IEEE Second International Conference on Data Stream Mining & Processing
August 21-25, 2018, Lviv, Ukraine

Online Ranking Learning on Clusters

Leonid Lyubchyk
Computer Mathematics and Data Analysis Dept.
National Technical University "Kharkiv Polytechnic Institute"
Kharkiv, Ukraine
lyubchyk@kpi.kharkiv.edu

Galyna Grinberg
Economic Cybernetics and Management Dept.
National Technical University "Kharkiv Polytechnic Institute"
Kharkiv, Ukraine
glngrinberg@gmail.com

***Abstract*—Online data stream ranking learning problem is considered using training data in the form of a sequence of identical items series, described by a number of features and relative rank within the series. It is assumed that feature values and relative ranks of the same items may vary slightly for different series of observations, and there are stable groups of items with similar properties. In this regard, the problem of learning to rank on clusters is stated, while training dataset consist of estimates of centers of clusters and average rank of the items inside each cluster. A unified approach to ranking learning on clusters using kernel models of utility function is proposed. Recurrent algorithms for estimating the parameters of a utility function model as well as recurrent ranking learning algorithm in the space of conjugate variables are developed.**

***Keywords*—data stream, kernel function, online learning, ranking learning, recurrent estimation, regularization, utility function.**

I. Introduction

The problem of ranking learning has been intensively studied recently due to broad variety of important practical applications, ranging from classical problems of multi-criteria choice of alternatives and decision-making [1] to modern ones, such as information retrieval, machine translation, computational linguistics and biology [2]. The goal of learning to rank is automatic building of a ranking model using training data, which consists of lists of subject items with some partial order specified between items in each list, usually set by indicating some numerical or ordinal score for all items. It is assumed, that ranking mechanism, determined by preferences of users, experts or perhaps some artificial ranking system, is usually unknown. Therefore, ranking learning, in fact, is data-based modeling of this mechanism, so that the results of ranking of elements of a new lists will be is similar to rankings in the training data in some sense.

Especially intensive the problem of ranking learning is studied in connection with the tasks of Internet data processing with the purpose of information retrieval. Ranking learning is widely used in such problems as document retrieval, recommendation system development, search engine modeling and over [3]. For example, in a document retrieval problem, for any given query, a ranking model assigns a relevance score to each document in obtained collection, and then ranks the documents in decreasing order of relevance scores. Therefore, the training data consists of queries and ranked sequence of documents. In this formulation, in particular, the important problem of adaptive modeling of search engine with unknown ranking mechanism is considered, using information concern search results, corresponding to a certain sequence of queries [4].

Learning to rank belongs to the class of supervised machine learning problem, using given training sample consist of some items with measured features and labels, representing its ranks. The purpose of learning is to obtain some ranking function estimate, which provides similar ranking results on the test sample. In turn, ranking function is usually found by empirical risk minimization, determined by averaging of certain loss function on the training data sample [5].

As a ranking function model, in preference learning framework, it is often used latent utility function, describes expert or user preferences. Utility function is usually specified in the form of a scalar positive function defined in feature space, while larger values of the utility function correspond to larger values of the ranks.

In practice, linear models of utility function in the form of a weighted sum of features were widely used, while weights determine the relative features importance [6], these weights are found by applying expert or statistical methods.

In fact, the linear utility function model does not always adequately reflect the real structure of user preferences, and the structure of the utility function, reflecting the actual ranking mechanisms, can be significantly more complex.

At present, a number of heuristic approaches to the choice of utility function non-linear model structure are proposed as a fairly simple functional dependencies [7, 8], but revealing its true form remains a difficult task.

More general non-linear models of utility functions can be chosen in the form of a linear combination of some pre-determined coordinate functions. In order to build a qualitative approximate model, it is necessary to use a large number of coordinate functions, the consequence is the need for high dimension model parameters vector estimating, which leads to significant computational problems.

Since the latent utility function can have a very complex structure and previous information about its structure is usually absent, it is advisable to use kernel-based machine learning technique [9]. In the framework of this method, utility functions estimate can be represented as a linear combination of kernel functions at training points. However, due to "kernel trick", there is no need for preliminary specification of a set of coordinate functions, which makes it possible to build models of limited complexity that successfully approximate rather complex utility functions [10].

