Improving Classification with Pairwise Constraints: A Margin-based Approach

Nam Nguyen and Rich Caruana

Department of Computer Science
Cornell University, USA
{nhnguyen,caruana}@cs.cornell.edu

Abstract. In this paper, we address the semi-supervised learning problem when there is a small amount of labeled data augmented with pairwise constraints indicating whether a pair of examples belongs to a same class or different classes. We introduce a discriminative learning approach that incorporates pairwise constraints into the conventional margin-based learning framework. We also present an efficient algorithm, PCSVM, to solve the pairwise constraint learning problem. Experiments with 15 data sets show that pairwise constraint information significantly increases the performance of classification.

Key words: classification, pairwise constraints, margin-based learning

1 Introduction

Learning with partially labeled training data, also known as semi-supervised learning, has received considerable attention, especially for classification and clustering [1–17]. While labeled data is usually expensive, time consuming to collect, and sometimes requires human domain experts to annotate, unlabeled data often is relatively easy to obtain. For this reason, semi-supervised learning has mainly focused on using the large amount of unlabeled data [18], together with a small amount of labeled data, to learn better classifiers. Note that unlabeled data may not always help. For example, [19] showed that unlabeled data can degrade classification performance even in situations where additional labeled data would increase the performance. Hence, partially labeled data is an attractive tradeoff between fully labeled data and unlabeled data.

In this paper, we investigate the usefulness of partially labeled information in the form of pairwise constraints. More specifically, a pairwise constraint between two items indicates whether they belong to the same class or not. Similar to unlabeled data, in many applications pairwise constraints can be collected automatically, e.g. in [1], pairwise constraints are extracted from surveillance video. Pairwise constraints also can be relatively easy to collect from human feedback: unlike labels that would require users to have prior knowledge or experience with a data set, pairwise constraints require often little effort from users. For example, in face recognition, it is far easier for users to determine if two faces

belong to the same nationality, than it would be for the same users to classify the faces into different nationalities.

In this paper, we propose a discriminative learning approach which incorporates pairwise constraints into the conventional margin-based learning framework. In extensive experiments with a variety of data sets, pairwise constraints significantly increase the performance of classification. The paper is structured as follow: in section 2, we describe in detail our classification algorithm, PCSVM, which incorporates pairwise constraints; in section 3, we review related work on semi-supervised learning with pairwise constraints; the experimental results and conclusion are given in section 4 and 5, respectively.

2 Classification with Pairwise Constraints

In the supervised setting, a learning algorithm typically takes a set of labeled training examples, $\mathbf{L} = \{(x_i, y_i)\}_{i=1}^{n}$ as input, where $x_i \in \mathcal{X}$ and y_i belongs to a finite set of classes called $\mathcal{Y}$. For our learning framework, in addition to the labeled data, there is additional partially labeled data in the form of pairwise constraints $\mathbf{C} = \{(x_i^{\alpha}, x_i^{\beta}, \widetilde{y}_i)\}_{i=1}^{m}$ where $x_i^{\alpha}, x_i^{\beta} \in \mathcal{X}$ and $\widetilde{y}_i \in \{+1, -1\}$ is the indicator of whether x_i^{α} and x_i^{β} belong to the same class ($\widetilde{y}_i = +1$), or not ($\widetilde{y}_i = -1$). Ultimately, the goal of classification is to form a hypothesis $h : \mathcal{X} \mapsto \mathcal{Y}$.

First, we review the margin-based multiclass classification, also known as the multiclass-SVM proposed by [20]. Consider a mapping $\Phi : \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{F}$ which projects each item-label pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ to $\Phi(x, y)$ in a new space $\mathcal{F}$,

$$\Phi(x, y) = \begin{bmatrix} x \cdot \mathcal{I}(y = 1) \\ \cdots \\ x \cdot \mathcal{I}(y = |\mathcal{Y}|) \end{bmatrix},$$

where $\mathcal{I}(\cdot)$ is the indicator function. The multiclass-SVM learns a weight vector w and slack variables ξ via the following quadratic optimization problem:

Optimization Problem I: Multiclass-SVM

$$\min_{w, \xi \geq 0} : \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \xi_i \tag{1}$$

subject to:

$$\forall (x_i, y_i) \in \mathbf{L}, \overline{y}_i \in \mathcal{Y} \backslash y_i : \; w^T \left[\Phi(x_i, y_i) - \Phi(x_i, \overline{y}_i)\right] \geq 1 - \xi_i.$$

After we have learned w and ξ, the classification of a new example,x, is done by

$$h(x) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \; w^T \Phi(x, y).$$

In this margin-based learning framework, we observed that for a training example $(x_i, y_i) \in \mathbf{L}$ the score associated with the correct label y_i, $w^T\Phi(x_i, y_i)$, is greater than the scores associated with any other labels $\overline{y}_i \neq y_i$, $w^T\Phi(x_i, \overline{y}_i)$, by at least the amount, $1 - \xi_i$. In Figure 1, we demonstrate how the relative positions of the scores associated with different labels, $w^T\Phi(x_i, \cdot)$, change from before training to after training for a fully labeled example, (x_i, y_i).

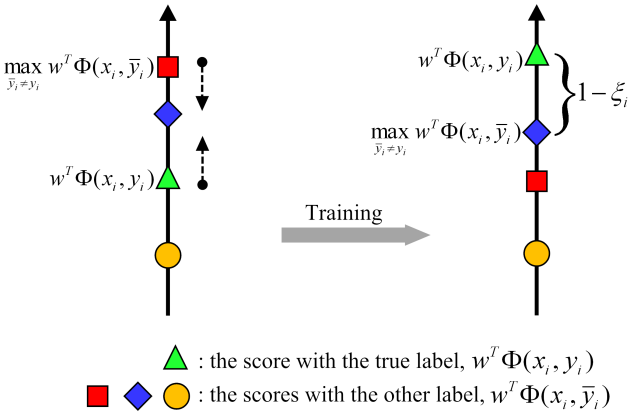

Fig. 1. Illustration of how the relative positions of the scores associated with different labels, $w^T\Phi(x_i, \cdot)$, change from before training to after training for a fully labeled example.

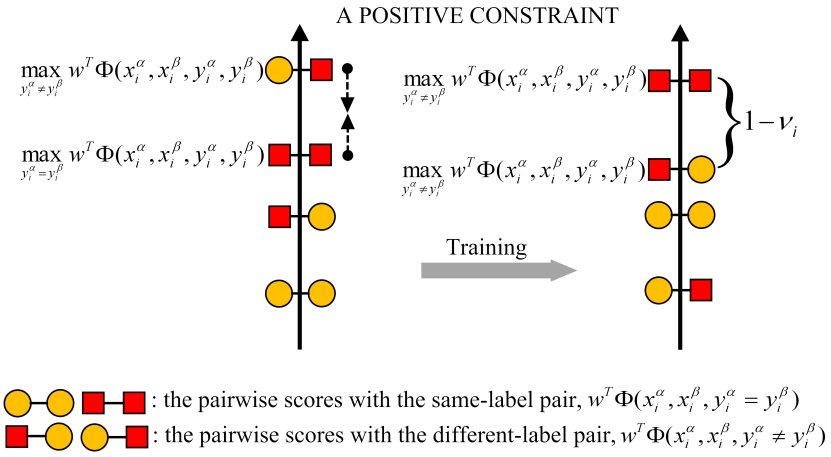

Fig. 2. Illustration of how the relative positions of the pairwise scores associated with label-pairs, $w^T\Phi(x_i^\alpha, x_i^\beta, \cdot, \cdot)$, change from before training to after training for a positive pairwise constraint.

In a similar manner, we will incorporate the pairwise constraint information into the margin-based learning framework. Specifically, given a positive pairwise constraint $(x_i^\alpha, x_i^\beta, +1)$, we want the maximum score associated with the same-label pairs $y_i^\alpha = y_i^\beta$,

$$\max_{y_i^\alpha = y_i^\beta} \left[w^T\Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right],$$

to be greater than the maximum score associated with any different-label pairs $y_i^\alpha \neq y_i^\beta$,

$$\max_{y_i^\alpha \neq y_i^\beta} \left[w^T \Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right],$$

by a soft margin of at least $1 - \nu_i$. Similarly, for a negative pairwise constraint $(x_i^\alpha, x_i^\beta, -1)$ we have the following inequality,

$$\max_{y_i^\alpha \neq y_i^\beta} \left[w^T \Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right] - \max_{y_i^\alpha = y_i^\beta} \left[w^T \Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right] \geq 1 - \nu_i.$$

