Large Scale Co-Regularized Ranking

Evgeni Tsivtsivadze[1] and **Katja Hofmann**[2] and **Tom Heskes**[3]

Abstract. As unlabeled data is usually easy to collect, semi-supervised learning algorithms that can be trained on large amounts of unlabeled and labeled data are becoming increasingly popular for ranking and preference learning problems [6, 23, 8, 21]. However, the computational complexity of the vast majority of these (pairwise) ranking and preference learning methods is super-linear, as optimizing an objective function over all possible pairs of data points is computationally expensive.

This paper builds upon [16] and proposes a novel large scale co-regularized algorithm that can take unlabeled data into account. This algorithm is suitable for learning to rank when large amounts of labeled and unlabeled data are available for training. Most importantly, the complexity of our algorithm does not depend on the size of the dataset. We evaluate the proposed algorithm using several publicly available datasets from the information retrieval (IR) domain, and show that it improves performance over supervised methods. Finally, we discuss possible implications of our algorithm for learning with implicit feedback in an online setting.

1 Introduction and background

Our paper proposes an algorithm that is applicable to large scale learning to rank. Unlike existing approaches the proposed algorithm can take into account unlabeled data, leading to improved ranking performance. Learning to rank algorithms have been successfully applied to various domains such as IR [12], bioinformatics [13], and automated reasoning [11]. One of the bottlenecks associated with ranking tasks is the quadratic dependence on the size of the dataset. That is, most of the (pairwise) methods suffer from the computational burden of optimizing an objective defined over $O(m^2)$ possible pairs for data points, where m is the size of the dataset. Usually, the complexity of ranking algorithms has super-linear dependency on m, except the work of [16] where the use of stochastic gradient descent on pairs results in an extremely efficient training procedure with strong generalization performance. Pairwise learning to rank with stochastic gradient descent results in a scalable methodology that we refer to as stochastic pairwise descent (SPD).

In [16] it is demonstrated that such stochastic gradient based learning methods provide state of the art results using a fraction of a second of CPU time for training. However, SPD is only applicable to supervised learning problems. A natural and useful extension of SPD is an extension to semi-supervised learning, where, in addition to labeled data, a large amount of unlabeled data is used for training. We address this problem and present a large scale co-regularized ranking algorithm for semi-supervised tasks.

Our large scale co-regularized ranking algorithm (LCRA) is formulated within a multi-view framework. In this framework the dataset attributes (i.e., features) are split into independent sets and an algorithm is trained based on these different "views". The goal of the learning process is to find a prediction function for every view performing well on the labeled data in such a way that all prediction functions agree on the unlabeled data. Closely related to this approach is the *co-regularization* framework described in [18], where the same idea of agreement maximization between the predictors is central. Recently it has been demonstrated that the co-regularization approach works well for various tasks e.g. domain adaptation [7], classification, regression [2], and clustering [3]. Moreover, theoretical investigations demonstrate that co-regularization reduces the Rademacher complexity by an amount that depends on the "distance" between the views [15, 19].

We think that our co-regularization algorithm is particularly promising in online learning to rank for IR settings, where a search engine learns from the limited feedback that can be inferred from direct interactions with a search engine users. In this paper, we first focus on co-regularization (in Section 2) and our co-regularized pairwise learning algorithm (3 and 4). Finally, we discuss possible extensions to the online learning to rank for IR setting (Section 5).

2 Large Scale Pairwise Learning to Rank and Co-regularization

Consider a training set $D = (X, Y, Q)$, where $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m)^T \in \mathbb{R}^m$ contains $n-$dimensional feature vectors of the data points, $Y = (y_1, \ldots, y_m)^T \in \mathbb{N}^m$ are the scores/ranks, and $Q = (q_1, \ldots, q_m)^T \in \mathbb{N}^m$ are the indices for identification to which group/query a particular data point belongs. Note that each entry $x_i \in X$ consists of features that encode the relation between a particular item (e.g., a document) to group or query q_i. In addition to the training set $D = (X, Y, Q)$ with *labeled* data we have a training set $\hat{D} = (\hat{X}, \hat{Q})$ with *unlabeled* data points. Also, let us consider M different hypotheses spaces $\mathcal{H}_1, \ldots, \mathcal{H}_M$ or so-called views. These views stem from different representations

