Online Co-Regularized Algorithms

Tom de Ruijter[1], Evgeni Tsivtsivadze[1,2], and Tom Heskes[1]

[1] Institute for Computing and Information Sciences,
Radboud University, The Netherlands
firstname.lastname@science.ru.nl,
[2] MSB Group, The Netherlands Organization for Applied Scientific Research,
Zeist, The Netherlands
firstname.lastname@tno.nl

Abstract. We propose an online co-regularized learning algorithm for classification and regression tasks. We demonstrate that by sequentially co-regularizing prediction functions on unlabeled data points, our algorithm provides improved performance in comparison to supervised methods on several UCI benchmarks and a real world natural language processing dataset. The presented algorithm is particularly applicable to learning tasks where large amounts of (unlabeled) data are available for training. We also provide an easy to set-up and use Python implementation of our algorithm[3].

1 Introduction and background

Semi-supervised learning algorithms have gained more and more attention in recent years as it uses unlabeled data. This type of information is typically much easier to obtain than labeled data. *Multi-view* learning algorithms split the attributes into independent sets and an algorithm is learnt based on these different "views". The goal of the learning process consists in finding a prediction function for every view performing well on the labeled data and so that all prediction functions agree on the unlabeled data. Closely related to this approach is the *co-regularization* framework described in [1], where the same idea of agreement maximization between the predictors is central. Briefly stated, algorithms based upon this approach search for hypotheses from different views, such that the training error of each hypothesis on the labeled data is small and, at the same time, the hypotheses give similar predictions for the unlabeled data. Within this framework, the disagreement among the predictors is taken into account via a co-regularization term. Empirical results show that the co-regularization approach works well for domain adaptation [2], classification [1, 3], regression [4], and clustering [5] tasks. Moreover, theoretical investigations demonstrate that the co-regularization approach reduces the Rademacher complexity by an amount that depends on the "distance" between the views [6, 7].

[3] Available at www.learning-machines.com

1.1 Co-Regularization Framework

A classical example of co-regularization is a web-document classification task where the document can be represented by keyword features or as the link features it contains, thus, creating two distinct views of the same data point [8]. For each of these views prediction function can be trained and co-regularized on unlabeled data to increase generalization performance of the algorithm.

Many of the multi-view algorithms are formulated within a regularization framework [9, 10]. In this framework, the learning algorithm selects a hypothesis f which minimizes a cost function and which is, at the same time, not too "complex", i.e. which does not overfit while training and is therefore able to generalize to unseen data.

Consider a training set $S = (X, Y)$ originating from a set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ of data points where $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m)^T \in \mathcal{X}^m$ and $Y = (y_1, \ldots, y_m)^T \in \mathbb{R}^m$. Also, let us consider M different hypotheses spaces $\mathcal{H}_1, \ldots, \mathcal{H}_M$ or so-called views. These views stem from different representation of the data points, meaning unique subsets of features. Let us assume that in addition to the training set $S = (X, Y)$ *with* labeled examples we have a training set $\widetilde{S} = (\widetilde{X})$ with unlabeled data points $\{\mathbf{x}_{m+i}\}_{i=1}^n$, $\widetilde{X} = (\mathbf{x}_{m+1}, \ldots, \mathbf{x}_{m+n})^T \in \mathcal{X}^n$. In the co-regularization setting we would like to identify functions $\mathbf{f} = (f^1, \ldots, f^M) \in \mathcal{H}^1 \times \ldots \times \mathcal{H}^M$ minimizing the objective function

$$J(\mathbf{f}) = \sum_{v=1}^{M} \mathcal{L}(f^v, S) + \lambda \sum_{v=1}^{M} \|f^v\|_{\mathcal{H}^v}^2 + \mu \sum_{v,u=1}^{M} \mathcal{L}_C(f^v, f^u, \widetilde{S}), \tag{1}$$

where $\lambda, \mu \in \mathbb{R}^+$ are regularization parameters and where $\mathcal{L}_C$ is the loss function measuring the disagreement between the prediction functions of the views on the unlabeled data.