978-1-5386-2874-4/18/$31.00 ©2018 IEEE

In many practical applications training data are generated as a data streams in the form of consecutive series of observations, arrives continuously in variable time [11]. Under the frequently changing items features and ranking results, it is advisable to use online ranking learning methods that provide the opportunity for effective training of ranking model in real time [12, 13].

The peculiarity of considered problem is in the fact that for the same items in different series of data streams the observed values of features and relative ranks can vary, because its properties and user preferences can change over time. Consequently, it is impossible to assign to each particular item the exact rank and it is reasonable to use some averaged ranks as supervised information. It is assumed that feature values and relative ranks of the same items may vary slightly for different series of observations, and there are stable groups of items with similar properties. This predetermines the need for prior aggregation of the ranked items into certain groups of similar properties by clustering them in feature space. This aggregation allows moving from the problem of ranking objects to the problem of ranking clusters. In this regard, it is reasonable to use *ranking learning on clusters* approach [14], based on preliminary clustering of ranking items followed by utility function model building using average ranks of items inside each cluster.

In the tasks of online ranking learning, the data stream is formed as sequences of series of the same items consist from number of observations of features and relative ranks within each series. The specificity of online ranking learning using data stream requires learning algorithms in recurrent form, wherein the number of the iteration step coincides with the number of series of observations.

In this paper, a unified approach to the cluster ranking recurrent learning algorithms development using kernel models of utility function is proposed. First, recursive algorithms for estimating the parameters of a linear utility function model are considered. In this case, expert estimates of features weights are used as *a priori* information for regularizing the estimation problem, realizing, in fact, optimal concordation of expert and statistical estimates [9, 15]. Further, based on kernel approach, a learning non-linear utility function model is obtained, while the estimates of the parameters of the linear model are used for regularization of optimized functional of empirical risk. On top of that, another one algorithm of recurrent ranking learning in the space of conjugate variables is also proposed.

II. Problem Statement

Consider the ranking learning problem for the set of same items $x \in \Omega$, characterized by its feature vector $\mathbf{x}^{\mathrm{T}} = (x^1, x^2, ..., x^N)$.

Supposed that training data are generated in real time and is representable as a sequence of observations series. Each series includes observations on the entire set of ranked items and has fixed length L.

In such a case, in each data stream series n training dataset are presented by the data matrix $\mathbf{X}_n = \{x_i^j(n)\}_{i,j=1}^{L,N}$, consists from feature observations

$$\mathbf{x}_i^{\mathrm{T}}(n) = (x_i^1(n), x_i^2(n), ..., x_i^N(n)), \ i = \overline{1, ..., L}.$$

Each item within the any stream observation series is assigned its relative rank $r_n(\mathbf{x}_i(n)), 1 \le i \le n,$ defined by some ranking function. It is assumed that the specified ranking function is unknown, and only relative ranks for any objects in each series are available to observation.

The ranking function is usually described by some scalar positive continuous utility function $f(\mathbf{x})$, such that $f(\mathbf{x}_i) > f(\mathbf{x}_j)$ *if* $r_n(\mathbf{x}_i(n)) < r_n(\mathbf{x}_j(n))$.

The problem of data stream learning to rank is to restore unknown utility function by finding its estimate $\hat{f}(\mathbf{x})$ using available sequence of observations series $\{\mathbf{X}_n, \mathbf{r}_n\}$, $\mathbf{r}_n^{\mathrm{T}} = (r_n(\mathbf{x}_1(n)), \ ... \ , r_n(\mathbf{x}_L(n)))$.

Suppose that using the appropriate clustering method the set of ranking items is divided into a set of M clusters $\{\Omega_m\}_{m=1}^{M}$ in feature space, described by a set of parameters $(\bar{\mathbf{x}}^m, \bar{r}^m)$, $m = \overline{1, M}$

$$\bar{\mathbf{x}}^m = \frac{1}{|J^m|} \sum_{i \in \Omega_m} \mathbf{x}_i, \quad \bar{r}^m = \frac{1}{|J^m|} \sum_{i \in \Omega_m} r(\mathbf{x}_i), \qquad (1)$$

where $\bar{\mathbf{x}}^m$ is center vector of m-th cluster and $\bar{r}^m$ is average rank of items belonging to the same cluster, $J^m = \{i \mid x_i \in \Omega_m\}$, $m = \overline{1, M}$.