In Figure 2, we demonstrate how the relative positions of the pairwise scores associated with label-pairs, $w^T \Phi(x_i^\alpha, x_i^\beta, \cdot, \cdot)$, change from before training to after training for a positive pairwise constraint, $(x_i^\alpha, x_i^\beta, +1)$. In our framework, we define the mapping of a pairwise constraint as the sum of the individual example-label scores,

$$\Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) = \Phi(x_i^\alpha, y_i^\alpha) + \Phi(x_i^\beta, y_i^\beta).$$

Formally, the pairwise constraint SVM classification (PCSVM) learns a weight vector w and slack variables ξ, ν via the following margin-based quadratic optimization problem:

Optimization Problem II: PCSVM

$$\min_{w, \xi \geq 0, \nu \geq 0} : \frac{\lambda}{2} \|w\|^2 + \frac{1}{n+m} \left(\sum_{i=1}^{n} \xi_i + \sum_{i=1}^{m} \nu_i \right) \tag{2}$$

subject to:

$$\forall (x_i, y_i) \in \mathbf{L}, \overline{y}_i \in \mathcal{Y} \backslash y_i : \; w^T \left[\Phi(x_i, y_i) - \Phi(x_i, \overline{y}_i) \right] \geq 1 - \xi_i,$$

$$\forall (x_i^\alpha, x_i^\beta, \widetilde{y}_i) \in \mathbf{C}^+ :$$

$$\max_{y_i^\alpha = y_i^\beta} \left[w^T \Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right] - \max_{y_i^\alpha \neq y_i^\beta} \left[w^T \Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right] \geq 1 - \nu_i,$$

$$\forall (x_i^\alpha, x_i^\beta, \widetilde{y}_i) \in \mathbf{C}^- :$$

$$\max_{y_i^\alpha \neq y_i^\beta} \left[w^T \Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right] - \max_{y_i^\alpha = y_i^\beta} \left[w^T \Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right] \geq 1 - \nu_i,$$

where $\mathbf{C}^+ = \{(x_i^\alpha, x_i^\beta, \widetilde{y}_i) \in \mathbf{C} \mid \widetilde{y}_i = +1\}$ and $\mathbf{C}^- = \{(x_i^\alpha, x_i^\beta, \widetilde{y}_i) \in \mathbf{C} \mid \widetilde{y}_i = -1\}$ are the set of same/positive constraints and different/negative constraints respectively. The classification of test examples is done in the same manner as for the multiclass SVM classification.

In order to solve the pairwise constraint SVM classification, we extend the Primal QP solver by [21]. The PCSVM is a simple and effective iterative algorithm for solving the above QP and does not require transforming to the dual

Algorithm 1 Pairwise Constraint SVM Classification (PCSVM)

Input: $\mathbf{L}$ - the labeled data, $\mathbf{C}$ - the pairwise constraint data
λ and T - parameters of the QP

Initialize: Choose w_1 such that $\|w_1\| \leq 1/\sqrt{\lambda}$

for $t = 1$ **to** T **do**

Set $\mathbf{A} = \left\{ (x_i, y_i) \in \mathbf{L} \mid w_t^T \Phi(x_i, y_i) - \max_{\overline{y}_i \neq y_i} w_t^T \Phi(x_i, \overline{y}_i) < 1 \right\}$

Set $\mathbf{A}^+ = \left\{ (x_i^\alpha, x_i^\beta, \widetilde{y}_i) \in \mathbf{C}^+ \mid \max_{y_i^\alpha = y_i^\beta} \left[w_t^T \Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right] - \max_{y_i^\alpha \neq y_i^\beta} \left[w_t^T \Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right] < 1 \right\}$

Set $\mathbf{A}^- = \left\{ (x_i^\alpha, x_i^\beta, \widetilde{y}_i) \in \mathbf{C}^- \mid \max_{y_i^\alpha \neq y_i^\beta} \left[w_t^T \Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right] - \max_{y_i^\alpha = y_i^\beta} \left[w_t^T \Phi(x_i^\alpha, x_i^\beta, y_i^\alpha, y_i^\beta) \right] < 1 \right\}$