We note that by considering a single view and specializing the loss in the above formulation we can obtain a variety of other algorithms. We obtain support vector machines [11] by choosing a hinge loss function and we obtain regularized least-squares (RLS) [12] by choosing a squared loss function. In turn, the RLS algorithm with slight modifications - possibly including a bias term - leads to a wide class of other learners, such as the least-squares support vector machine [13], proximal vector machines [14] and kernel ridge regression [15].

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 the training. Several recently proposed algorithms have complexity that is linear in the number of unlabeled data points and superlinear in the number of labeled examples (e.g. cubic as in case of co-regularized least squares [4]). Such methods become impossible to use as the dataset size increases.

2 Online Co-Regularized algorithm

Online algorithms are amongst the most popular approaches for large scale learning. Methods such as PEGASOS [16], LASVM [17] and GURLS [18] have been successfully applied to a wide range of large scale problems leading to state-of-the-art generalization performance. Our algorithm is related to the above mentioned methods but is preferable in case unlabeled data points are available for learning.

A popular approach to tackle large scale learning problems is by using efficient approximation techniques such as stochastic gradient decent (see e.g. [19]). Let us consider the co-regularized algorithm in the online setting. Slightly overloading our notations, we write the objective function as

$$J^*(W) = \sum_{v=1}^{M} \left(\sum_{i=1}^{m} \mathcal{L}(\mathbf{x}_i^v, y_i; \mathbf{w}^v) + \lambda \mathcal{L}_R(\mathbf{w}^v) \right) + \mu \sum_{\substack{v,u=1 \\ v \neq u}}^{M} \sum_{i=m+1}^{m+n} \mathcal{L}_C(\mathbf{x}_i^v, \mathbf{x}_i^u; \mathbf{w}^v, \mathbf{w}^u), \tag{2}$$

where the first term corresponds to the loss function mentioned previously and the second term to a regularization on the individual prediction functions. The third is again a co-regularization term that measures the disagreement between the different prediction functions on unlabeled data. We can approximate the optimal solution (obtained when minimizing (2)) by means of gradient descent

$$\mathbf{w}_{t+1}^v = \mathbf{w}_t^v - \eta_t^v \nabla_{\mathbf{w}^v} J^*(W). \tag{3}$$

Let us consider the setting in which the squared loss function is used for the co-regularization and L_2 norm for the regularization terms. The choice of 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 choose a set $A_t \subseteq S$ of size k. Similarly we choose $\tilde{A}_t \subseteq \tilde{S}$ of size l for each round t on the unlabeled dataset. Then, we replace the "true" objective (2) with an approximate objective function and write the update rule as follows

$$\begin{aligned} \mathbf{w}_{t+1}^v = (1 - \eta_t^v \lambda)\mathbf{w}_t^v - \ & \eta_t^v \sum_{(\mathbf{x}, y \in A_t)} \nabla \mathcal{L}(\mathbf{x}^v, y; \mathbf{w}_t^v) \ - \\ & 4\mu\eta_t^v \sum_{\substack{v,u=1 \\ v \neq u}}^{M} \sum_{(\mathbf{x}, y \in \tilde{A}_t)} \left(\mathbf{w}_t^{vT} \mathbf{x}^v - \mathbf{w}_t^{uT} \mathbf{x}^u \right) \mathbf{x}^v. \end{aligned} \tag{4}$$

Note that if we choose $A_t = S$ and $\tilde{A}_t = \tilde{S}$ on each round t we obtain the gradient projection method. At the other extreme, if we choose A_t to contain a single randomly selected example, we recover a variant of the stochastic gradient

method. In general, we allow A_t to be a set of k and $\tilde{A}_t$ to be a set of l data points sampled i.i.d. from S and $\tilde{S}$, respectively.