Then, to restore the utility function, aggregated training data is used in the form of a sequence of estimated centers of clusters and average ranks of items inside the corresponding cluster for each data stream series of observations $\bar{\mathbf{X}}_n^{\mathrm{T}} = (\bar{\mathbf{x}}^1(n), ..., \bar{\mathbf{x}}^M(n))$, $\bar{\mathbf{r}}_n^{\mathrm{T}} = (\bar{r}^1(n), ..., \bar{r}^M(n))$.

We take utility function model in the quasilinear form $f(\mathbf{x}) = \varphi^{\mathrm{T}}(\mathbf{x})\mathbf{c}$, where $\mathbf{c}^{\mathrm{T}} = (c^1, ..., c^D)$ – vector of utility model parameters, $\varphi^{\mathrm{T}}(\mathbf{x}) = (\varphi^1(\mathbf{x}), ..., \varphi^D(\mathbf{x}))$ – vector of model coordinate functions, D is a model dimension.

Then the problem of ranking learning based on streaming data by restoring the utility function on clusters is reduced to finding model parameters estimates $\hat{\mathbf{c}}_n$ using a sequence of aggregated streaming training data $\{\bar{\mathbf{X}}_n, \bar{\mathbf{r}}_n\}$, $n = 1, 2, ...$.

In the kernel-based learning framework coordinate functions are taken hereby that its scalar products will be positive definite functions $\varphi^{\mathrm{T}}(\mathbf{x})\varphi(\mathbf{x}') = \kappa(\mathbf{x}, \mathbf{x}')$, at that utility function model are linear combination of kernel function, located in centers of clusters

$$\hat{f}(\mathbf{x}) = \sum_{m=1}^{M} d_m \cdot \kappa(\mathbf{x}, \bar{\mathbf{x}}^m), \qquad (2)$$

where d_m, $m = \overline{1, M}$ – kernel-based utility function model parameters are determined by center and average rank of clusters estimates.

Then the problem of online rankings kernel-based learning on clusters is reduced to the construction of a recurrent algorithm for estimating parameters d_m, $m=\overline{1,M}$ of kernel model (2) based on training data stream $\{\bar{\mathbf{X}}_n, \mathbf{r}_n\}$, $n=1,\ldots$.

III. Liner Utility Function Model Identification

We first consider the problem of estimating the parameters of the linear model of utility function

$$f(\mathbf{x})=\mathbf{x}^{\mathrm{T}}\mathbf{w}, \tag{3}$$

where $\mathbf{w}^{\mathrm{T}}=(w^1,w^2,\ldots,w^N)$ – vector of linear utility function model parameters.

To construct model parameter estimates, streaming training data $\{\bar{\mathbf{X}}_n, \mathbf{r}_n\}$, $n=1,\ldots$ is used. Because the elements of this sequence are random vectors and matrices, at the data preprocessing stage it is advisable to smooth the sequence of the training sample elements using a suitable current averaging algorithm, for example, the method of exponential smoothing:

$$\begin{aligned}&\hat{\mathbf{X}}_{n+1}=\tau_x\cdot\hat{\mathbf{X}}_n+(1-\tau_x)\cdot\bar{\mathbf{X}}_{n+1},\\&\hat{\mathbf{r}}_{n+1}=\tau_r\cdot\hat{\mathbf{r}}_n+(1-\tau_r)\cdot\bar{\mathbf{r}}_{n+1},\quad 0<\tau_x,\tau_r<1.\end{aligned} \tag{4}$$

Then the problem of estimating the parameters of the linear model of utility function is reduced to the problem of multi-dimensional dynamic linear regression

$$\hat{\mathbf{r}}_{n+1}=\hat{\mathbf{X}}_{n+1}\mathbf{w}+\mathbf{e}^w_{n+1}, \tag{5}$$

where $\mathbf{e}^w_{n+1}=(e^1_{n+1}\ e^2_{n+1}\ldots e^M_{n+1})^{\mathrm{T}}$ – vector of average rank om clusters estimation errors.