[1] Institute for Computing and Information Sciences, Radboud University, The Netherlands & MSB Group, The Netherlands Organization for Applied Scientific Research, Zeist, The Netherlands. Email: evgeni@science.ru.nl

[2] Intelligent Systems Lab Amsterdam, University of Amsterdam, The Netherlands. Email: k.hofmann@uva.nl

[3] Institute for Computing and Information Sciences, Radboud University, The Netherlands. Email: t.heskes@science.ru.nl

of the data points, unique subsets of features. Finally, we define a set of candidate pairs P and $\hat{P}$ (implied by the datasets D and $\hat{D}$) as the set of all tuples $((\mathbf{a}, y_a, q_a), (\mathbf{b}, y_b, q_b))$ and $((\mathbf{a}, q_a), (\mathbf{b}, q_b))$, where $y_a \neq y_b$ and $q_a = q_b$.

Co-regularized algorithms are usually not straightforwardly applicable to large scale learning tasks, when large amounts of unlabeled as well as labeled data are available for training. Several recently proposed algorithms have complexity that is linear in the number of unlabeled data points and super-linear in the number of labeled examples (e.g. cubic as in case of co-regularized least squares [2, 23]). Such methods become infeasible to use as the dataset size increases. In particular, this applies to the pairwise learning setting (note that in the worst case $|P|$ grows quadratically with the dataset size).

2.1 Constructing the pairs

In the pairwise learning setting it is important to be able to sample from $|P|$ and $|\hat{P}|$ without explicitly constructing the datasets of pairs. Several approaches to address this problem are suggested in [16]. For simplicity we adopt one of the most basic techniques: we repeatedly select two examples $(\mathbf{a}, y_a, q_a)$ and $(\mathbf{b}, y_b, q_b)$ from the data until a pair is found such that $y_a \neq y_b$ and $q_a = q_b$. Then we construct a feature vector of the corresponding pair as $\mathbf{p} = \mathbf{a} - \mathbf{b}$ and the label $y = y_a - y_b$.

3 The Algorithm

Stochastic gradient based algorithms are amongst the most popular approaches for large scale learning. Methods such as Pegasos [17], LaSVM [1], GURLS [22] and many others have been successfully applied to large scale classification and regression problems, leading to state-of-the-art generalization performance. Recently, the SPD algorithm [16] has been successfully used to address large scale learning to rank tasks. The main idea is to sample candidate pairs from P for stochastic steps, without constructing P explicitly. This avoids dependence on $|P|$. In essence, the approach proposed in [16] reduces learning to rank to learning a binary classifier via stochastic gradient descent. This reduction preserves the convergence properties of stochastic gradient descent. Our algorithm is related to the above mentioned methods but is preferable in case unlabeled data points are available for learning.

Let us consider the large scale co-regularized ranking algorithm. We write the objective function as

$$J(W) = \sum_{v=1}^{M} \left(\sum_{i=1}^{|P|} \mathcal{L}(\mathbf{p}_i^v, y_i; \mathbf{w}^v) + \lambda \mathcal{L}_R(\mathbf{w}^v) \right) \qquad (1)$$
$$+\mu \sum_{\substack{v,u=1 \\ v \neq u}}^{M} \sum_{i=1}^{|\hat{P}|} \mathcal{L}_C(\mathbf{p}_i^v, \mathbf{p}_i^u; \mathbf{w}^v, \mathbf{w}^u),$$

where the first term corresponds to the loss function on the labeled pairs and the second term to a regularization on the individual prediction functions. The third is the co-regularization term that measures the disagreement between the different prediction functions on unlabeled pairs. Note that $W \in \mathbb{R}^{M \times n}$ is a matrix containing weight vectors for different views. Once the model is trained the final prediction can be obtained, for example, by averaging individual predictions for different views (as in [15]). We can approximate the optimal solution (obtained when minimizing (1)) by means of gradient descent

$$\mathbf{w}_{t+1}^v = \mathbf{w}_t^v - \eta_t^v \nabla_{\mathbf{w}^v} J(W). \qquad (2)$$

