Semi-supervised Regression with Order Preferences

Xiaojin Zhu
Department of Computer Sciences
University of Wisconsin, Madison
Madison, WI 53705
jerryzhu@cs.wisc.edu

Andrew B. Goldberg
Department of Computer Sciences
University of Wisconsin, Madison
Madison, WI 53705
goldberg@cs.wisc.edu

Abstract

Following a discussion on the general form of regularization for semi-supervised learning, we propose a semi-supervised regression algorithm. It is based on the assumption that we have certain order preferences on unlabeled data (e.g., point x_1 has a larger target value than x_2). Semi-supervised learning consists of enforcing the order preferences as regularization in a risk minimization framework. The optimization problem can be effectively solved by a linear program. Experiments show that the proposed semi-supervised regression outperforms standard regression.

1 Semi-supervised learning as regularization on unlabeled data

Semi-supervised learning works when its assumption on unlabeled data, often expressed as regularization, fits the reality of the problem domain. In this paper we first generalize the regularization formulation of some common semi-supervised learning approaches, namely manifold regularization, semi-supervised support vector machines, and multi-view learning [1, 2, 3]. Regularization for each *individual* approach is not new. However these approaches have been studied largely in isolation. Our general form serves as a bridge to connect them, and to inspire novel semi-supervised approaches. As an example of the latter, we propose a novel algorithm for semi-supervised regression. The proposed regression algorithm is able to incorporate domain knowledge about the relative order of target values on unlabeled points. It thus differs from, and complements, existing semi-supervised regression methods, which do not use such domain knowledge but require multiple views [4, 5].

Let us review the three common semi-supervised learning methods. *Manifold regularization* [6, 7] generalizes several graph-based semi-supervised learning methods. Let l be the number of labeled points, u the number of unlabeled points. Graph-based semi-supervised learning requires a weighted, undirected graph, characterized by an $(l+u) \times (l+u)$ weight matrix $\mathbf{W}$ defined on labeled and unlabeled data. It is assumed that from the features of two points x_i, x_j (e.g., by computing their Euclidean distance), domain experts can assign a non-negative weight w_{ij}. A large w_{ij} implies a preference for $f(x_i), f(x_j)$ to be similar. Therefore subgraphs with large weights tend to have the same label. This is sometimes called the cluster assumption. Let K be a kernel and $\mathcal{H}$ the corresponding Reproducing Kernel Hilbert Space (RKHS). Let $\mathbf{y}$ be the labels, which can be categories for classification or real numbers for regression. Manifold regularization seeks a prediction function $\mathbf{f} \in \mathcal{H}$, such that $\mathbf{f}$ is the solution to

$$\min_{\mathbf{f}\in\mathcal{H}} \sum_{i=1}^{l} c(y_i, f(x_i)) + \lambda_1 \|\mathbf{f}\|_{\mathcal{H}}^2 + \lambda_2 {f_{lu}}^\top L f_{lu}. \qquad (1)$$

The first two terms are standard in kernel machines. The function $c()$ is any loss function, e.g., the hinge loss $c(y, f) = \max(1 - yf, 0)$ used in support vector machines; $\|\mathbf{f}\|_{\mathcal{H}}$ is the RKHS norm of $\mathbf{f}$, which serves as regularization. The two λ's are tunable weights. The third term ${f_{lu}}^\top L f_{lu}$ regularizes $\mathbf{f}$ so that it is smooth over the graph, where $f_{lu} = (f(x_1) \dots f(x_{l+u}))^\top$ is the vector of $\mathbf{f}$ values on labeled and unlabeled data. The $(l+u)\times(l+u)$ matrix L can be the combinatorial graph Laplacian $\text{diag}(\mathbf{W1}) - \mathbf{W}$ where $\mathbf{1}$ is the all-one vector. Other variants are possible. If L is the combinatorial graph Laplacian, the third term can be shown to be ${f_{lu}}^\top L f_{lu} = \sum_{ij} w_{ij}(f(x_i) - f(x_j))^2$. This term penalizes the difference between $f(x_i), f(x_j)$ more when w_{ij} is large, thus enforcing the smoothness assumption.