Let us find the current estimate of the vector of linear model parameters from the condition of minimization of a one-step regularized functional with constraints

$$\begin{aligned}&R^w_{n+1}(\mathbf{w})=\left\|\mathbf{e}_{n+1}\right\|^2+\alpha\left\|(\mathbf{w}-\hat{\mathbf{w}}_n)\right\|^2\to\min_w,\\&\mathbf{e}^w_{n+1}=\hat{\mathbf{r}}_{n+1}-\hat{\mathbf{X}}_{n+1}\mathbf{w},\end{aligned} \tag{6}$$

where $\alpha>0$ – regularization parameter, and the previous estimation of the parameter vector $\hat{\mathbf{w}}_n$ is used as *a priori* information for regularization at data stream series $n+1$, which provides the possibility of obtaining a recurrent estimate.

To solve the optimization problem with constraints, we use Lagrange function

$$\begin{aligned}&L(\mathbf{w},\mathbf{e}^w_{n+1},\boldsymbol{\mu})=\\&\quad=0.5\cdot R^w_{n+1}(\mathbf{w})+\boldsymbol{\mu}^{\mathrm{T}}(\hat{\mathbf{r}}_{n+1}-\bar{\mathbf{X}}_{n+1}\mathbf{w}-\mathbf{e}^w_{n+1}),\end{aligned} \tag{7}$$

where μ are Lagrange multipliers.

Using optimality conditions for (7) in the form of the Kuhn-Tucker:

$$\begin{aligned}&\alpha\cdot(\mathbf{w}-\hat{\mathbf{w}}_n)-\hat{\mathbf{X}}^{\mathrm{T}}_{n+1}\boldsymbol{\mu}=0,\ \boldsymbol{\mu}=\mathbf{e}^w_{n+1},\\&\hat{\mathbf{r}}_{n+1}-\hat{\mathbf{X}}_{n+1}\mathbf{w}-\mathbf{e}^w_{n+1}=0,\end{aligned} \tag{8}$$

we obtain an explicit expression for $\hat{\mathbf{w}}_{n+1}$ estimate in the form of a recurrent estimation algorithm

$$\begin{aligned}&\hat{\mathbf{w}}_{n+1}=\Psi^{-1}_{n+1}(\alpha)\cdot(\alpha\cdot\hat{\mathbf{w}}_n+\hat{\mathbf{X}}^{\mathrm{T}}_{n+1}\hat{\mathbf{r}}_{n+1}),\\&\Psi^{-1}_{n+1}(\alpha)=\alpha\cdot\mathbf{I}_N+\hat{\mathbf{X}}^{\mathrm{T}}_{n+1}\hat{\mathbf{X}}_{n+1}.\end{aligned} \tag{9}$$

The obtained algorithm, which relates to the class of one-step regularized projection identification algorithms, allows tracing slow changes in cluster parameters, and the choice of the regularization parameter provides a balance between its tracking and filtering properties (9).

IV. Recursive Nonlinear Preference Learning

Measurement equation for quasilinear utility function model identification at stream series $n+1$ may be represented as following:

$$\hat{r}^m_{n+1}=\hat{f}(\hat{\mathbf{x}}^m_{n+1})=\varphi^{\mathrm{T}}(\hat{\mathbf{x}}^m_{n+1})\mathbf{c}+e^m_{n+1},\ \ m=\overline{0,M}. \tag{10}$$

In matrix form this equation is $\hat{\mathbf{r}}_{n+1}=\boldsymbol{\Phi}^{\mathrm{T}}_{n+1}\mathbf{c}+\mathbf{e}_{n+1}$,

where $\hat{\mathbf{r}}_{n+1}=(\hat{r}^1_{n+1}\ \hat{r}^2_{n+1}\ldots\hat{r}^M_{n+1})^{\mathrm{T}}$ – observation vector, composed from average rank on clusters estimates, $\boldsymbol{\Phi}_{n+1}=\left(\varphi(\hat{\mathbf{x}}^1_{n+1})\ \ \varphi(\hat{\mathbf{x}}^2_{n+1})\ \ \ldots\ \ \varphi(\hat{\mathbf{x}}^M_{n+1})\right)$ – feature matrix estimate, $\mathbf{e}_{n+1}=(e^1_{n+1}\ e^2_{n+1}\ldots e^M_{n+1})^{\mathrm{T}}$ – average rank estimation errors.