The hinge loss function is usually considered as more appropriate for classification problems, although in several studies it has been empirically demonstrated that squared loss often leads to similar performance (see [20, 21]). Let us define A^+ to be the set of examples for which $\mathbf{w}^v$ obtains a non-zero loss, that is $A^+ = \{(\mathbf{x}^v, y) \in A_t : y\langle \mathbf{x}^v, \mathbf{w}^v \rangle < 1\}$. Then by substituting the second term in the equation (4) with $\eta_t^v \sum_{(\mathbf{x}, y \in A^+)} y\mathbf{x}^v$ we obtain the update rule for the online 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 (4) with $\eta_t^v \sum_{(\mathbf{x}, y \in A_t)} (y - \mathbf{w}^{vT}\mathbf{x}^v)\mathbf{x}^v$. Finally, if the number of dimensions in the dataset is not large we can use all unlabeled data points at every iteration by precomputing multiplication terms in $4\mu\eta_t^v \sum_{\substack{v,u=1 \\ v \neq u}}^{M} (X^{vT}\mathbf{w}^v - X^{uT}\mathbf{w}^u)X^v$. Below we provide a description of the proposed online co-regularized algorithm for classification task.

Algorithm Online co-regularized algorithm (OCA-k-l)

Require: Datasets S and $\tilde{S}$, regularization parameter λ, batch sizes k and l, number of views M, number of iterations N, co-regularization parameter μ.

Ensure: $\mathbf{w}^v = 0$

1: **for** $t = 1, 2, \ldots, N$ **do**

2: Choose $A_t \subseteq S$, where $|A_t| = k$ and $\tilde{A}_t \subseteq \tilde{S}$, where $|\tilde{A}_t| = l$

3: Set $A_t^+ = \{(\mathbf{x}^v, y) \in A_t : y\langle \mathbf{x}^v, \mathbf{w}_t^v \rangle < 1\}$

4: Set $\eta_t^v = \frac{1}{\lambda t}$

5: $\mathbf{w}_{t+1}^v \leftarrow (1 - \eta_t^v \lambda)\mathbf{w}_t^v - \eta_t^v \sum_{(\mathbf{x}, y \in A_t^+)} y\mathbf{x}^v$

6: $\qquad -4\mu\eta_t^v \sum_{\substack{v,u=1 \\ v \neq u}}^{M} \sum_{(\mathbf{x}, y \in \tilde{A}_t)} \left(\mathbf{w}_t^{vT}\mathbf{x}^v - \mathbf{w}_t^{uT}\mathbf{x}^u\right)\mathbf{x}^v$

7: Output $\mathbf{w}_{N+1}^v$ (weight vector for a single view)

2.1 Discussion

Although the proposed algorithm is presented with hinge loss function, the extensions to logarithmic, ϵ-intensive, and several others are relatively straightforward. Moreover, similarly to the PEGASOS algorithm, OCA can be also formulated to use kernel functions. The benefit in this case is that no direct access to the feature vectors of $\mathbf{x}^v$ is needed and we can also consider non-linear kernel functions for the learning task. However, the drawback of such a "kernelized" version of OCA is that although the number of iterations required by the algorithm does not depend on the number of training examples, the runtime does. Also note that in the above formulation we considered a version of the algorithm that

makes use of randomly sampled subsets A_t and $\tilde{A}_t$ at every iteration. Flexibility to vary the sizes of these sets at every time step can be beneficial in some circumstances, for example when prediction functions in multiple views start to diverge significantly one can consider increasing the number of unlabeled data points in the co-regularized term.

In our empirical evaluation we test the performance of the algorithm on various datasets, including one from the natural language processing domain where it is common to have very sparse and high dimensional feature representations of the data. To deal with such a scenario we follow the suggestion presented in [16]. That is, when each data point has very few non-zero elements we can represent a weight vector $\mathbf{w}^v$ as a pair $(\mathbf{z}, a)$ where $\mathbf{z} \in \mathbb{R}^m$ is a vector and a is a scalar. The vector $\mathbf{w}^v$ is defined as $\mathbf{w}^v = a\mathbf{z}$. Using this representation, it can be verified that the total number of operations required for performing one iteration of our online co-regularization algorithm is $O(Md)$, where d is the number of non-zero elements in $\mathbf{x}^v$.