Let us consider the setting in which the squared loss function is used for co-regularization, and the L_2 norm is used for regularization. Choosing the squared loss for the co-regularization term is quite natural as it penalizes the differences among the prediction functions constructed for multiple views (similar to the standard regression setting where the differences between the predicted and true scores are penalized). For every iteration t of the algorithm, we first construct pairs via the procedure described in section 2.1 and denote the set of selected pairs by $A_t \subseteq P$ of size k. Similarly we choose $\hat{A}_t \subseteq \hat{P}$ of size l for each round t on the unlabeled dataset. Let us also denote by A_t^v the set A_t as seen in the view v. Then, we replace the "true" objective (1) with an approximate objective function and write the update rule as follows

$$\mathbf{w}_{t+1}^v = (1 - \eta_t^v \lambda)\mathbf{w}_t^v - \eta_t^v \sum_{(\mathbf{p}, y \in A_t^v)} \nabla \mathcal{L}(\mathbf{p}^v, y; \mathbf{w}_t^v) \;-$$
$$4\mu\eta_t^v \sum_{\substack{v,u=1 \\ v \neq u}}^{M} \sum_{(\mathbf{p}, y \in \hat{A}_t^v \cup \hat{A}_t^u)} \left(\mathbf{w}_t^{vT} \mathbf{p}^v - \mathbf{w}_t^{uT} \mathbf{p}^u \right) \mathbf{p}^v. \qquad (3)$$

Note that if we choose A_t to contain a single randomly selected pair, we recover a variant of the stochastic gradient method. In general, we allow A_t to be a set of k and $\hat{A}_t$ to be a set of l data points sampled i.i.d. from P and $\hat{P}$, respectively.

Recall that in the setting described above we are solving a learning to rank problem via reduction to classification of pairs of data points. For classification tasks, the hinge loss is usually considered as more appropriate, although in several studies it has been empirically demonstrated that the squared loss often leads to similar performance (see [14, 27]). The update rule using the hinge loss is derived as follows. Let us define A^{v+} to be the set of examples for which $\mathbf{w}^v$ obtains a non-zero loss, that is $A^{v+} = \{(\mathbf{p}^v, y) \in A_t^v : y\langle \mathbf{p}^v, \mathbf{w}^v \rangle < 1\}$. Then by substituting the second term in equation (3) with $\eta_t^v \sum_{(\mathbf{p}, y \in A^{v+})} y\mathbf{p}^v$ we obtain the update rule for the large scale co-regularized algorithm with hinge loss. When the squared loss function is used for labeled and unlabeled data we obtain the update rule by substituting the second term in equation (3) with $\eta_t^v \sum_{(\mathbf{p}, y \in A_t^v)} (y - \mathbf{w}^{vT}\mathbf{p}^v)\mathbf{p}^v$. Our large scale co-regularized ranking algorithm has complexity of $O(Md)$, where d is the number of nonzero elements in $\mathbf{p}^v$. The pseudocode is shown in Algorithm 1.

4 Experiments

The task of ranking query-document pairs is a problem central to document retrieval - given a query some of the available documents are more relevant in regards to it than some others. Because the user will usually be most interested in the top results returned, document retrieval systems are typically evaluated using performance measures such as mean average precision (MAP).

Algorithm 1 Large scale co-regularized ranking algorithm (LCRA-k-l)

```
Require: Datasets $D$ and $\hat{D}$, regularization parameter $\lambda$, batch sizes $k$ and $l$, number of iterations $N$, number of views $M$, co-regularization parameter $\mu$.
Ensure: $W = 0$
for $t = 1, 2, \ldots, N$ do
    Construct $A_t \subseteq P$ (using procedure from Sec 2.1), where $|A_t| = k$ and $\hat{A}_t \subseteq \hat{P}$, where $|\hat{A}_t| = l$
    for $v = 1, 2, \ldots, M$ do
        Set $A_t^{v+} = \{(\mathbf{p}^v, y) \in A_t^v : y\langle \mathbf{p}^v, \mathbf{w}_t^v \rangle < 1\}$
        Set $\eta_t^v = \frac{1}{\lambda t}$
        $\mathbf{w}_{t+1}^v \leftarrow (1 - \eta_t^v \lambda)\mathbf{w}_t^v - \eta_t^v \sum_{(\mathbf{p}, y \in A_t^{v+})} y\mathbf{p}^v - 4\mu\eta_t^v \sum_{\substack{v,u=1 \\ v \neq u}}^{M} \sum_{(\mathbf{p}, y \in \hat{A}_t^v \cup \hat{A}_t^u)} \left(\mathbf{w}_t^{vT}\mathbf{p}^v - \mathbf{w}_t^{uT}\mathbf{p}^u\right) \mathbf{p}^v$
Output $W$
```

