 20 linear, 16 nominal values	Dataset
Output Arrow(s) of "Create Corpus" Activity	
Definition	**Name**
a structured set of texts from a dataset	Corpus
Control Arrow(s) of "Create Corpus" Activity	
Definition	**Name**
Software user manual for users and programmers, manuals for the system development	Manuals
Mechanism Arrow(s) of "Create Corpus" Activity	
Definition	**Name**
Algorithms are used in the recommendation system	Algorithms

Table 4.3.8 – The second level of the decomposition “To process Request”

Name	To process Request
Number	A15
Definition	The work “To process Request” uses a dictionary, stopwords and a request to create a process request. The process request is an output. Manuals control the work. Algorithms manage it. A user manages a request.

Input Arrow(s) of "To process Request" Activity	
Definition	**Name**
a collection of words from a dataset and requests	Dictionary
words which are filtered out before processing data. Stopwords are interjections, particles, prepositions, pronouns, question words and unions	Stopwords
A request from a user – description of the clothing: text description, price, color	Request
Output Arrow(s) of "To process Request" Activity	
Definition	**Name**
a process user request	Process request
Control Arrow(s) of "To process Request" Activity	
Definition	**Name**
Software user manual for users and programmers, manuals for the system development	Manuals
Mechanism Arrow(s) of "To process Request" Activity	
Definition	**Name**
Algorithms are used in the recommendation system	Algorithms
Users who use the recommendation system	User

The second level of the decomposition “Choose *m* similar documents” consists of four related works: “text processing”, “linear scale processing”, “nominal scale processing” and “page rank”. The works performed by algorithms and a user using manuals and HEOM metric. (figure 4.3.3).

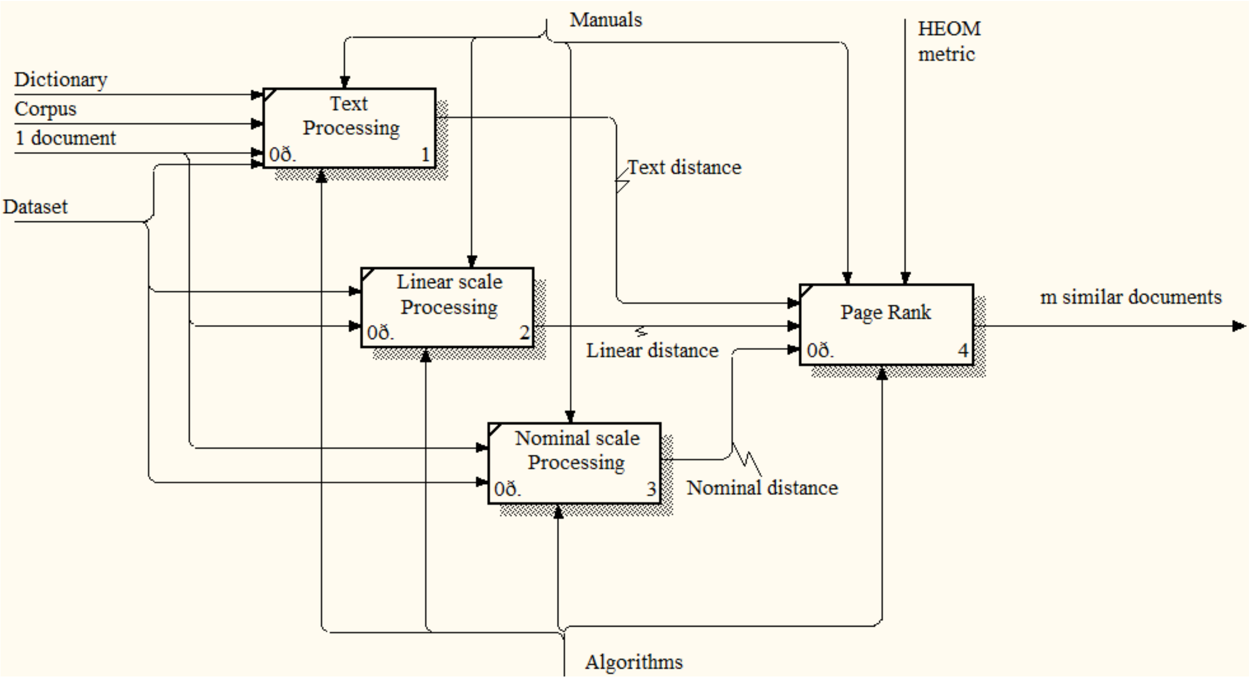


Figure 4.3.3 –The second level of the decomposition “Choose *m* similar documents”

The work “Text Preprocessing” uses a dictionary, a corpus, 1 document and a dataset to measure distance between text features of 1 document and documents from a dataset. A text distance is an output. Manuals control the work. Algorithms manage it (table 4.3.9).