3 Experiments

We evaluate the performance of the proposed algorithm on publicly available datasets from the UCI repository[4] [5] and the BioInfer corpus[6] - a real world natural language processing dataset. To benchmark the performance of our algorithm we select a number of standard regression and classification datasets from the repository, namely ABALONE, CADATA, HOUSING, MG, SPACE, SVMGUIDE3, GERMANNUMER, AUSTRALIAN and use the BioInfer corpus to evaluate performance of our method on complex natural language processing data. To simulate a semi-supervised learning setting, we remove part of the labels from each of the datasets. We use the classical learning setting, where 70% of the data is used for training and the remaining 30% as testing. 20% of the training data is randomly selected to be labeled, and the others are used as unlabeled data. Note that the used datasets vary in size from several hundred samples to several tens of thousands and the density varies from sparse to dense. Depending on the learning task, the performance measure is either AUC for classification or RMSE for regression. The datasets are preprocessed by applying a linear scaling to each feature to the interval $[-1, 1]$. For regression datasets we also apply a linear scaling on the labels, to the interval $[0, 100]$.

We compare the performance of our online co-regularized algorithm with several other methods, namely the baseline - supervised - version of the algorithm, excluding the co-regularization term, which is in essence equivalent to the PEGASOS algorithm [16]. We also compare with the multi-view version of the algorithm,

[4] `http://archive.ics.uci.edu/ml/`
[5] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`
[6] Available at `www.it.utu.fi/BioInfer`

also excluding the co-regularization term, termed as PEGASOS MV. For classification sets we use a hinge loss for all methods, denoted by appending HL. For regression sets we do the same with the squared loss, SL. We compare with several instantiations of the online co-regularized algorithm, termed as OCA-k-l, using various sizes of unlabeled batch examples. For the supervised learning algorithms, only the labeled part of 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 10-fold cross-validation over the train 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 [4]. 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.

Table 1. Results on the Abalone dataset. The OCA-1-5 algorithm outperforms supervised learning methods. Improvement in performance is statistically significant according to the Wilcoxon signed rank test. The difference between the co-regularized algorithms OCA-1-1 and OCA-1-5 is also statistically significant.

Abalone	CV PERF (RMSE)	η_0	λ_1	λ_2	μ	TEST PERF (RMSE)
PEGASOS SL	14.40	0.5	2.0	N/A	N/A	19.46
PEGASOS MV SL	11.70	0.5	16.0	0.25	N/A	15.52
OCA-1-1	11.63	0.25	16.0	0.25	0.5	14.23
OCA-1-5	11.73	0.25	16	0.25	0.5	**13.50**

Table 2. Results on the Cadata dataset. OCA-1-1 leads to a statistically significant performance improvement compared to supervised Pegasos SL and Pegasos MV SL according to the Wilcoxon signed rank test. The difference between the co-regularized algorithms OCA-1-1 and OCA-1-5 is not statistically significant.

Cadata	CV PERF (RMSE)	η_0	λ_1	λ_2	μ	TEST PERF (RMSE)
PEGASOS SL	24.45	2.5	8.0	N/A	N/A	26.00
PEGASOS MV SL	24.29	1.5	16.0	4.0	N/A	27.26
OCA-1-1	23.97	1.5	1.0	16.0	1.5	**25.76**
OCA-1-5	23.30	1.5	1.0	16.0	1.5	25.80

The results of the experiments are included in the Tables 1-8. It can be observed that in all experiments except the housing dataset, the proposed co-regularized

Table 3. Results on the Housing dataset. Pegasos SL outperforms other methods on the smallest dataset used in our empirical evaluations.