Semi-supervised support vector machines (also known as transductive SVMs) [8, 9] are based on a different assumption, that the decision boundary should avoid dense regions. The problem can be defined as [10]

$$\min_{\mathbf{f}\in\mathcal{H}} \sum_{i=1}^{l} c(y_i, f(x_i)) + \lambda_1 \|\mathbf{f}\|^2_{\mathcal{H}} + \lambda_2 \sum_{i=l+1}^{l+u} \max(1-|f(x_i)|, 0). \tag{2}$$

As before, the function $c()$ is the hinge loss on labeled points. Since $\text{sign}(f(x_i))f(x_i) = |f(x_i)|$, the third term is the hinge loss on *unlabeled* points, if we assign the putative label $\text{sign}(f(x_i))$ to unlabeled point x_i according to the predictor $\mathbf{f}$. Such loss is zero if $f(x_i) \notin (-1, 1)$. To avoid loss from the third term, the predictor $\mathbf{f}$ should attempt to produce $|f(x_i)| \geq 1$ on unlabeled points. It is equivalent to finding a decision boundary $\mathbf{f} = 0$ so that the unlabeled points are outside the margin. This in turn means the decision boundary will avoid dense unlabeled regions. Because the third term is not convex, much research has focused on effectively solving (2).

Multi-view learning [11] employes multiple learners. The regularization term encodes the domain knowledge that the M learners should agree with each other on unlabeled data [5, 12]:

$$\begin{aligned} \min_{\mathbf{f}\in\mathcal{H}} \quad & \textstyle\sum_{v=1}^{M} \left(\sum_{i=1}^{l} c(y_i, f_v(x_i)) + \lambda_1 \|\mathbf{f}_v\|^2_{\mathcal{H}_v} \right) \\ & + \lambda_2 \textstyle\sum_{u,v=1}^{M} \sum_{i=l+1}^{l+u} (f_u(x_i) - f_v(x_i))^2. \end{aligned} \tag{3}$$

Comparing the three approaches (1)(2)(3), we note the common role of unlabeled data: It acts as *data-dependent regularization* in addition to the standard RKHS norm $\|\mathbf{f}\|_{\mathcal{H}}$. Such regularization encodes the assumptions of each method. We argue that novel assumptions, stemming from domain knowledge and taking the form of regularization, give rise to novel semi-supervised learning algorithms. We unify a large family of semi-supervised learning algorithms by the optimization problem

$$\min_{\mathbf{f}\in\mathcal{H}} \sum_{i=1}^{l} c(y_i, f(x_i)) + \lambda_1 \Omega(\|\mathbf{f}\|_{\mathcal{H}}) + \lambda_2 r(f(x_1) \ldots f(x_{l+u})). \tag{4}$$

The function $c()$ is a loss function that we choose for classification or regression; $\Omega()$ is a strictly monotonic increasing function; $r()$ is a regularization term that depends on $\mathbf{f}()$ values on labeled and unlabeled data. It may also depend on $\mathbf{x}$, and the labeled data's labels. The function $r()$ encodes the assumptions of semi-supervised learning, and should be chosen carefully to fit the problem domain. The solution of (4) can be characterized by a representer theorem for semi-supervised learning:

Theorem 1 (Representer Theorem for Semi-Supervised Learning) *Let K be a kernel and $\mathcal{H}$ the RKHS. The minimizer $\mathbf{f}^* \in \mathcal{H}$ of (4) admits the form $f^*(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x)$.*

The theorem states that the minimizer is expressed by a finite set of representers $K(x_i, \cdot)$ over *both labeled and unlabeled data*. It is a special case of the original representer theorem [13, 14], and a simple generalization of Theorem 2.2 in [7] to arbitrary functions $r()$. The proof uses the standard orthogonality argument, and is omitted for space consideration. The significance of (4) lies in its interpretation for semi-supervised learning:

1. The representer theorem holds for arbitrary $r()$. This in theory allows one to encode complex domain knowledge about the unlabeled data for semi-supervised learning. In particular $r()$ does not need to be convex (for example, it is not in (2)). However, for computational reasons we will focus on a simple, convex $r()$ in the next section.