Set $\eta_t = \dfrac{1}{\lambda t}$

$$\text{Set } w_{t+\frac{1}{2}} = (1 - \eta_t \lambda) w_t + \frac{\eta_t}{n+m} \left\{ \sum_{(x_i, y_i) \in \mathbf{A}} [\Phi(x_i, y_i) - \Phi(x_i, \overline{y}_i)] + \sum_{(x_i^\alpha, x_i^\beta, \widetilde{y}_i) \in \mathbf{A}^+} \left[\Phi(x_i^\alpha, x_i^\beta, y_+^\alpha, y_+^\beta) - \Phi(x_i^\alpha, x_i^\beta, y_-^\alpha, y_-^\beta) \right] + \sum_{(x_i^\alpha, x_i^\beta, \widetilde{y}_i) \in \mathbf{A}^-} \left[\Phi(x_i^\alpha, x_i^\beta, y_-^\alpha, y_-^\beta) - \Phi(x_i^\alpha, x_i^\beta, y_+^\alpha, y_+^\beta) \right] \right\}$$

where $\overline{y} = \operatorname{argmax}_{\overline{y}_i \neq y} w_t^T \Phi(x_i, \overline{y}_i)$,

$(y_+^\alpha, y_+^\beta) = \operatorname{argmax}_{y^\alpha = y^\beta} w_t^T \Phi(x_i^\alpha, x_i^\beta, y^\alpha, y^\beta)$,

$(y_-^\alpha, y_-^\beta) = \operatorname{argmax}_{y^\alpha \neq y^\beta} w_t^T \Phi(x_i^\alpha, x_i^\beta, y^\alpha, y^\beta)$

$$\text{Set } w_{t+1} = \min \left\{ 1, \frac{1/\sqrt{\lambda}}{\|w_{t+\frac{1}{2}}\|} \right\} w_{t+\frac{1}{2}}$$

end for

Output: w_{T+1}

formulation. The algorithm alternates between gradient descent steps and projection steps. In each iteration, the algorithm first computes a set of labeled examples $\mathbf{A} \subset \mathbf{L}$, a set of positive pairwise constraints $\mathbf{A}^+ \subset \mathbf{C}^+$, and a set of negative pairwise constraints $\mathbf{A}^- \subset \mathbf{C}^-$ that contain violated examples and pairwise constraints. Then the weight vector w is updated according to the violated sets $\mathbf{A}$, $\mathbf{A}^+$, and $\mathbf{A}^-$. In the projection step, the weight vector w is projected to the sphere of radius $1/\sqrt{\lambda}$. The details of the PCSVM are given in Algorithm 1.

We observed that if $w_1 = 0$ then w_t can be written as

$$w_t = \sum_{x,y} \varphi_{xy} \Phi(x, y).$$

Hence, we can incorporate the usage of kernel when computing inner product operations, i.e.:

$$\langle w, \Phi(x', y') \rangle = \sum_{x,y} \quad \varphi_{xy} \mathbf{K}(x, y, x', y')$$
$$\|w\|^2 = \sum_{x,y} \sum_{x',y'} \varphi_{xy} \varphi_{x'y'} \mathbf{K}(x, y, x', y')$$

In our experiments, we use the polynomial kernel,

$$\mathbf{K}(x, y, x', y') = \langle \Phi(x, y), \Phi(x', y') \rangle^d,$$

where polynomial kernel degree d is chosen from the set $\{1, 2, 3, 4, 5\}$.

The efficiency and guaranteed performance of PCSVM in solving the quadratic optimization problem is shown by the following theorem:

Theorem 1 *Let*

$$R \;=\; 2 \max \left\{ \begin{array}{c} \max\limits_{x,y} \|\Phi(x, y)\|, \\ \max\limits_{x^\alpha, x^\beta, y^\alpha, y^\beta} \|\Phi(x^\alpha, x^\beta, y^\alpha, y^\beta)\| \end{array} \right\}$$

then the number of iterations for Algorithm 1 to achieving a solution of accuracy $\delta > 0$ *is* $\tilde{O}(R^2/(\lambda\delta))$.[1]