To benchmark the performance of our algorithm we use Letor 3.0 (LEarning TO Rank) - a collection of several datasets extracted from corresponding IR data collections. The whole collection consists of a set of document-query pairs. Each document-query pair is represented as an example with a quite small number of highly abstract features. Our experiments are performed on each of the datasets separately. We preprocess the datasets by normalizing all feature values to values between 0 and 1 on a per query basis. We use the classical learning setting, where 70% of the data is used for training and the remaining 30% as testing. To simulate a semi-supervised learning setting, a subset of 20% of the training data is randomly selected to be used as labeled data. From the remaining training data, labels are removed.

We compare the performance of our large scale co-regularized ranking algorithm with several other methods, namely the baseline - supervised - version of the algorithm (without co-regularization), which is in equivalent to SPD [16] and to the pairwise Pegasos algorithm [17]. We also compare with the multi-view version of the algorithm, also excluding the co-regularization term, referred to as SPD MV. The comparison is made with several instantiations of the large scale co-regularized ranking algorithm, termed as LCRA-k-l, using various sizes of unlabeled batch examples. For the supervised learning algorithms, only the labeled part of the dataset is used for training. The same set is then used for training the co-regularized model, together with the unlabeled data.

Parameter selection for each model is done by 5-fold cross-validation over the training partition of the data. For the supervised models, parameters to be selected are learning rate η_0 and regularization parameter λ. For the supervised and semi-supervised multi-view models we consider two views that are constructed via random partitioning of the data attributes into two unique sets. Such division of the attributes for constructing multiple views has been previously used in [2]. For the multi-view model we have to estimate the learning rate η_0, as well as the λ_1 and λ_2 parameters. The semi-supervised model has an additional parameter μ controlling the influence of the co-regularization on model selection.

The results of our experiments are included in Table 1. It can be observed that in all experiments the proposed LCRA algorithm outperforms supervised learning methods. The obtained results are expected, as it has been previously demonstrated that co-regularization leads to improved classification performance. Note that our algorithm can be considered a pairwise classification approach for learning to rank.

Dataset	LCRA-1-5	LCRA-1-1	SPD	SPD MV
TD2003	0.15	0.11	0.10	0.11
TD2004	0.14	0.12	0.10	0.10
NP2003	0.54	0.54	0.47	0.49
NP2004	0.51	0.48	0.44	0.45
HP2003	0.60	0.57	0.50	0.52
HP2004	0.52	0.50	0.45	0.45
OHSUMED	0.33	0.31	0.29	0.30

Table 1. MAP-performance comparison of the LCRA algorithm and the baseline methods on the Letor dataset. Note that results of supervised learning algorithms are not comparable to previously reported benchmarks on Letor dataset due to the fact that they are trained only on 20% of the labeled data points.

5 Co-regularization in Online Learning to Rank for IR

In the previous sections we introduced a co-regularization algorithm for semi-supervised learning to rank. We think that this approach is particularly promising in the context of online learning to rank for IR, and discuss its application below.

In online learning for IR a search engine learns improved ranking functions by directly interacting with a user[4] [10, 25]. It is typically modeled as a contextual bandit problem[5] [20], where the query is the context provided by the user, and feedback can be inferred from user clicks on result documents that the search engine returned in response to the query.

5.1 Online learning for IR

The most important difference between the online learning to rank setting and the traditionally considered supervised learning to rank for IR setting is the feedback available to the learning algorithm. Like in other reinforcement learning (RL) settings [20], a retrieval system that learns from user interactions can only infer feedback about the documents or document rankings that it actually presents to the user (and that are inspected by the user). This results in much more limited information to learn from than is available in the supervised

[4] Here we use *online* in the RL sense, meaning that learning and application of the learned solution are performed at the same time. Note that this differs from the term's use in the optimization literature, where is usually refers to the scalability of algorithms.

[5] Contextual bandit problems are a type of reinforcement learning problem actions (i.e., presented documents) depend on the context (i.e., the query), but not on previous interactions between system and environment.

setting, where it is assumed that labels are exhaustive [24], or sampled in some systematic way [4].