2. It allows higher-order interactions among unlabeled data points (e.g. [15] and references therein), which is important for certain applications like computer vision.

3. It allows assumptions to be combined. For instance, one can create a hybrid of manifold regularization (1) and semi-supervised support vector machines (2) by $r(\mathbf{f}) = \lambda \sum_{ij} w_{ij}(f(x_i) - f(x_j))^2 + (1-\lambda)\sum_i \max(1 - |f(x_i)|, 0)$. Such combination has only briefly been attempted in semi-supervised learning [16].

Next we focus on a special case of (4), where $r()$ encodes domain knowledge about the relative order of $f(x)$ on unlabeled points. This leads to a novel semi-supervised regression algorithm.

2 Semi-supervised regression with order preferences

As a motivating example, consider the task of predicting real estate prices. The price of a house varies significantly depending on its location and many other factors. However everything else being roughly equal, a 4-bedroom house is more expensive than a 3-bedroom one. A domain expert can define 'roughly equal', and claim that under such condition the feature *number-of-rooms* determines the order of house prices. It is worth noting that modeling such knowledge as positive correlation between the original feature and the target can be difficult in *non-linear* kernel regression, because of kernel feature mapping. Besides, in general the correlation may only hold for part of the range of the feature value, and it would be inappropriate to force the same correlation across the range. Instead, we can encode such domain knowledge with *order preferences* on unlabeled points in a semi-supervised learning setting. That is, for all pairs of unlabeled points x_i, x_j satisfying the 'roughly equal' condition, such knowledge specifies the *order* between their target values $f(x_i)$ and $f(x_j)$, even though their actual target values are unknown. Respecting the domain knowledge then amounts to incorporating the order preferences into semi-supervised learning. When labeled data is scarce, these order preferences should improve our regression model. A similar situation arises in predicting Internet file transfer rates based on network properties like round trip time, available bandwidth, queuing delay, package loss rate, etc. The features have intuitive impact on transfer rate, but the exact relation is highly non-linear and unknown. We can however easily create order preferences on unlabeled data using domain knowledge. In general, order preferences can encode potentially complex domain knowledge.

Let us formally define our regression problem. Besides a labeled training set, we assume that we are given p *order preferences* between pairs of unlabeled points. An order preference is defined by a tuple (i, j, d, w), with the interpretation that we would like $f(x_i) - f(x_j) \geq d$. As we see later, it is a preference rather than a hard constraint. The scalar $w \geq 0$ is the weight (confidence) for the preference. Obviously knowing the order preferences is much weaker than knowing the labels of unlabeled points. We would like to use the order preferences to improve regression.

It is possible to represent the order preferences as directed edges in a graph [17]. The graph differs from graph-based semi-supervised learning (1): the former expresses asymmetric order information, while the latter expresses symmetric similarity information. However, order preferences can also encode similarity. For example, the two preferences $(i, j, 0, w), (j, i, 0, w)$ encode $f(x_i) = f(x_j)$. More generally $(i, j, -\epsilon, w), (j, i, -\epsilon, w)$ encodes $|f(x_i) - f(x_j)| \leq \epsilon$. It is also easy to encode $a \leq f(x_i) - f(x_j) \leq b$. Unary preferences $f(x_i) \leq g(x_i)$, or $f(x_i) = g(x_i)$, or $f(x_i) \geq g(x_i)$, where g is some given function, are special cases of order preference. The unary preferences are closely related to the work of Mangasarian et al. [18], which adds such domain knowledge to kernel machines.