3 Related Work

For classification, pairwise constraints have been shown to improve the performance of classifiers. In [3–9], pairwise constraints is used to learn a Mahalanobis metric and then apply distance-based classifier such as KNN to the transformed data. Unlike our proposed method, most metric learning algorithms deal with labeled data indirectly by converting into pairwise constraints. In addition, the work of [1, 2] is most related to our proposed algorithm. In [1], the authors also presented a discriminative learning framework which can learn the decision boundary with labeled data as well as additional pairwise constraints. However, in the binary algorithm, PKLR proposed by [1], a logistic regression loss function is used for binary classification instead of the hinge loss. In [2], the authors proposed a binary classifier which also utilizes pairwise constraint information. The proposed classifier, Linear-PC, is a sign-insensitive estimator of the optimal linear decision boundary.

[1] The proof of Theorem 1 is omitted since it is similar to the one given in [21].

Similarly, pairwise constraints have also shown to be successful in the semi-supervised clustering [10–17]. In particular, COPKmeans [11] is a semi-supervised variant of Kmeans. COPKmeans follows the same clustering procedure of Kmeans while avoiding violations of pairwise constraints. In addition, MPCKmeans [17] utilized both metric learning and pairwise constraints in the clustering process. In MPCKmeans, a separate weight matrix for each cluster is learned to minimize the distance between must-linked instances and maximize the distance between cannot-link instances. Hence, the objective function of MPCKmeans minimizes cluster dispersion under the learned metrics while reducing constraint violations. However, most existing algorithms can only find a local-optimal solution for the clustering problem with pairwise constraints as users' feedback.

4 Experiments

We evaluate our proposed algorithms on fifteen data sets from the UCI repository [22] and the LIBSVM data [23]. A summary of the data sets is given in Table 1. For the PCSVM algorithm, we set the parameters used in the experiments as follows: (i) the SVM λ parameter is chosen from $\{10^i\}_{i=-3}^{3}$; (ii) the kernel degree, d, is selected from the set $\{1, 2, 3, 4, 5\}$; (iii) the number of pairwise constraints is from the set $\{10, 20, 40, 80, 160\}$; (iv) the number of label examples is chosen from the set $\{1, \ldots, 5\}$[2]. The parameters, λ and d, are selected using two fold cross validation on the training pairwise constraints.

Table 1. A summary of the data sets.

Data sets	Classes	Size	Features
australian	2	690	14
spambase	2	2300	57
ionosphere	2	351	34
german	2	1000	24
heart	2	270	13
diabetes	2	768	8
liver-disorder	2	345	6
splice	2	3175	60
mushroom	2	8124	112
svmguide2	3	391	20
vehicle	4	846	18
dermatology	6	179	34
satimage	6	6435	36
segment	7	2310	19
vowel	11	990	10

[2] Both the pairwise constraints and label examples are randomly generated.

In the first set of experiments, we compare the performance of the PCSVM against the two other methods proposed by [1, 2], called PKLR and Linear-PC respectively, on 9 binary data sets. In Figure 3, we plot the performance of PCSVM, Linear-PC, and PKLR versus the number of pairwise constraints when there are 5 fully labeled examples per class. To summarize the information, Figure 4 presents the same information by averaging across 9 binary data sets. For different numbers of pairwise constraints and different numbers of fully labeled examples, we observe that both PCSVM and PKLR show significant improvement over Linear-PC. The inferior performance of Linear-PC is due to the fact that the estimator only finds the optimal linear decision boundary. On the other hand, PCSVM and PKLR are able to handle the non-linear separable case by utilizing the non-linear kernel functions. In addition, we also observe that PKLR tends to produce better performance than PCSVM when the number of training pairwise constraints is small. As the number of pairwise constraints increases, PCSVM outperforms PKLR. An explanation of this phenomenon is that the loss function of PCSVM is not formulated specifically for binary classification.