The work “Linear scale Processing” uses 1 document and a dataset to measure distance between linear features of 1 document and documents from a dataset. A linear distance is an output. Manuals control the work. Algorithms manage it (table 4.3.10).

The work “Nominal scale Processing” uses 1 document and a dataset to measure distance between nominal features of 1 document and documents from a dataset. A nominal distance is an output. Manuals control the work. Algorithms manage it (table 4.3.11).

The work “Page Rank” uses a text distance, a linear distance and a nominal distance together with HEOM metric to measure a distance between all features of 1 document and documents from a dataset. *m* similar documents are an output. Manuals control the work. Algorithms manage it (table 4.3.12).

Table 4.3.9 – The second level of the decomposition “Text Preprocessing”

Name	Text Preprocessing
Number	A31
Definition	The work “Text Preprocessing” uses a dictionary, a corpus, 1 document and a dataset to measure distance between text features of 1 document and documents from a dataset. A text distance is an output. Manuals control the work. Algorithms manage it.

Input Arrow(s) of "Text Preprocessing" Activity	
Definition	**Name**
a collection of words from a dataset and requests	Dictionary
a structured set of texts from a dataset	Corpus
a document of a user choice – one user-selected clothes	1 document
a collection of the documents – descriptions of clothes. 37 features: 1 text, 20 linear, 16 nominal values	Dataset
Output Arrow(s) of "Text Preprocessing" Activity	
Definition	**Name**
a cosine distance between text features of a request and responses	Text distance
Control Arrow(s) of "Text Preprocessing" Activity	
Definition	**Name**
Software user manual for users and programmers, manuals for the system development	Manuals
Mechanism Arrow(s) of "Text Preprocessing" Activity	
Definition	**Name**
Algorithms are used in the recommendation system	Algorithms

Table 4.3.10 – The second level of the decomposition “Linear scale Processing”

Name	Linear scale Processing
Number	A32
Definition	The work “Linear scale Processing” uses 1 document and a dataset to measure distance between linear features of 1 document and documents from a dataset. A linear distance is an output. Manuals control the work. Algorithms manage it.

Input Arrow(s) of "Linear scale Processing" Activity	
Definition	**Name**
a document of a user choice – one user-selected clothes	1 document
a collection of the documents – descriptions of clothes. 37 features: 1 text, 20 linear, 16 nominal values	Dataset
Output Arrow(s) of "Linear scale Processing" Activity	
Definition	**Name**
an Euclidean distance between linear features of a request and responses	Linear distance
Control Arrow(s) of "Linear scale Processing" Activity	
Definition	**Name**
Software user manual for users and programmers, manuals for the system development	Manuals
Mechanism Arrow(s) of "Linear scale Processing" Activity	
Definition	**Name**
Algorithms are used in the recommendation system	Algorithms

Table 4.3.11 – The second level of the decomposition “Nominal scale Processing”

<table>
<tr><td>Name</td><td colspan="2">Nominal scale Processing</td></tr>
<tr><td>Number</td><td colspan="2">A33</td></tr>
<tr><td>Definition</td><td colspan="2">The work “Nominal scale Processing” uses 1 document and a dataset to measure distance between nominal features of 1 document and documents from a dataset. A nominal distance is an output. Manuals control the work. Algorithms manage it.</td></tr>
<tr><td colspan="3">Input Arrow(s) of "Nominal scale Processing" Activity</td></tr>
<tr><td colspan="2">Definition</td><td>Name</td></tr>
<tr><td colspan="2">a document of a user choice – one user-selected clothes</td><td>1 document</td></tr>
<tr><td colspan="2">a collection of the documents – descriptions of clothes. 37 features: 1 text, 20 linear, 16 nominal values</td><td>Dataset</td></tr>
<tr><td colspan="3">Output Arrow(s) of "Nominal scale Processing" Activity</td></tr>
<tr><td colspan="2">Definition</td><td>Name</td></tr>
<tr><td colspan="2">a distance between nominal features of a request and responses (0 or 1)</td><td>Nominal distance</td></tr>
<tr><td colspan="3">Control Arrow(s) of "Nominal scale Processing" Activity</td></tr>
<tr><td colspan="2">Definition</td><td>Name</td></tr>
<tr><td colspan="2">Software user manual for users and programmers, manuals for the system development</td><td>Manuals</td></tr>
<tr><td colspan="3">Mechanism Arrow(s) of "Nominal scale Processing" Activity</td></tr>
<tr><td colspan="2">Definition</td><td>Name</td></tr>
<tr><td colspan="2">Algorithms are used in the recommendation system</td><td>Algorithms</td></tr>
</table>