With the order preferences we now define the regularization term r in (4) for semi-supervised regression. Intuitively if $\mathbf{f}$ satisfies all order preferences, r should be zero; if $\mathbf{f}$ violates some, r increases. A natural choice is to use a shifted hinge function: for order preference (i, j, d, w), the regularization term is $w \max(d - (f(x_i) - f(x_j)), 0)$. That is, it is zero if the preference is satisfied; otherwise it is the amount the preference falls short, weighted by w. We define the regularization term r in (4) as the sum of shifted hinge function on all order preferences:

$$r(\mathbf{f}) = \sum_{q=1}^{p} w_q \max(d_q - (f(x_{iq}) - f(x_{jq})), 0). \qquad (5)$$

We note that order preferences have been used in ranking problems [19, 20, 21, 22]; in particular [23] employed a similar shifted hinge function for ranking. However they have not been used in regression before. For $c()$ in (4) we use the ϵ-insensitive loss $c(y, f) = |y - f|_\epsilon$ in support vector regression [24]:

$$|y - f|_\epsilon = \begin{cases} 0 & \text{if } |y - f| \leq \epsilon \\ |y - f| - \epsilon & \text{otherwise.} \end{cases} \qquad (6)$$

If we further choose $\Omega(\|f\|_\mathcal{H}) = \|f\|^2_\mathcal{H}$, we end up with the optimization problem:

$$\min_{\mathbf{f} \in \mathcal{H}} \quad \textstyle\sum_{i=1}^{l} |y_i - f(x_i)|_\epsilon + \lambda_1 \|\mathbf{f}\|^2_\mathcal{H} + \lambda_2 \sum_{q=1}^{p} w_q \max(d_q - (f(x_{iq}) - f(x_{jq})), 0). (7)$$

The first two terms constitute standard support vector regression [24]. The third term extends it to semi-supervised learning. The optimization can be solved by a quadratic program. However, we will not develop (7) further in the paper. Instead, noticing both $c()$ and $r()$ are piece-wise linear, we propose an alternative optimization problem that can be solved by a *linear program.*

3 The linear program formula

We replace $\|\mathbf{f}\|^2_\mathcal{H}$ in (7) with a linear term, in this case the 1-norm of the dual parameters. The formulation originates from generalized support vector machines [25]. Such 1-norm support vector machines [26, 27, 28] are comparable in performance to the standard 2-norm support vector machines. Let $K(x, \mathbf{x}_{1:l})$ denote the row vector of kernel values between a point x and the labeled data $\mathbf{x}_{1:l}$. We represent our function $\mathbf{f}$ in dual form by

$$f(x) = K(x, \mathbf{x}_{l:l})\alpha + \alpha_0 \qquad (8)$$

where α is a column vector of dual parameters, one for each labeled point; α_0 is a bias scalar. (8) amounts to approximating the representer theorem (Theorem 1) by setting dual parameters on unlabeled data to zero for efficiency. One can also select a subset of unlabeled points and add them to (8). Our semi-supervised regression problem is

$$\min_{\alpha, \alpha_0} \quad \textstyle\frac{1}{l} \sum_{i=1}^{l} |y_i - f(x_i)|_\epsilon + \lambda_1 \|\alpha\|_1 + \lambda_2 \frac{1}{p} \sum_{q=1}^{p} w_q \max(d_q - (f(x_{iq}) - f(x_{jq})), 0) (9)$$

$\|\alpha\|_1 = \sum_{i=1}^{l} |\alpha_i|$ is the 1-norm of α. The bias α_0 is not regularized. We transform (9) into a linear program by introducing auxiliary variables for the three terms respectively. Let $\mathbf{1}$ be the all-one vector, ξ an l-vector of slack variables. Vector inequalities are element-wise. In matrix notation the first term of (9) is equivalent to

$$\begin{aligned} &\min_{\alpha,\alpha_0,\xi} \quad \tfrac{1}{l}\mathbf{1}^\top \xi \\ &\text{s.t.} - \xi - \epsilon\mathbf{1} \le \mathbf{y}_{1:l} - K(\mathbf{x}_{1:l}, \mathbf{x}_{1:l})\alpha - \alpha_0\mathbf{1} \le \xi + \epsilon\mathbf{1} \\ &\quad \xi \ge 0. \end{aligned} \tag{10}$$