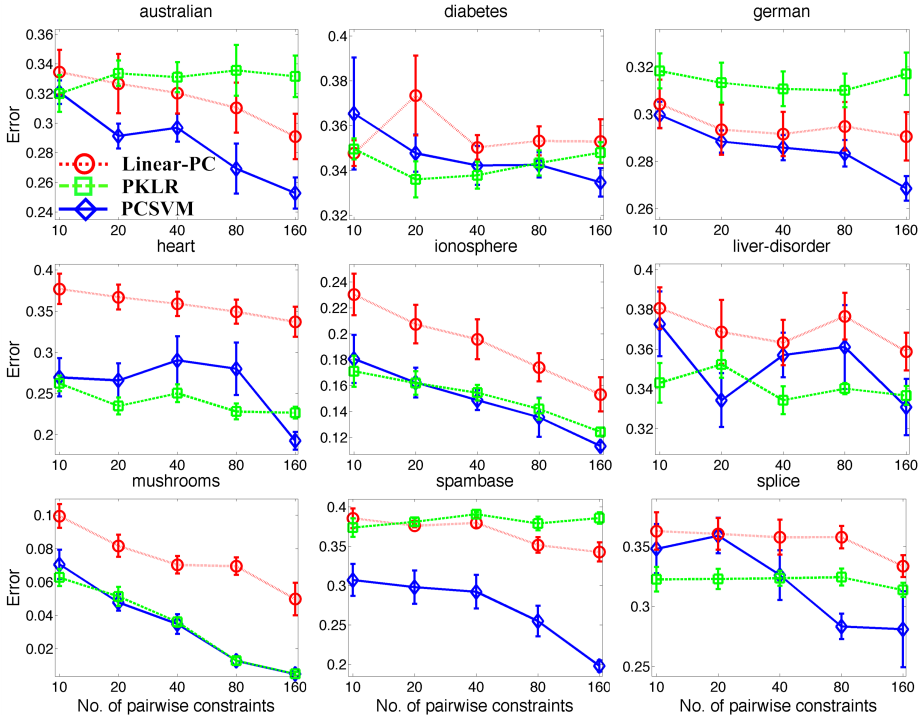

Fig. 3. Classification Performance of 9 binary data sets using 5 label points per class: PCSVM, Linear-PC, and PKLR

In the second set of experiments, we compare the performance of the PCSVM against SVM and SVM-All. SVM is only trained on the labeled data but ignores

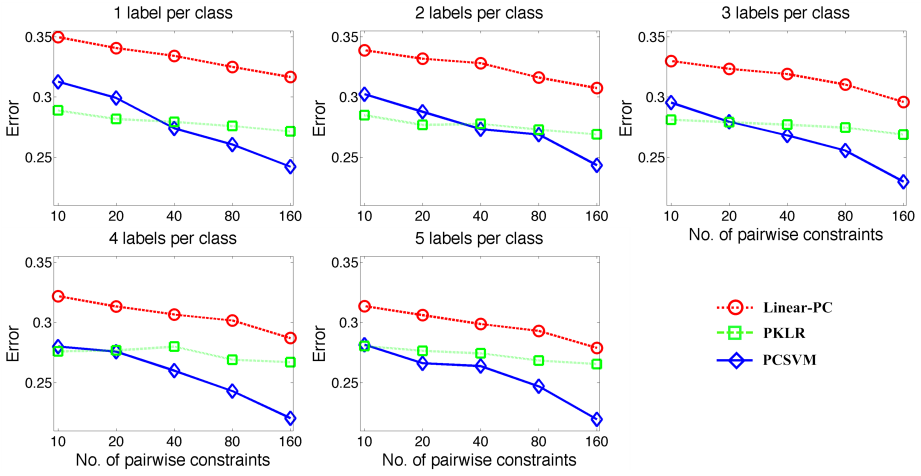

Fig. 4. Average classification performance of 9 binary data sets: PCSVM, Linear-PC, and PKLR

the pairwise constraint information. On the other hand, SVM-All is not only trained on the labeled data but also use the examples in the pairwise constraint data where the true labels are revealed to the algorithm. In Figure 5, we plot the performance of the PCSVM versus the number of pairwise constraints presented in the training set when there are 5 labeled examples per class for all 15 data sets. To summarize the information, Figure 6 shows the same information but averaging across 15 data sets. Across all data sets, we observe that the performance of the PCSVM is between that of SVM and SVM-All. This behavior is what we should expect since pairwise constraint information helps to improve the performance of PCSVM over SVM which does not use this information; and labeled data should still provide more discriminative information to the SVM-All than pairwise constraint information could do to the PCSVM. Note that PCSVM, by learning from the pairwise constraints, on average yields half or more of the error reduction that could be achieved by learning with labels. Hence, SVM and SVM-All can be viewed as the lower and upper bound on the performance of PCSVM.