In addition to the limited *amount* of feedback available in the online learning to rank for IR setting, the *quality* of the feedback is constrained as follows. Users of a retrieval system expect a ranked list of results, that is more or less ordered by the usefulness of these results given their information need (expressed as e.g., a text query). Consequently, they are most likely to inspect results presented at the top of the returned result list, and continue examining and/or interacting with documents at subsequently lower ranks until their information need is met, or until they run out of time, patience, or some other restricted resource. For the most effective learning outcomes, this means that the documents (or pairs of documents) on which feedback would be most useful for learning should be presented first (we call this strategy of presenting result documents *exploration*). However, the documents that are most useful for learning may not be the ones that are most likely to fulfill the users information need (*exploitation*). Consequently, an effective learning algorithm should balance exploration and exploitation to optimize online performance, i.e., performance while learning from user interactions.

5.2 Related work

Several recent approaches address the problem of learning to rank for IR in an online setting. Yue et al. formulate the Dueling Bandit problem [26] and the K-armed Bandit problem [25]. In both formulations, learning is based on observing pairwise feedback on complete rankings, obtained using so-called interleaved comparison methods, where user clicks are observed on specially constructed result lists that allow inferring a preference between the two rankings [5, 9]. The algorithms proposed to address these tasks follow an exploit-then-explore approach, where it is assumed an algorithm can learn quickly before starting to exploit what has been learned, and performance while learning is largely ignored.

Follow-up work suggests that online performance can be improved by balancing exploration and exploitation [10]. In this work, it was shown that different learning approaches are affected by a balance of exploration and exploitation in different ways. For the listwise Dueling Bandit approach, it was shown that the originally proposed, purely exploratory algorithm over-explored, and that effective learning could be achieved by injecting only two exploratory documents into an otherwise exploitative result list. For the alternative pairwise approach, that is the most similar to the SPD approach evaluated in this paper, it was found that a purely exploitative algorithm performed very well when feedback could be reliably inferred from user clicks. However, when user feedback was noisy, bias introduced by the preference of users for higher-ranked results caused learning outcomes to deteriorate dramatically. To combat this performance loss, exploration had to be increased (which, in its simplest form can consist of random exploration).

5.3 Relation to online co-regularization

The co-regularization algorithm presented in this paper is directly applicable to existing pairwise learning approaches for the online learning to rank for IR setting. Because labeled feedback is particularly limited in the start-up phase of an online learning task, high initial learning gains are expected in this setting when easily obtainable unlabeled data can be used to complement this data. The resulting higher-quality result lists are expected to result in more reliable feedback [10]. As a result, learning could be sped up while reducing the need for exploration, leading to increased online performance. An experimental investigation of this hypothesis will be conducted in follow-up work.

6 Conclusion

In this paper we have presented a large-scale co-regularized algorithm for ranking and preference learning. Our algorithm can use unlabeled data to improve learning when labeled data is scarce. Our experiments on 7 standard learning to rank data sets show that our co-regularization component consistently improves performance over algorithms without co-regularization. We think that this approach can be particularly beneficial in an online learning to rank setting, where algorithms learn directly from interacting with users and effective use of limited feedback is paramount. We conclude with a brief outlook on future work in this area.

Acknowledgments

This research was partially supported by the Netherlands Organisation for Scientific Research (NWO) under project nr 612.061.814 and a Vici grant 639.023.604.

References

[1] Leon Bottou, Antoine Bordes, and Seyda Ertekin. Lasvm, 2009. `http://mloss.org/software/view/23/`.

[2] Ulf Brefeld, Thomas Gärtner, Tobias Scheffer, and Stefan Wrobel, 'Efficient co-regularised least squares regression', in *ICML '06*, pp. 137–144. ACM, (2006).

[3] Ulf Brefeld and Tobias Scheffer, 'Co-em support vector learning', in *ICML'04*, p. 16, New York, NY, USA, (2004). ACM.

[4] Ben Carterette, James Allan, and Ramesh Sitaraman, 'Minimal test collections for retrieval evaluation', in *SIGIR '06*, pp. 268–275. ACM, (2006).

[5] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue, 'Large-scale validation and analysis of interleaved search evaluation', *ACM Trans. Inf. Syst.*, **30**(1), 6:1–6:41, (2012).