Let η be an l-vector. The second term of (9) is equivalent to

$$\begin{aligned} \min_{\alpha,\eta} \quad & \lambda_1 \mathbf{1}^\top \eta \\ \text{s.t.} \quad & -\eta \le \alpha \le \eta. \end{aligned} \tag{11}$$

We do not need non-negativity constraints $\eta \ge 0$ since this is implied. For the third term, let ν be a p-vector, $\mathbf{d}$ the difference vector, $\mathbf{w}$ the weight vector, $K(\mathbf{x}^i_{1:p}, \mathbf{x}_{1:l})$ the $p \times l$ kernel matrix between the first points in the order constraints and the labeled data, and $K(\mathbf{x}^j_{1:p}, \mathbf{x}_{1:l})$ the same sized kernel matrix between the second points in the order constraints and the labeled data. The third term is equivalent to

$$\begin{aligned} \min_{\alpha,\nu} \quad & \tfrac{\lambda_2}{p}\mathbf{w}^\top \nu \\ \text{s.t.} \quad & (K(\mathbf{x}^i_{1:p}, \mathbf{x}_{1:l}) - K(\mathbf{x}^j_{1:p}, \mathbf{x}_{1:l}))\alpha \ge \mathbf{d} - \nu \\ & \nu \ge 0. \end{aligned} \tag{12}$$

Putting the three terms together, our final linear program for semi-supervised learning with order preferences is

$$\begin{aligned} &\min_{\alpha,\alpha_0,\xi,\eta,\nu} \quad \tfrac{1}{l}\mathbf{1}^\top \xi + \lambda_1 \mathbf{1}^\top \eta + \tfrac{\lambda_2}{p}\mathbf{w}^\top \nu \\ &\text{s.t.} - \xi - \epsilon\mathbf{1} \le \mathbf{y}_{1:l} - K(\mathbf{x}_{1:l}, \mathbf{x}_{1:l})\alpha - \alpha_0\mathbf{1} \le \xi + \epsilon\mathbf{1} \\ &\quad \xi \ge 0 \\ &\quad - \eta \le \alpha \le \eta \\ &\quad (K(\mathbf{x}^i_{1:p}, \mathbf{x}_{1:l}) - K(\mathbf{x}^j_{1:p}, \mathbf{x}_{1:l}))\alpha \ge \mathbf{d} - \nu \\ &\quad \nu \ge 0. \end{aligned} \tag{13}$$

This is a linear program with $3l + p + 1$ variables and $5l + 2p$ constraints. The global optimal solution can be easily found.

4 Experiments

We demonstrate the benefit of semi-supervised regression with three groups of experiments. We implemented our linear program (13) using CPLEX. All experiments ran quickly. In all experiments, ϵ in the ϵ-insensitive loss (6) was set to 0, and preference weights $\mathbf{w}$ were set to 1. We use the acronym SSL for (13), and SVR for the corresponding supervised 1-norm support vector regression (i.e., $\lambda_2 = 0$). We also experimented with standard 2-norm support vector regression using SVM$^{\text{light}}$ [29], and the results were comparable to SVR. Since our focus is on the effect of order preference in improving SVR, we will use SVR as our baseline in the experiments.

4.1 A toy example

First we use a toy example to illustrate order preferences. We constructed a polynomial function of degree 3 as our target (the dotted line in Figure 1(a)). We randomly sampled three points (the open circles) from the target function as training data and gave them to SVR. For this experiment we used a linear kernel and set $\lambda_1 = 0$. Since there were not enough training data points, SVR produced a fit (the dashed line) through the training points but very different from the target.

We then randomly selected a pair of unlabeled points $-0.15, 0.30$. Note they did not coincide with the training points. Without revealing the actual target values at these points, we constructed an order preference using their true order: $(0.30, -0.15, 0, 1)$, or equivalently $f(0.30) - f(-0.15) \ge 0$. Note we set $d = 0$ so that the order preference specified their order but not the true difference; hence it was weaker. We set $w = 1$. In Figure 1(a) the order preference is shown at the lower left as a line linking the two unlabeled points (black dots). The point with the larger value has a larger dot. SVR happened to violate the order preference. With the three training points and this order preference, SSL produced a better fit (the solid line).