Table 4.3.12 – The second level of the decomposition “Page Rank”

Name	Page Rank
Number	A34
Definition	The work “Page Rank” uses a text distance, a linear distance and a nominal distance together with HEOM metric to measure a distance between all features of 1 document and documents from a dataset. *m* similar documents are an output. Manuals control the work. Algorithms manage it.
Input Arrow(s) of "Page Rank" Activity	
Definition	**Name**
a cosine distance between text features of a request and responses	Text distance
an Euclidean distance between linear features of a request and responses	Linear distance
a distance between nominal features of a request and responses (0 or 1)	Nominal distance
Output Arrow(s) of "Page Rank" Activity	
Definition	**Name**
m documents, similar to a one user-selected document	*m* similar documents
Control Arrow(s) of "Page Rank" Activity	
Definition	**Name**
Software user manual for users and programmers, manuals for the system development	Manuals
HEOM metric measures the distance between a request and responses, between two documents	HEOM metric
Mechanism Arrow(s) of "Page Rank" Activity	
Definition	**Name**
Algorithms are used in the recommendation system	Algorithms

In the fig. 4.3.4 you see a node tree diagram of the shopping recommendation system.

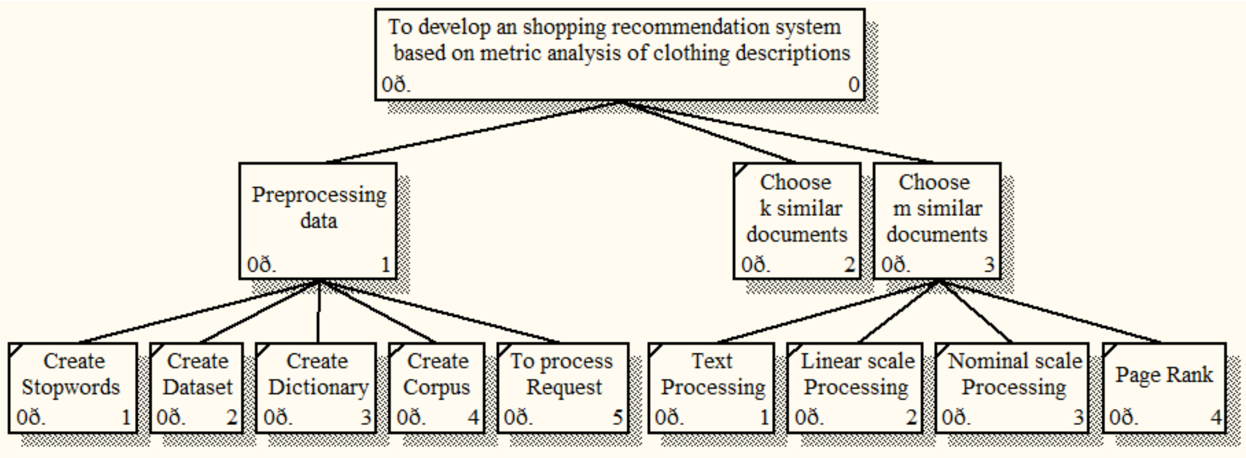


Figure 4.3.4 – Node tree diagram of the shopping recommendation system

In this part, you saw the functional diagram and decompositions of the shopping recommendation system in notation IDEF0. You saw the node tree diagram. The diagrams give us better understanding of a functionality and a structure of the shopping system.

5. Computational experiment

The purpose of the computational experiment is to verify the adequacy of the proposed mixed metric for solving the problem of training ranking.

Firstly, calculate the quality criteria the recommendation system without importance of features (each $a_q = 1$).

Next, calculate the quality criteria of the system using coefficients after optimization methods. Finally, compare two results and write a conclusion.

5.1. Quality criteria

P@20

Precision is the fraction of retrieved documents that are relevant to the query.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrivied documents}\}|} \tag{34}$$

Precision takes all retrieved documents into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system. This measure is called precision at n or P@n.

In our case, topmost results, n = 20. We use P@20.

$$(\text{P@20})_m = \frac{|\mathbb{Y}_r^m|}{20}, \tag{35}$$

where $m = 1 \ldots 100$.

MAP

MAP – mean average precision. MAP is the single-number measure for comparing search algorithms. Mean average precision for a set of queries is the mean of the average precision scores for each query.

$$\text{MAP} = \frac{1}{m} \cdot \sum_{j=1}^{m} \frac{1}{|\mathbb{Y}_r^j|} \cdot \sum_{i=1}^{|\mathbb{Y}_r^j|} P(\text{doc}_i) \tag{36}$$

where:

$|\mathbb{Y}_r^j|$ – number of relevant documents for query *j*;

m – number of queries;

$P(\text{doc}_i)$ – precision at *i*th relevant document.