5 Conclusion

In this paper, we study the problem of classification in the presence of pairwise constraints. We propose a discriminative learning approach which incorporates pairwise constraints into the margin-based learning framework. We also present an efficient algorithm, PCSVM, that integrates pairwise constraints into the multiclass-SVM classification. In experiments with 15 data sets, pairwise constraints not only improves the performance of the binary classification in com-

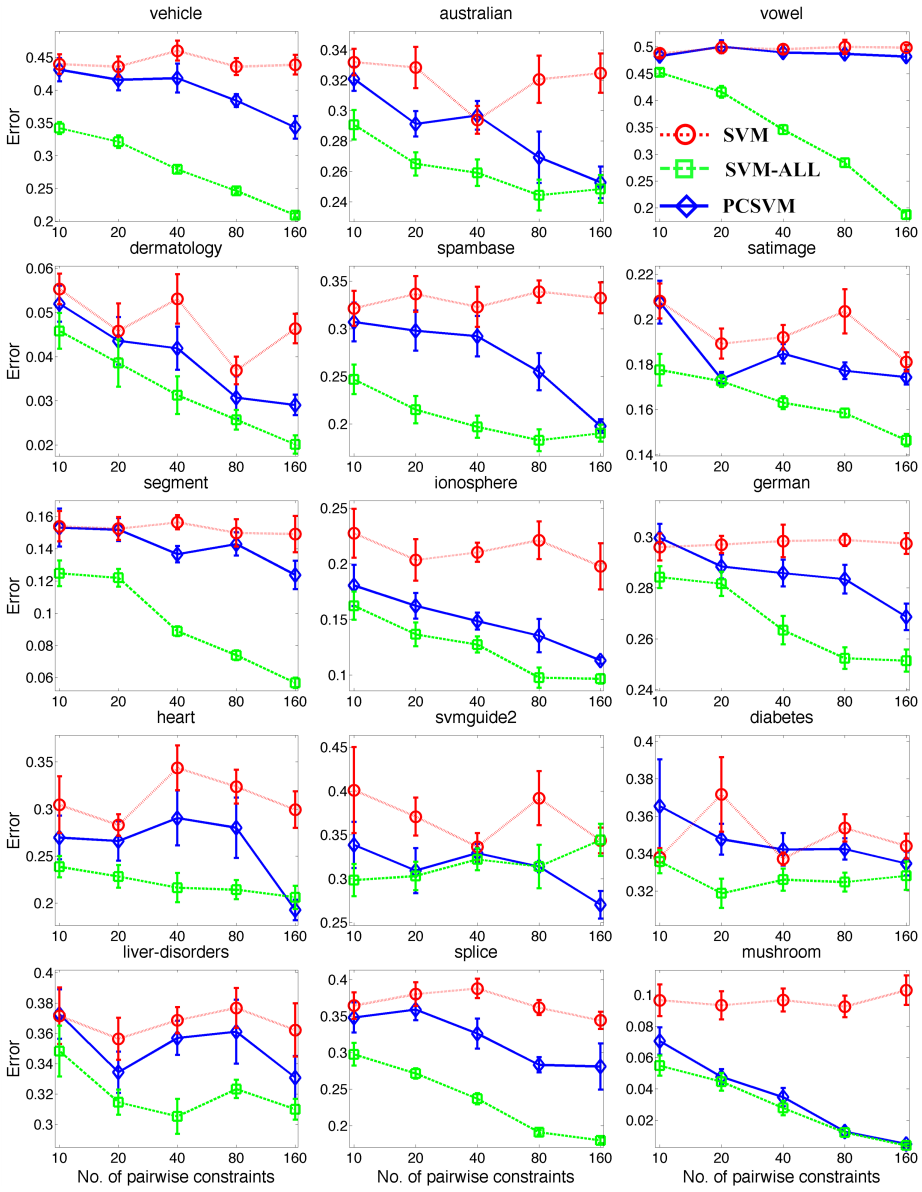

Fig. 5. Classification Performance of 15 data sets using 5 label points per class: PCSVM, SVM, and SVM-All

parison with two other methods (Linear-PC and PKLR) but also significantly increase the performance of the multiclass classification.