[6] W Chu and Z Ghahramani, 'Extensions of Gaussian processes for ranking: semi-supervised and active learning', in *NIPS Workshop on Learning to Rank*, pp. 29–34, (2005).

[7] Hal Daume, Abhishek Kumar, and Avishek Saha, 'Co-regularization based semi-supervised domain adaptation', in *NIPS '10*, eds., J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, 478–486, (2010).

[8] Kevin Duh and Katrin Kirchhoff, 'Semi-supervised ranking for document retrieval', *Comput. Speech Lang.*, **25**(2), 261–281, (2011).

[9] K. Hofmann, S. Whiteson, and M. de Rijke, 'A probabilistic method for inferring preferences from clicks', in *CIKM '11*, pp. 249–258, (2011).

[10] K. Hofmann, S. Whiteson, and M. de Rijke, 'Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval', *Information Retrieval Journal (to appear)*, (2012).

[11] Daniel Kühlwein, Josef Urban, Evgeni Tsivtsivadze, Herman Geuvers, and Tom Heskes, 'Multi-output ranking for automated reasoning', in *KDIR'11*, (2011).

[12] Tie-Yan Liu, 'Learning to rank for information retrieval', *Foundations and Trends in Information Retrieval*, **3**(3), 225–331, (2009).

[13] Iain Melvin, Jason Weston, Christina S. Leslie, and William Stafford Noble, 'Rankprop: a web server for protein remote homology detection', *Bioinformatics*, **25**(1), 121–122, (2009).

[14] Ryan Rifkin, Gene Yeo, and Tomaso Poggio, 'Regularized least-squares classification', in *Advances in Learning Theory: Methods, Model and Applications*, eds., J.A.K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, volume 190 of *NATO Science Series III: Computer and System Sciences*, chapter 7, 131–154, IOS Press, Amsterdam, (2003).

[15] David Rosenberg and Peter L. Bartlett, 'The Rademacher complexity of co-regularized kernel classes', in *AISTATS'07*, eds., Marina Meila and Xiaotong Shen, pp. 396–403, (2007).

[16] D. Sculley, 'Large Scale Learning to Rank', in *NIPS Workshop on Advances in Ranking*, (2009).

[17] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro, 'Pegasos: Primal Estimated sub-GrAdient SOlver for SVM', in *ICML '07*, pp. 807–814. ACM, (2007).

[18] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin, 'A co-regularization approach to semi-supervised learning with multiple views', in *ICML Workshop on Learning with Multiple Views*, (2005).

[19] Vikas Sindhwani and David Rosenberg, 'An RKHS for multi-view learning and manifold co-regularization', in *ICML'08*, eds., Andrew McCallum and Sam Roweis, pp. 976–983, Helsinki, Finland, (2008).

[20] Richard S. Sutton and Andrew G. Barto, *Reinforcement learning: An introduction*, MIT Press, Cambridge, MA, USA, 1998.

[21] Martin Szummer and Emine Yilmaz, 'Semi-supervised learning to rank with preference regularization', in *CIKM '11*, pp. 269–278. ACM, (2011).

[22] Andrea Tacchetti, Pavan Mallapragada, Matteo Santoro, and Lorenzo Rosasco, 'GURLS: a toolbox for large scale multiclass learning', in *NIPS 2011 workshop on parallel and large-scale machine learning*. `http://cbcl.mit.edu/gurls/`.

[23] Evgeni Tsivtsivadze, Tapio Pahikkala, Jorma Boberg, Tapio Salakoski, and Tom Heskes, 'Co-regularized least-squares for label ranking', in *Preference Learning*, eds., Eyke Hüllermeier and Johannes Fürnkranz, pp. 107–123, (2010).

[24] Ellen M. Voorhees and Donna K. Harman, *TREC: Experiment and Evaluation in Information Retrieval*, Digital Libraries and Electronic Publishing, MIT Press, 2005.

[25] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims, 'The k-armed dueling bandits problem', *Journal of Computer and System Sciences*, **78**(5), 1538 – 1556, (2012).

[26] Yisong Yue and Thorsten Joachims, 'Interactively optimizing information retrieval systems as a dueling bandits problem', in *ICML'09*, pp. 1201–1208, (2009).

[27] Peng Zhang and Jing Peng, 'Svm vs regularized least squares classification', in *Proceedings of the International Conference on Pattern Recognition*, ICPR '04, pp. 176–179, (2004).