DCG

DCG – discounted cumulative gain. DCG is a measure of quality of ranking. It is a measure of web search engine algorithms or related applications. DCG has two assumptions:

- highly relevant documents are more useful than marginally relevant document;
- the lower the ranked position of a relevant document, the less useful it is for the user, since it is less likely to be examined.

DCG is the total gain accumulated at a particular rank *p*:

$$\mathrm{DCG}_p = rel_1 + \sum_{i=2}^{p} \frac{\mathrm{rel}_i}{\log_i 2} \text{ or} \tag{37}$$

$$\mathrm{DCG}_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(1+i)}. \tag{38}$$

nDCG

nDCG – normalized DCG.

$$\mathrm{nDCG}_p = \frac{\mathrm{DCG}_p}{\mathrm{IDCG_p}}, \tag{39}$$

where $IDCG_p$ – ideal ranking. It first returns the documents with the highest relevance level, then the next highest relevance level, etc. In a perfect ranking algorithm, the DCG_p is the same as the $IDCG_p$ producing an $nDCG_p$ of 1.0.

5.2. Results of the computational experiment

P@20

On figure 5.2.1, you can see the results of the computational experiment: Precision at 20 for 20 ranking responses.

Firstly, calculate P@20 for both cases. For the case with each $a_q = 1$, the minimum value of P@20 is 0,100; the maximum value of P@20 is 0.9500. The mean value is 0.6315 (figure 5.2.1).

For the case with new a_q, after optimization algorithms, the minimum value of P@20 is 0,4000; the maximum value of P@20 is 1.0000. The mean value is 0.8230 (figure 5.2.1).

It means that the precision at 20 with the new values of a_q better than the precision at 20 of initial system with $a_q = 1$.

MAP

On figure 5.2.2, you can see the results of the computational experiment: Mean Average Precision for the 10 different test sets.

Firstly, calculate MAP for both cases. For the case with each $a_q = 1$, the minimum value of MAP is 0,7805; the maximum value of MAP is 0.8055. The mean value is 0.7956 (figure 5.2.2).

For the case with new a_q, after optimization algorithms, the minimum value of MAP is 0,8668; the maximum value of P@20 is 0.9055. The mean value is 0.8811 (figure 5.2.2).

It means that the mean average precision with the new values of a_q better than the mean average precision of initial system with $a_q = 1$.

nDCG

On figure 5.2.3, you can see the results of the computational experiment: normal discounted cumulative gain for 20 ranking responses.

Firstly, calculate DCG and ideal DCG for both cases for 100 test requests. Next, calculate nDCG for 100 test requests. Finally, get mean value of 100 test requests for both cases. For the case with each $a_q = 1$, the minimum value of nDCG is 0.7872; the maximum value of nDCG is 1.0000. The mean value is 0.8467 (figure 5.2.3).

For the case with new a_q, after optimization algorithms, the minimum value of nDCG is 0.8508; the maximum value of nDCG is 1.0000. The mean value is 0.9088 (figure 5.2.3).

It means that the normal discounted cumulative gain with the new values of a_q better than the normal discounted cumulative gain of initial system with $a_q = 1$.

Summarize, the quality of the results of the computational experiment: P@20, MAP, nDCG, for the values after optimization methods are better than before the optimization of parameters a_q.

In a conclusion, the proposed mixed metric for solving the problem of training ranking with the new a_q parameters has better results according to the chosen quality criteria.