Introduce kernel matrix $\mathbf{K}_{n+1}=\boldsymbol{\Phi}^{\mathrm{T}}_{n+1}\boldsymbol{\Phi}_{n+1}$,

$$\mathbf{K}_n=\left\|k_{q,s}\right\|,\ k_{q,s}=\kappa(\hat{\mathbf{x}}^q_n,\hat{\mathbf{x}}^s_n),\ q,s=\overline{1,M},$$

where $\kappa(\mathbf{x},\mathbf{x}')$ is an appropriate kernel function.

The utility function model parameters estimates $\hat{\mathbf{c}}_{n+1}$ at any data stream series $n+1$ may be obtained as a solution of regularized constrained optimization problem

$$\begin{aligned}&R^c_{n+1}(\mathbf{c})=\left\|\mathbf{e}_{n+1}\right\|^2+\beta\left\|(\mathbf{c}-\mathbf{c}^0_{n+1})\right\|^2\to\min_c,\\&\mathbf{e}_{n+1}=\hat{\mathbf{r}}_{n+1}-\boldsymbol{\Phi}^{\mathrm{T}}_{n+1}\mathbf{c},\end{aligned} \tag{11}$$

where $\hat{\mathbf{c}}^0_{n+1}$ – vector of *a priory* value of utility function model parameters for data stream series $n+1$, $\beta>0$ – regularization parameter.

To solve the optimization problem (11) we use the Lagrange function

$$L(\mathbf{c}, \mathbf{e}_{n+1}, \boldsymbol{\lambda}) = 0.5 \cdot R_n(\mathbf{c}) + \\ + \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{r}_{n+1} - \boldsymbol{\Phi}_{n+1}^{\mathrm{T}} \mathrm{c} - \mathbf{e}_{n+1}), \tag{12}$$

where $\boldsymbol{\lambda}^{\mathrm{T}} = (\lambda_1, ..., \lambda_M)$ – vector of Lagrange multipliers.

Using the conditions for optimality for (12)

$$\mathbf{c} = \mathbf{c}_{n+1}^0 + \beta^{-1}\boldsymbol{\Phi}_{n+1}\boldsymbol{\lambda}, \quad \boldsymbol{\lambda} = \mathbf{e}_{n+1}, \\ \hat{\mathbf{r}}_{n+1} - \boldsymbol{\Phi}_{n+1}^{\mathrm{T}}\mathbf{c} = \mathbf{e}_{n+1}, \tag{13}$$

model parameters and conjugate variables optimal estimates can be presented in the form

$$\hat{\mathbf{c}}_{n+1} = \boldsymbol{\Phi}_{n+1}\mathbf{A}_{n+1}^{-1}(\beta)\hat{\mathbf{r}}_{n+1} + \\ + (\mathbf{I}_D - \boldsymbol{\Phi}_{n+1}\mathbf{A}_{n+1}^{-1}(\beta)\boldsymbol{\Phi}_{n+1}^{\mathrm{T}})\mathbf{c}_{n+1}^0, \\ \hat{\boldsymbol{\lambda}}_n = \mathbf{A}_{n+1}^{-1}(\beta)(\hat{\mathbf{r}}_{n+1} - \boldsymbol{\Phi}_{n+1}^{\mathrm{T}}\mathbf{c}_{n+1}^0), \\ \mathbf{A}_{n+1}(\beta) = \beta^{-1}\mathbf{I}_M + \mathbf{K}_{n+1}. \tag{14}$$

The use of kernel approach requires the elimination of direct evaluation of model parameters. To do this, we express *a priori* value of utility function model parameters $\mathbf{c}_{n+1}^0$ through available estimates, as which we choose utility function linear approximation parameters $\hat{\mathbf{w}}_{n+1}$ estimates defined by algorithm (9).

To do this, we take the linear model $f^0(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{w}$ as the first approximation of utility function. In accordance with this assumption, we find an optimal *a priory* value of utility function model parameters $\mathbf{c}_{n+1}^0$ at stream series $n+1$ from the condition of best approximation of linear utility function model values $\hat{\mathbf{r}}_{n+1}^0 = \hat{\mathbf{X}}_{n+1}\hat{\mathbf{w}}_{n+1}$, estimated on measured data $\hat{\mathbf{X}}_{n+1}$, by *a priory* average rank vector $\mathbf{r}_{n+1}^0 = \boldsymbol{\Phi}_{n+1}^{\mathrm{T}}\mathbf{c}_{n+1}^0$.