Housing	CV perf (RMSE)	η_0	λ_1	λ_2	μ	Test perf (RMSE)
Pegasos SL	19.34	0.0625	8.0	n/a	n/a	**16.34**
Pegasos MV SL	17.07	0.01	16.0	64.0	n/a	17.59
OCA-1-1	17.75	0.01	4.0	256	1.5	18.54
OCA-1-5	16.13	0.01	4.0	256	1.5	18.69

Table 4. Results on the MG dataset. OCA-1-1 leads to statistically significant performance improvement compared to supervised Pegasos SL according to the Wilcoxon signed rank test. The differences between the co-regularized algorithms OCA-1-1, OCA-1-5, and Pegasos MV SL are not statistically significant.

MG	CV perf (RMSE)	η_0	λ_1	λ_2	μ	Test perf(RMSE)
Pegasos SL	45.53	0.125	1.0	n/a	n/a	46.71
Pegasos MV SL	45.88	0.125	0.5	0.125	n/a	45.73
OCA-1-1	44.51	1.5	64	32	0.1	**45.57**
OCA-1-5	44.93	0.125	0.5	0.125	0.01	45.91

Table 5. Results on the Space dataset. OCA-1-1 leads to statistically significant performance improvement compared to supervised Pegasos SL and Pegasos MV SL according to the Wilcoxon signed rank test. The difference between the co-regularized algorithms OCA-1-1 and OCA-1-5 is not statistically significant.

Space	CV perf (RMSE)	η_0	λ_1	λ_2	μ	Test perf (RMSE)
Pegasos SL	58.32	0.25	0.5	n/a	n/a	58.17
Pegasos MV SL	50.42	0.125	1.0	0.125	n/a	51.90
OCA-1-1	41.95	1.0	0.125	0.5	n/a	**36.60**
OCA-1-5	42.80	0.125	1.0	0.125	0.5	36.84

Table 6. Results on the Germannumer dataset. OCA-1-1 leads to statistically significant performance improvement compared to supervised Pegasos HL and Pegasos MV HL according to the Wilcoxon signed rank test. The difference between the co-regularized algorithms OCA-1-1 and OCA-1-5 is also statistically significant.

Germannumer	CV perf (AUC)	η_0	λ_1	λ_2	μ	Test perf (AUC)
Pegasos HL	0.72	10	0.03125	n/a	n/a	0.74
Pegasos MV HL	0.76	0.75	4	0.125	n/a	0.71
OCA-1-1	0.75	0.125	0.125	0.125	0.01	**0.75**
OCA-1-5	0.75	0.125	16	0.5	0.2	0.74

Table 7. Results on the Svmguide3 dataset. OCA-1-5 leads to statistically significant performance improvement compared to supervised Pegasos HL and Pegasos MV HL according to the Wilcoxon signed rank test. The difference between the co-regularized algorithms OCA-1-1 and OCA-1-5 is also statistically significant.

Svmguide3	CV PERF (AUC)	η_0	λ_1	λ_2	μ	TEST PERF (AUC)
PEGASOS HL	0.85	1	0.25	N/A	N/A	0.74
PEGASOS MV HL	0.83	1.5	0.125	0.125	N/A	0.75
OCA-1-1	0.77	1.5	0.125	0.125	0.05	0.73
OCA-1-5	0.82	0.25	0.0625	0.125	0.01	**0.76**

Table 8. Results on the Australian dataset. OCA leads to statistically significant performance improvement compared to supervised Pegasos HL and Pegasos MV HL according to the Wilcoxon signed rank test. The difference between the co-regularized algorithms OCA-1-1 and OCA-1-5 is not statistically significant.

Australian	CV PERF (AUC)	η_0	λ_1	λ_2	μ	TEST PERF (AUC)
PEGASOS HL	0.95	0.0625	0.03125	N/A	N/A	0.92
PEGASOS MV HL	0.95	0.25	0.0625	0.03125	N/A	0.92
OCA-1-1	0.93	0.5	0.0625	0.03125	0.001	**0.93**
OCA-1-5	0.94	1	0.0625	0.03125	0.001	**0.93**

algorithm outperforms supervised learning methods. The housing dataset is also the smallest dataset considered in our empirical evaluation. We use a Wilcoxon signed-rank test [22] to estimate whether the differences in performance are statistically significant. In all cases (with the exception of housing dataset) the OCA leads to statistically significant improvement over the standard PEGASOS algorithm. Detailed information for each dataset is reported in the caption of the corresponding table.