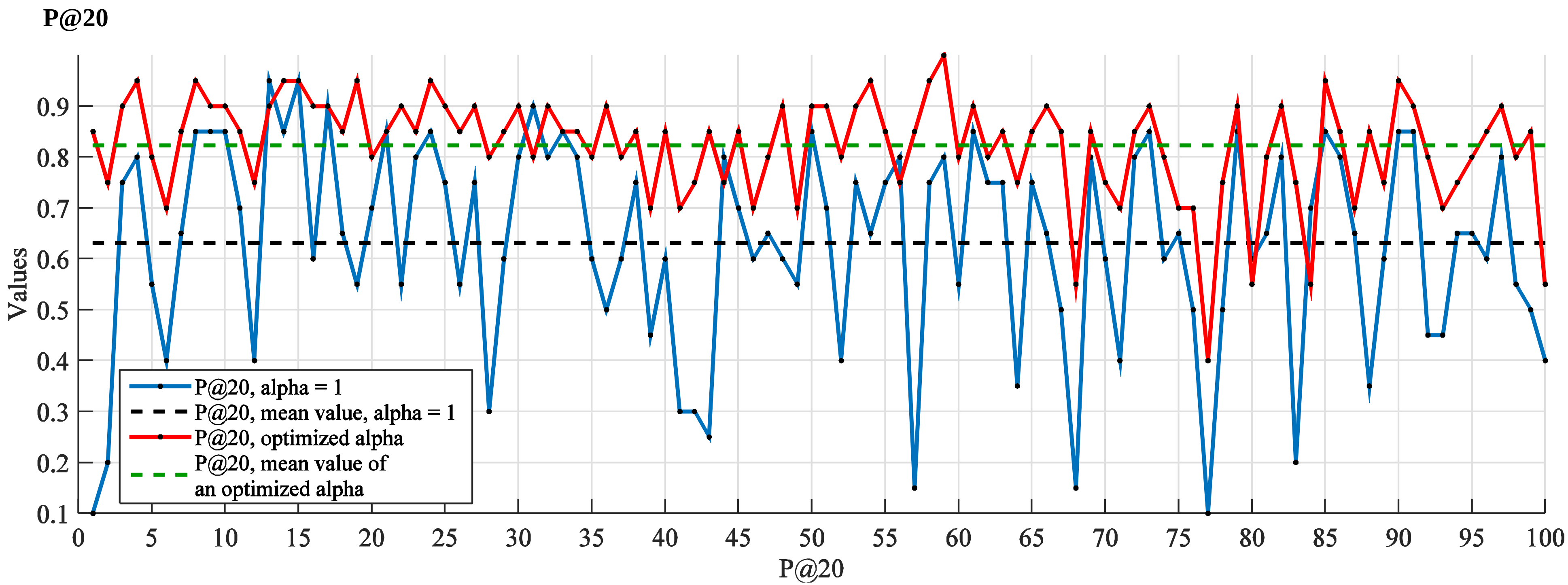


Figure 5.2.1 – Results of P@20 for the ranking models before the optimization (alphas = 1) and after the optimization (new alphas)

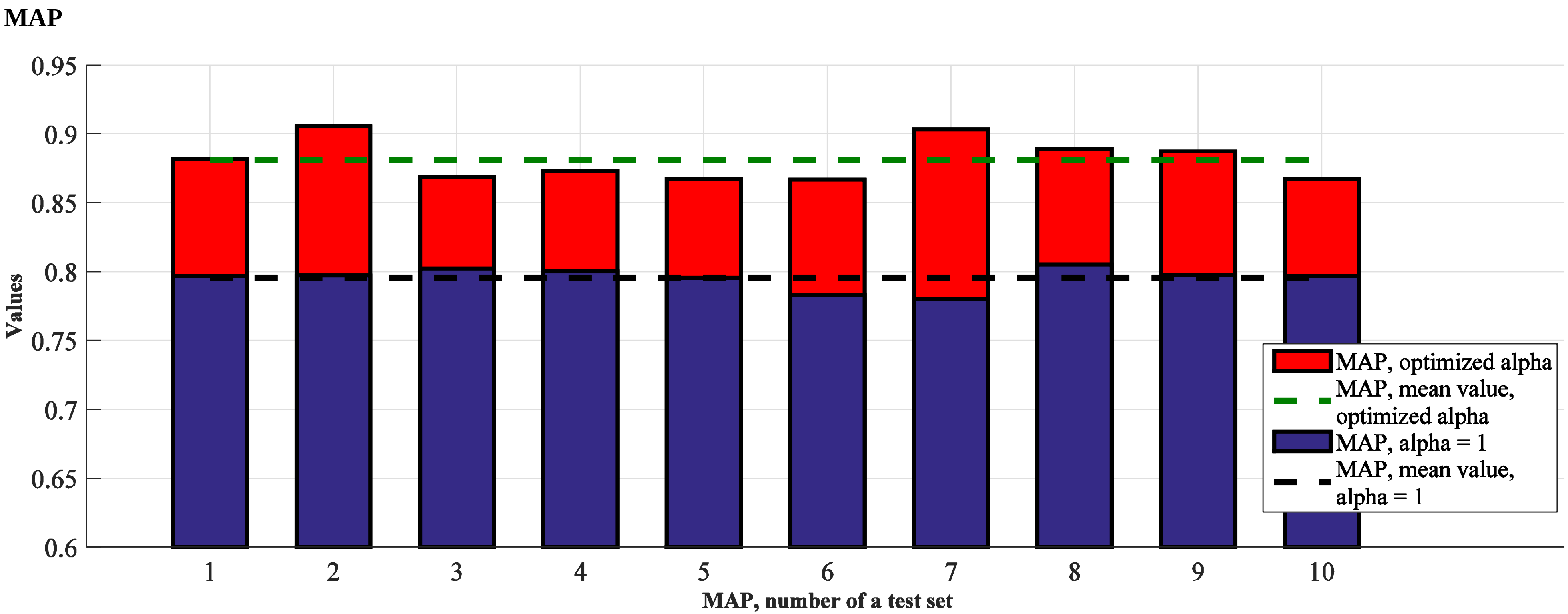


Figure 5.2.2 – Results of MAP for the ranking models before the optimization (alphas = 1) and after the optimization (new alphas). 10 different test sets