Consequently, to find *a priory* value $\mathbf{c}_{n+1}^0$, consider auxiliary optimization problem for regularized functional:

$$Q_0(\mathbf{c}^0) = \|\boldsymbol{\zeta}\|^2 + \gamma\|\mathbf{c}^0\|^2 \to \min_{c_0}, \\ \boldsymbol{\zeta} = \hat{\mathbf{r}}_{n+1}^0 - \mathbf{r}_{n+1}^0 = \hat{\mathbf{X}}_{n+1}\hat{\mathbf{w}}_{n+1} - \boldsymbol{\Phi}_{n+1}^{\mathrm{T}}\mathbf{c}^0. \tag{15}$$

where $\gamma > 0$ – regularization parameter.

Using Lagrange function for constrained optimization problem

$$L(\mathbf{c}_0, \boldsymbol{\zeta}, \mathbf{v}) = 0.5 \cdot Q_0(\mathbf{c}_0) + \\ + \mathbf{v}^{\mathrm{T}}(\hat{\mathbf{X}}_{n+1}\hat{\mathbf{w}}_{n+1} - \boldsymbol{\Phi}_{n+1}^{\mathrm{T}}\mathbf{c}_0 - \boldsymbol{\zeta}), \tag{16}$$

where $\mathbf{v}^{\mathrm{T}} = (\nu_1, ..., \nu_n)$ – appropriate vector of Lagrange multipliers, we can obtain its solution of problem (15) as

$$\mathbf{c}_{n+1}^0 = \boldsymbol{\Phi}_{n+1}\mathbf{B}_{n+1}^{-1}(\gamma)\hat{\mathbf{X}}_{n+1}\hat{\mathbf{w}}_{n+1}, \\ \mathbf{B}_{n+1}(\gamma) = \gamma\mathbf{I}_M + \mathbf{K}_{n+1}. \tag{17}$$

Taking into account the obvious kernel relation

$$\chi_{n+1}^{\mathrm{T}}(\mathbf{x}) = \varphi^{\mathrm{T}}(\mathbf{x}) \cdot \boldsymbol{\Phi}_{n+1} = \\ = \left(\kappa(\mathbf{x}, \hat{\mathbf{x}}_{n+1}^1), \kappa(\mathbf{x}, \hat{\mathbf{x}}_{n+1}^2), \ldots, \kappa(\mathbf{x}, \hat{\mathbf{x}}_{n+1}^M)\right)^{\mathrm{T}}, \tag{18}$$

the optimal utility function nonlinear model estimate $\hat{f}_{n+1}(\mathbf{x}) = \varphi^{\mathrm{T}}(\mathbf{x})\hat{\mathbf{c}}_{n+1}$ takes the following form:

$$\hat{f}_{n+1}(\mathbf{x}) = \chi_{n+1}^{\mathrm{T}}(\mathbf{x}) \cdot (\mathbf{A}_{n+1}^{-1}(\beta)\hat{\mathbf{r}}_{n+1} + \\ + [\mathbf{I}_M - \mathbf{A}_{n+1}^{-1}(\beta)\mathbf{K}_{n+1})]\mathbf{B}_{n+1}^{-1}(\gamma)\hat{\mathbf{X}}_{n+1}\hat{\mathbf{w}}_{n+1}) \tag{19}$$

Thus, utility function model estimate depends only of kernel function, located in the estimated clusters centers

$$\hat{f}_{n+1}(\mathbf{x}) = \sum_{m=1}^{M} d_m(\hat{\mathbf{x}}_{n+1}, \hat{\mathbf{r}}_{n+1}, \hat{\mathbf{w}}_{n+1}) \cdot \kappa(\mathbf{x}, \hat{\mathbf{x}}_{n+1}^m), \tag{20}$$

where kernel model coefficients d_m, $m = \overline{1, M}$ are calculated recurrently using available data stream.

V. Conjugate Variables Learning Algorithm

A different way of estimating the parameters of the utility function can be proposed using the algorithm for recurrent estimation of conjugate variables [16].