3.1 Parse goodness estimation

Throughout this experiment, we use the BioInfer corpus [23] which consists of 1100 manually annotated sentences.[7] For each sentence, we generate a set of candidate parses with a link grammar (LG) parser [24]. The LG parser is a full dependency parser based on a broad-coverage hand-written grammar. It generates all parses allowed by its grammar and applies a set of built-in heuristics to predict goodness of the parses. However, the performance of its heuristics has been found to be poor when applied to biomedical text [25], and hence subsequent selection methods are needed. In our experiment we use proposed online co-regularized algorithm instead of LG parser built-in heuristics to predict goodness of the generated parse.

Our dataset consists of 3000 parses represented as sparse vectors of dimensionality 201740. We obtain a scoring for an input by comparing its parse to the

[7] Available at `www.it.utu.fi/BioInfer`

hand annotated correct parse of its sentence. In order to select the parameter values, we divide the dataset into training 70% and test set 30% (we ensure that the parses that belong to the same sentence belong to a single set). Also, 20% of the training data are randomly selected to be labeled, and the rest is used as unlabeled data.

The first dataset is used for parameter estimation and the second one is reserved for the final validation. The appropriate values of the regularization parameters are determined by grid search with 10-fold cross-validation on the parameter estimation data. Finally, the algorithm is trained on the whole training set with the selected parameter values and tested with the test parses reserved for the final validation. The results of the experiment are presented in Table 9. It can be observed that OCA notably outperforms both supervised methods and the improvement in performance is statistically significant according to a Wilcoxon signed rank test. Those results indicate that our algorithm is applicable to the tasks in natural language processing and other domains where sparse, high dimensional data are commonplace.

Table 9. Results on the BioInfer dataset. OCA-1-5 leads to statistically significant performance improvement compared to supervised Pegasos SL and Pegasos MV SL according to Wilcoxon signed rank test. The difference between the co-regularized algorithms OCA-1-1 and OCA-1-5 is not statistically significant.

BioInfer	CV perf (RMSE)	η_0	λ_1	λ_2	μ	Test perf (RMSE)
Pegasos SL	45.36	0.5	32	n/a	n/a	63.86
Pegasos MV SL	44.94	0.5	16	16	n/a	63.16
OCA-1-1	39.78	0.5	16	16	0.4	61.47
OCA-1-5	39.85	0.5	16	16	0.4	**61.29**

4 Conclusions

This work presents an online co-regularized algorithm for regression and classification tasks. Our algorithm is computationally efficient and is naturally suited for learning tasks in which large amounts of unlabeled and labeled data are available for training. Our algorithm is related to online methods such as such as Pegasos [16], LaSVM [17] and GURLS [18] and unlike many co-regularized algorithms has computational complexity independent of the number of training data points. In the empirical evaluation we demonstrate that our method consistently performs well on publicly available datasets as well as notably outperforms supervised learning algorithms on the BioInfer corpus from the natural language processing domain. Last but not least, we make available an efficient implementation of our algorithm coded in Python.

Our algorithm can be extended to be applicable to various learning tasks. For instance, it can be adapted for the task of large scale preference learning and ranking. Large scale learning to rank has recently received notable attention and while several supervised learning algorithms have been proposed [26], taking into account large amount of unlabeled data (naturally abundant in IR domain) can help to even further improve predictive performance of the models. Thus, an interesting future research direction is to adapt and apply the online co-regularized algorithm to large scale learning to rank tasks.