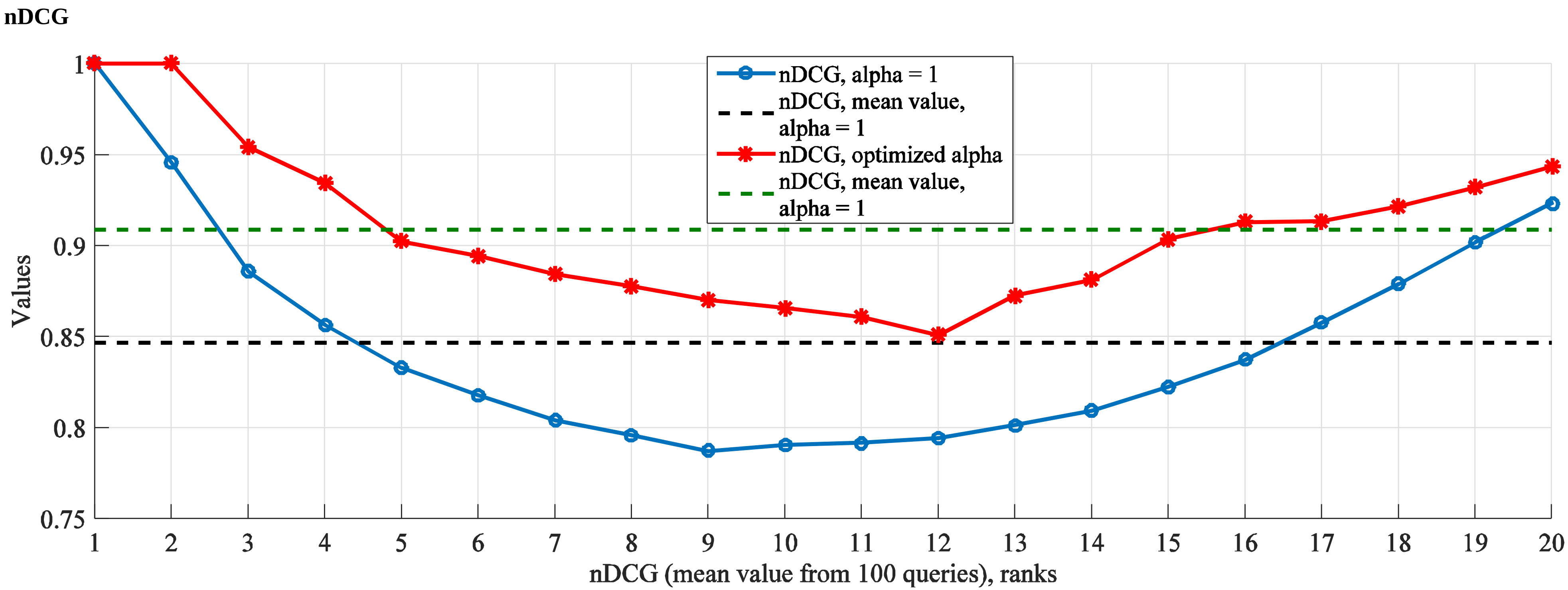
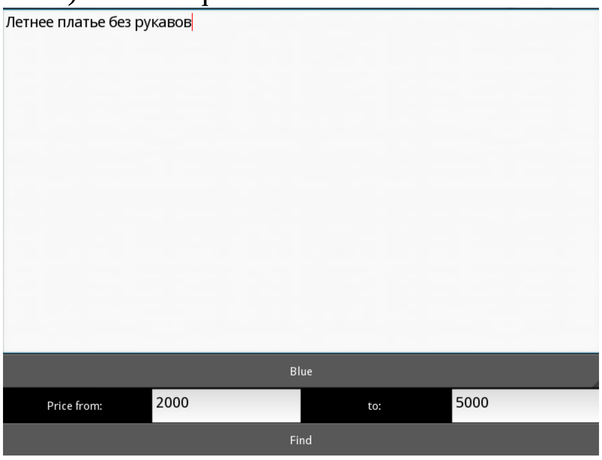


Figure 5.2.3 – Results of nDCG for the ranking models before the optimization (alphas = 1) and after the optimization (new alphas). n = 20.

5.3. Software prototype

In the thesis, there is a software prototype of the shopping recommendation system based on metric analysis of clothing descriptions (table 5.3.1).