Using the relation (13) with the parameters corresponding to stream series $n+1$ $\mathbf{c}_{n+1} = \beta^{-1}\boldsymbol{\Phi}_{n+1}\boldsymbol{\lambda}_{n+1} + \mathbf{c}_{n+1}^0$ and multiplying (13) by matrix $\boldsymbol{\Phi}_{n+1}^{\mathrm{T}}$, we obtain corresponding measurement equation for conjugate variables

$$\hat{\mathbf{r}}_{n+1} = \beta^{-1}\mathbf{K}_{n+1}\boldsymbol{\lambda}_{n+1} + \boldsymbol{\Phi}_{n+1}^{\mathrm{T}}\mathbf{c}_{n+1}^0 + \mathbf{e}_{n+1}, \\ \mathbf{c}_{n+1}^0 = \boldsymbol{\Phi}_{n+1}\mathbf{B}_{n+1}^{-1}(\gamma)\hat{\mathbf{X}}_{n+1}\hat{\mathbf{w}}_{n+1}. \tag{21}$$

Introduce a local identification criterion [16] as the moving estimation cost includes regularization term, determined by conjugate variables estimate $\hat{\boldsymbol{\lambda}}_n$ at previously data stream series $n+1$:

$$R_{n+1}(\boldsymbol{\lambda}_{n+1}) = \delta \cdot (\boldsymbol{\lambda}_{n+1} - \hat{\boldsymbol{\lambda}}_n)^{\mathrm{T}}\mathbf{K}_{n+1}(\boldsymbol{\lambda}_{n+1} - \hat{\boldsymbol{\lambda}}_n) + \\ + \left\|\hat{\mathbf{r}}_{n+1} - \beta^{-1}\mathbf{K}_{n+1}\boldsymbol{\lambda}_{n+1} - \mathbf{K}_{n+1}\mathbf{B}_{n+1}^{-1}(\gamma)\mathbf{X}_{n+1}\hat{\mathbf{w}}_{n+1}\right\|^2. \tag{22}$$

where $\delta > 0$ – regularization parameter.

Such a choice of the regulariser restricts the rate of change of estimates $\hat{\boldsymbol{\lambda}}_n$, which ensures effective smoothing of conjugate variables estimates.

Condition of optimality for the problem of functional (22) minimization, leads to normal matrix equations

$$\left(\beta^{-2}\mathbf{K}_{n+1}^{\mathrm{T}}\mathbf{K}_{n+1} + \delta \cdot \mathbf{K}_{n+1}\right)\boldsymbol{\lambda}_{n+1} = \delta\mathbf{K}_{n+1}\hat{\boldsymbol{\lambda}}_n + \\ + \beta^{-1}\mathbf{K}_{n+1}^{\mathrm{T}}[\hat{\mathbf{r}}_{n+1} - \mathbf{K}_{n+1}\mathbf{B}_{n+1}^{-1}(\gamma)\hat{\mathbf{X}}_{n+1}\hat{\mathbf{w}}_{n+1}]. \tag{23}$$

From (13), (21), (23) we obtain recurrent estimation algorithm for conjugate variables

$$\hat{\lambda}_{n+1} = \delta \cdot \mathbf{A}_{n+1}^{-1}(\rho) \cdot \mathbf{K}_{n+1} \cdot \hat{\lambda}_n + \\ +\beta \cdot \mathbf{A}_{n+1}^{-1}(\rho) \cdot (\hat{\mathbf{r}}_{n+1} - \beta \cdot \mathbf{K}_{n+1}\mathbf{B}_{n+1}^{-1}(\gamma)\mathbf{X}_{n+1}\hat{w}_{n+1}), \quad (24)$$

where $\rho = \beta^2 / \delta$.

Finally, the utility function model estimate based on the recursively estimated conjugate variables can be represented as

$$\hat{f}_{n+1}(x) = \chi_{n+1}^{\mathrm{T}}(x) \cdot [\rho^{-1}\mathbf{A}_{n+1}^{-1}(\rho)\mathbf{K}_{n+1}\hat{\lambda}_n + \\ + \mathbf{A}_{n+1}^{-1}(\rho)\hat{\mathbf{r}}_{n+1} - \\ -(\mathbf{I}_M - \mathbf{A}_{n+1}^{-1}(\rho)\mathbf{K}_{n+1})\mathbf{B}_{n+1}^{-1}(\gamma)\hat{\mathbf{X}}_{n+1}\hat{\mathbf{w}}_{n+1}]. \quad (25)$$

The convergence of recurrent algorithm for estimating conjugate variables (26) can be provided by an appropriate choice of the regularization parameter δ .