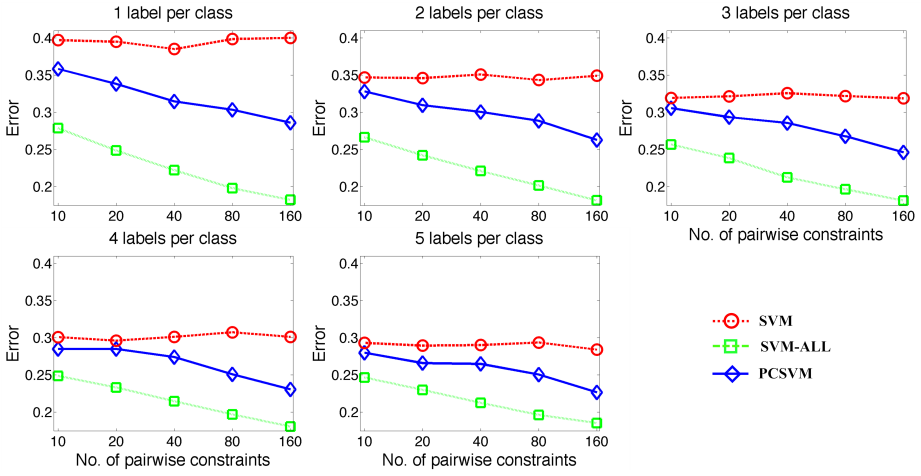

Fig. 6. Average classification performance of 15 data sets

Acknowledgments: This work was supported by NSF CAREER Grant # 0347318.

References

1. Yan, R., Zhang, J., Yang, J., Hauptmann, A.G.: A discriminative learning framework with pairwise constraints for video object classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **28(4)** (2006) 578–593
2. Zhang, J., Yan, R.: On the value of pairwise constraints in classification and consistency. In: Proceedings of the 24th International Conference on Machine Learning. (2007) 1111–1118
3. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the 24th International Conference on Machine Learning. (2007)
4. Globerson, A., Roweis, S.: Metric learning by collapsing classes. In: Advances in Neural Information Processing Systems (NIPS). (2005)
5. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. In: Advances in Neural Information Processing Systems (NIPS). (2004)
6. Schultz, M., Joachims, T.: Learning a distance metric from relative comparisons. In: Advances in Neural Information Processing Systems (NIPS). (2004)
7. Shalev-Shwartz, S., Singer, Y., Ng, A.Y.: Online and batch learning of pseudo-metrics. In: Proceedings of the 21st International Conference on Machine Learning. (2004)
8. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: Advances ing Neural Information Processing Systems (NIPS). (2006)
9. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2005)

10. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. In: Proceedings of the 20th International Conference on Machine Learning. (2003)
11. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Proceedings of 18th International Conference on Machine Learning. (2001)
12. Bilenko, M., Basu, S., Mooney, R.J.: Semi-supervised clustering by seeding. In: Proceedings of 19th International Conference on Machine Learning. (2002)
13. Klein, D., Kamvar, S.D., Manning, C.D.: From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: Proceedings of 19th International Conference on Machine Learning. (2002)
14. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: Advances in Neural Information Processing Systems 15. (2003)
15. Cohn, D., Caruana, R., McCallum, A.: Semi-supervised clustering with user feedback. In: Cornell University Technical Report TR2003-1892. (2003)
16. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. In: Proceedings of 20th International Conference on Machine Learning. (2003)
17. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: Proceedings of 21th International Conference on Machine Learning. (2004)
18. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison (2005)
19. Cozman, F., Cohen, I., Cirelo, M.: Semi-supervised learning of mixture models and bayesian networks. In: Proceedings of the Twentieth International Conference of Machine Learning. (2003)
20. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. Journal of Machine Learning Research **2** (2001) 265–292
21. 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, New York, NY, USA, ACM (2007) 807–814
22. Asuncion, A., Newman, D.: UCI machine learning repository. In: http://www.ics.uci.edu/∼mlearn/MLRepository.html. (2007)
23. Chang, C.C., Lin, C.J.: Libsvm data. In: http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/. (2001)