Table 5.3.1 – Software prototype of the shopping recommendation system

1) User request 	- A user fills out the form with text description of the wishes clothes, chooses the color (for example “Blue”) and price range (for example “2000” and “5000”). - A user clicks on the button “Find”
2) User chooses one picture by clicking 	- The system finds *k* similar clothes, based on the user request (3 features), each clothes with a picture, a type and a text description. - The user should chooses one the most suitable picture and clicks on it.
3) The final results 	- The system finds *l* similar clothes, based on the user request in step 2 (35 features), each clothes with a picture, a type and a text description.

Conclusion

In this thesis, have developed the shopping recommendation system based on metric analysis of clothing descriptions. Proposed HEOM metric for the mixed scales. Minimax problem of ranking of documents in mixed scales was defined and solved. Elaborated the algorithm of the metric ranking of documents on request. Defined the importance of the features. The most important features for users were “description”, “acrylic”, “cotton”, “subtype”, “color” and “brand”. The computational experiment showed that the results of P@20, MAP, nDCG quality criteria for the ranking, using proposed mixed metric, after optimization methods, were better than before the optimization of parameters a_q. Created the software prototype (technical system) of the shopping recommendation system based on metric analysis of clothing descriptions.

Bibliography

1. Statistics and facts about Online Shopping [Electronic source]. – Access mode: http://www.statista.com/topics/871/online-shopping/, free. – Title screen. – Language English.
2. WalkerSands Communications [Electronic source]. – Access mode: http://www.walkersands.com/, free. – Title screen. – Language English.
3. Online shopping trends 2013: most popular categories, top purchase drivers [Electronic source]. – http://www.marketingprofs.com/charts/2013/12195/online-shopping-trends-most-popular-categories-top-purchase-drivers, free. – Title screen. – Language English.
4. 45% of consumers prefer shopping for clothes online [Electronic source]. – Access mode: https://econsultancy.com/blog/7960-45-of-consumers-prefer-shopping-for-clothes-online/, free. – Title screen. – Language English.
5. Statistics of e-commerce in the world [Electronic source]. – Access mode: http://www.shopolog.ru/metodichka/analytics/statistika-internet-torgovli-v-stranakh-mira, free. – Title from screen. – Language English.
6. China Online Shopping (B2C) Market Report, 2011-2012 [Electronic source]. – Access mode: http://www.bizjournals.com/prnewswire/press_releases/2012/04/02/SP80325, free. – Title from screen. – Language English.
7. Ricci F., Rokach L., Shapira B. Introduction to Recommender Systems (Tutorial) / Intelligent Information Access IIA08 / Cagliari, Italy. – 9-11 December 2008. – P.1-20
8. Ekstrand M., Riedl J., Konstan J. Collaborative Filtering Recommender Systems / Foundations and Trends in Human-Computer Interaction / USA. – 2011. – P.81
9. Melville P., Sindhwani V. Recommender Systems / Encyclopedia of Machine Learning / Springer. – 2010. – P. 829-838.
10. Claypool M., Gokhale A., Miranda T. Combining content-based and collaborative filters in an online newspaper / In Proceedings of the SIGIR-99 Workshop on Recommender Systems: Algorithms and Evaluation / Berkeley, CA, USA. – 1999. – P. 1-15
11. Michael J. Pazzani. A framework for collaborative, content-based and demographic filtering / Artificial Intelligence Review / USA. – 1999. – 13(5-6): P. 393-408
12. Ge, Yong, et al. An energy-efficient mobile recommender system / Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining / ACM. – 2010. – P. 983-991
13. Leskovec J., Rajaraman A., Ullman J. Mining of Massive Datasets / Cambridge University Press /ACM. – 2014. – P. 307-341