Thus, recurrent algorithms (19) and (26) define computational procedures for identifying kernel models of utility function on clusters in feature space that can be used to find estimates of the ranks of new items with similar features.

VI. Conclusion

The proposed approach to the problem of online learning to rank using training data stream is based on combining of dynamic items clustering in feature space and recurrent utility function estimating on clusters. The obtained model of learning ranking function is a linear combination of kernel functions with recurrently tuning parameters, at that the complexity of utility function model is determined by the number of clusters.

Implementation of the proposed method of online ranking learning on clusters first of all involves the improvement of algorithms of both cluster parameter and average clusters rank estimating. For this purpose it seems expedient to use semi-supervised clustering methods with partial use of data describes relative ranks. Further development of the proposed approach can be carried out in the direction of optimizing the number of clusters in feature space using the complexity ratings of learning ranking model.

References

[1] . Figueira, S. Greco and M. Ehrgott, Multiple criteria decision analysis: state of the art surveys. Springer, Switzerland, 2005.

[2] H. Li, Learning to rank for information retrieval and natural language processing. Morgan & Claypool, Toronto, 2011.

[3] T. Y. Liu, "Learning to rank for information retrieval," Foundations and trends in information retrieval, vol. 3, no. 3, pp.225–331, 2009.

[4] L. Hang, "A Short Introduction to Learning to Rank," IEICE Trans. Information & Systrems, vol. E94-D, no.10, October 2011.

[5] J. Furnkranz and E. Hullermeier, "Pairwise preference learning and ranking," Lecture Notes in Computer Science, vol. 2837, pp. 145-156, 2003.

[6] V. Noghin, "Relative importance of criteria: a quantitative approach," Journal Multi-Criteria Decision Analysis, v. 6, pp. 355-363, 1997.

[7] J. Barzilai, "Measurement and preference function modeling", Intern. Trans. in Operational Research, vol. 12, pp. 173-183, 2005.

[8] V. Podvezko, and A. Podviezko, "Dependence of multicriteria evaluation result on choice of preference functions and their parameters," Technological and Economic Development of Economy. Baltic Jornal of Sustainability,16(1), pp.143–158, 2010.

[9] L. Lyubchyk and G. Grinberg, "Preference function reconstruction for multiple criteria decision making dased on machine learning Approach," Recent developments and new directions in Soft Computing, L.A. Zadeh et al. (Eds), Springer, pp. 53-63, 2014.

[10] M. Espinoza, J. Suykens, and B. Moor, "Kernel based partially linear models and nonlinear identification," IEEE Transactions on Automatic Control, 2005, vol. 50 (10), pp. 1602–1606.

[11] J. A. Papini, S. de Amo, A. Kardec and S. Soares, "FPS Mining: A fast algorithm for mining user preferences in data streams," Journal of Information and Data Management, vol. 5, no. 1, pp. 4–15, February 2014.

[12] K. Hofmann, S. Whiteson, and M. de Rijke, "Balancing exploration and exploitation in learning to rank online", In ECIR 2011: 33-rd European Conference on Information Retrieval. Springer, pp, 251-263, April 2011.

[13] K. Hofmann, A. Schuth, S. Whiteson, and M. de Rijke, "Reusing historical interaction data for faster online learning to rank for information retrieval", In WSDM 2013: International Conference on Web Search and Data Mining, ACM, 2013.

[14] L. Lyubchyk and G. Grinberg, "Real time recursive preference learning to rank from data stream," 1-th IEEE International Conference on Data Stream Mining & Processing, Lviv, Ukraine, pp. 280=285, 2016.

[15] V. Strijov, and V. Shakin, "Index construction: the expert-statistical method," Environmental research, engineering and management, no. 4 (26), pp.51-55, 2003.

[16] L. Lyubchyk, V. Kolbasin and R. Shafeev, "Nonlinear signal reconstruction based on recursive moving window kernel method," 8-th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS'2015), Warsaw, Poland, pp. 298-302, 2015.