References

1. Sindhwani, V., Niyogi, P., Belkin, M.: A co-regularization approach to semi-supervised learning with multiple views. In: Proceedings of ICML Workshop on Learning with Multiple Views. (2005)
2. Daume, H., Kumar, A., Saha, A.: Co-regularization based semi-supervised domain adaptation. In Lafferty, J., Williams, C.K.I., Shawe-Taylor, J., Zemel, R., Culotta, A., eds.: Advances in Neural Information Processing Systems 23. (2010) 478–486
3. Goldberg, A.B., Li, M., Zhu, X.: Online manifold regularization: A new learning setting and empirical study. In: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases, Springer (2008) 393–407
4. Brefeld, U., Gärtner, T., Scheffer, T., Wrobel, S.: Efficient co-regularised least squares regression. In: Proceedings of the International Conference on Machine learning, New York, NY, USA, ACM (2006) 137–144
5. Brefeld, U., Scheffer, T.: Co-em support vector learning. In: Proceedings of the 21st International Conference on Machine learning, New York, NY, USA, ACM (2004) 16
6. Rosenberg, D., Bartlett, P.L.: The Rademacher complexity of co-regularized kernel classes. In Meila, M., Shen, X., eds.: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics. (2007) 396–403
7. Sindhwani, V., Rosenberg, D.: An RKHS for multi-view learning and manifold co-regularization. In McCallum, A., Roweis, S., eds.: Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008), Helsinki, Finland, Omnipress (2008) 976–983
8. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory, New York, NY, USA, ACM (1998) 92–100
9. Scholkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA (2001)
10. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York, NY, USA (2004)
11. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
12. Rifkin, R., Yeo, G., Poggio, T.: Regularized least-squares classification. In: Advances in Learning Theory: Methods, Model and Applications, Amsterdam, IOS Press (2003) 131–154
13. Suykens, J.A.K., Vandewalle, J.: Least squares support vector machine classifiers. Neural Processing Letters **9**(3) (1999) 293–300

14. Fung, G., Mangasarian, O.L.: Proximal support vector machine classifiers. In: The seventh ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2001) 77–86
15. Saunders, C., Gammerman, A., Vovk, V.: Ridge regression learning algorithm in dual variables. In: Fifteenth International Conference on Machine Learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1998) 515–521
16. Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In: Proceedings of the 24th international conference on Machine learning, ACM (2007) 807–814
17. Bottou, L., Bordes, A., Ertekin, S.: Lasvm (2009) `http://mloss.org/software/view/23/`.
18. Tacchetti, A., Mallapragada, P., Santoro, M., Rosasco, L.: GURLS: a toolbox for large scale multiclass learning. In: NIPS 2011 workshop on parallel and large-scale machine learning. `http://cbcl.mit.edu/gurls/`.
19. Yuan, G.x., Ho, C.h., Lin, C.j.: Recent advances of large-scale linear classification. Proceedings of the IEEE (3) (2011) 1–15
20. Rifkin, R., Yeo, G., Poggio, T.: Regularized least-squares classification. In Suykens, J., Horvath, G., Basu, S., Micchelli, C., Vandewalle, J., eds.: Advances in Learning Theory: Methods, Model and Applications. Volume 190 of NATO Science Series III: Computer and System Sciences. IOS Press, Amsterdam (2003) 131–154
21. Zhang, P., Peng, J.: Svm vs regularized least squares classification. In: Proceedings of the International Conference on Pattern Recognition. ICPR '04 (2004) 176–179
22. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research **7** (2006) 1–30
23. Pyysalo, S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., Salakoski, T.: BioInfer: A corpus for information extraction in the biomedical domain. BMC Bioinformatics **8** (2007) 50
24. Sleator, D.D., Temperley, D.: Parsing english with a link grammar. Technical Report CMU-CS-91-196, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA (October 1991)
25. Pyysalo, S., Ginter, F., Pahikkala, T., Boberg, J., Järvinen, J., Salakoski, T.: Evaluation of two dependency parsers on biomedical corpus targeted at protein-protein interactions. Recent Advances in Natural Language Processing for Biomedical Applications, special issue of the International Journal of Medical Informatics **75**(6) (2006) 430–442
26. Sculley, D.: Large Scale Learning to Rank. In: NIPS 2009 Workshop on Advances in Ranking. (2009) 1–6