14. Adomavicius G., Tuzhilin A. Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions / IEEE Trans. on Data and Knowledge Engineering 17:6 / USA. – 2005. – P. 734–749
15. Shen E., Lam F. Fashion Recommendation and Social Networking based on Commonsense Computing / IUI'12 Proceedings of the 17th international conference on Intelligent user interfaces / ACM. – 2012. – P. 365-368
16. Liu H., Lieberman H., Selker T. A Model of Textual Affect Sensing using Real-World Knowledge / American Psychologist / USA. – 2003. – P. 26-34
17. Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T., and Zhu, W. L. Open Mind Common Sense: Knowledge acquisition from the general public. In Proc. of the 1st International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems. Irvine, CA, 2002.
18. Daily dress me [Electronic source]. – Access mode: http://dailydressme.com/, free. – Title from screen. – Language English.
19. LookBook [Electronic source]. – Access mode: http://lookbook.nu/, free. – Title from screen. – Language English.
20. The sartorialist [Electronic source]. – Access mode: http://www.thesartorialist.com/, free. – Title from screen. – Language English.
21. Style for hire [Electronic source]. – Access mode: http://www.styleforhire.tumblr.com/, free. – Title from screen. – Language English.
22. Zeng Z. An Intelligent E-commerce Recommender System Based on Web Mining / International Journal of Business and Management / Wuhan University. – 2009. – P. 10-14
23. Huang Z, Zeng D, Chen H. A comparison of collaborative-filtering recommendation algorithms for e-commerce / IEEE Intelligent Systems / USA. – 2007. – P. 68-78
24. Surowiecki J. The wisdom of crowds / Random House Digital, Inc / USA. – 2005. – P. 1-330
25. Choa Y., Kimb J., Kim S. A personalized recommender system based on web usage mining and decision tree induction / Expert Systems with Applications / USA. – 2002. – P. 329-342
26. Tu Q., Dong L. An Intelligent Personalized Fashion Recommendation System / Communications, Circuits and Systems (ICCCAS) / Chengdu, China. – 2010. – P. 479-485
27. Ghani R., Fano A. Building Recommender Systems using a Knowledge Base of Product Semantics / AH 2002 / Malaga, Spain. – 2002. – P.15-25
28. Iwata T., Watanabe S., Sawada H. Fashion Coordinates Recommender System Using Photographs from Fashion Magazines / 22 International Joint Conference on Artificial Intelligence / Barcelona, Catalonia, Spain. – 2011. – P.2262-2267

29. Lamche B., Adigüzel U., Wörndl W. Interactive Explanations in Mobile Shopping Recommender Systems / IntRS 2014 / Silicon Valley, CA, USA. – 2014. – P. 111-119
30. Liu S. Hi, Magic Closet, Tell Me What to Wear! / MM’12 / ACM, USA. – 2012. – P. 619-628
31. Sekozawa T. One to One Recommendation System for Apparel Online Shopping / WSEAS transactions on systems / Japan. – 2009. – P. 94-103
32. Yu-Chu L., Kawakita Y., Suzuki E., Ichikawa H. Personalized Clothing-Recommendation System based on a Modified Bayesian Network / SAINT’12 Proceedings of the IEEE/IPSJ International Symposium on Applications and the Internet / Washington, DC, USA. – 2012. – P. 414-417
33. Li X. Sparse representation based visual element analysis / 18th IEEE International Conference on Image Processing / Brussels, Belgium. – 2011. – P. 665-668
34. Shen E., Lieberman H., Lam F. What am I gonna wear?: Scenario-Oriented Recommendation / IUI’07 Proceedings of the 12th international conference on Intelligent user interfaces / ACM, USA. – 2007. – P. 365-368
35. Schafer B., Konstan J., Riedl J. Recommender Systems in E-Commerce / EC’99 Proceedings of the 1st ACM conference on Electronic commerce / ACM, USA. – 1999. – P. 158-166
36. Wilson D., Martinez T. Improved Heterogeneous Distance Functions / Journal of Artificial Intelligence Research / USA. – 1997. – P. 1-34
37. tf-idf [Electronic source]. – Access mode: http://en.wikipedia.org/wiki/Tf-idf, free. – Title from screen. – Language English.
38. Salton G., Buckley C. Term-weighting approaches in automatic text retrieval / Inf. Proc. and Management 24 / USA. – 1988. – P.513-523
39. Gradient descent [Electronic source]. – Access mode: http://en.wikipedia.org/wiki/Gradient_descent, free. – Title from screen. – Language English.
40. Sequential Quadratic Programming [Electronic source]. – Access mode: http://neos-guide.org/content/sequential-quadratic-programming, free. – Title from screen. – Language English.
41. Level of measurement. Nominal scale [Electronic source]. – Access mode: http://en.wikipedia.org/wiki/Level_of_measurement#Nominal_scale, free. – Title from screen. – Language English.
42. Служебные слова [Electronic source]. – Access mode: https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D1%8B%D0%B5_%D1%81%D0%BB%D0%BE%D0%B2%D0%B0, free. – Title from screen. – Language Russian.

Appendices

A. The dataset of the shopping recommendation system http://www.machinelearning.ru/wiki/images/3/31/Dataset_final.zip

B. Optimization algorithms of the shopping recommendation system, quality criteria
Matlab:
https://drive.google.com/open?id=0B_uEcX-Dj4kyVGp6cUhTZVJXNVU&authuser=1

C. Photos of the dataset
https://drive.google.com/open?id=0B_uEcX-Dj4kyd0MxX3F6cElFVDA&authuser=1

D. The code of the software prototype of the shopping recommendation system based on metric analysis of clothing descriptions
https://drive.google.com/open?id=0B_uEcX-Dj4kyTDhOYk1KZDc5RHM&authuser=1