In Figure 1(b) we added more order preferences, generated similarly from random unlabeled point pairs and their true order. Note some preferences were already satisfied by SVR. The SSL function was further improved. We consistently observed such behavior in repeated random trials.

4.2 Benchmark datasets

We experimented with five regression benchmark datasets (Boston, Abalone, Computer, California, Census; Available at `http://www.liacc.up.pt/~ltorgo/Regression/DataSets.html`), and report results on all of them. One difficulty in working with such standard datasets is creating sensible order preferences on unlabeled data. Ideally the order preferences would be prepared by experts with domain knowledge on the tasks. Lacking such knowledge, we had to create simulated order preferences from the relation of true values on unlabeled points (more details later; Note, however, we never give out the true values themselves). Therefore our results on benchmark datasets should be viewed as 'oracle experiments.' Nonetheless they are useful indications of how well our semi-supervised regression would perform given such domain knowledge.

For each benchmark dataset, we normalized its input features to zero mean, unit variance. For categorical features with k distinct values, we mapped them into indicator vectors of length k. We used Radial Basis Function (RBF) kernels $k(x, x') = \exp(-\sigma\|x - x'\|^2)$ for all datasets. We used 5-fold cross validation to find the optimal RBF bandwidth σ, and SVR 1-norm weight λ_1. The parameters were tuned for SVR on a 9×9 logarithmic grid in $10^{-4} \leq \sigma \leq 10^4$ and $10^{-4} \leq \lambda_1 \leq 10^4$. λ_2 is a nuance parameter. In our experiments we simply fixed it at 1. This is partly justified by the fact that in (9), the 'shifted hinge function' is on a similar scale to the ϵ-insensitive loss; both incur a linear penalty when violated. Tuning λ_2 might produce better results than reported here, but with limited labeled data (which has been used to tune λ_1 and σ for SVR already) it is hard to do.

All experiments were repeated for 20 random trials. Different algorithms shared the same random trials so we could perform paired statistical tests. In each trial we split the data into three parts: l labeled points, u unlabeled points that were used to generate order preferences, and test points that were the rest of the dataset (see Table 1 Partition). Test points were unseen by either algorithm during training. All results we report are test-set mean-absolute-error over the 20 trials. Let t be the test set size. Test-set mean-absolute-error is defined as $\sum_{i \in \text{test}} |y_i - f(x_i)|/t$. We address the following questions:

Can order preferences improve regression? We randomly sampled with replacement $p = 1000$ pairs (x_i, x_j) from the u unlabeled points. For each sampled pair, we generated an order preference from the true target values y_i, y_j. Without loss of generality let $y_i \geq y_j$. Our simulated order preference was

$$f(x_i) - f(x_j) \geq 0.5(y_i - y_j). \qquad (14)$$

Let us explain our order preferences. We could have created the 'perfect' order preferences with the pair: $f(x_i) - f(x_j) \geq y_i - y_j$ and $f(x_i) - f(x_j) \leq y_i - y_j$. They together encode $f(x_i) - f(x_j) = y_i - y_j$. But we felt it might be difficult to know the exact difference $y_i - y_j$ in real tasks. So we chose not to encode equality preferences. With inequality preferences, we could have set $f(x_i) - f(x_j) \geq 0$. It would only encode order, without any information on the actual difference. But in real tasks one might have some rough estimate of the difference, and (14) was meant to simulate this estimate. Another alternative, $f(x_i) - f(x_j) \geq y_i - y_j$, actually produces slightly inferior preferences as we will soon see. Table 1 compares the test-set mean-absolute-error of SVR and SSL. The differences on all datasets are significant with a paired t-test at the 0.01 level. We conclude that, with the order preferences (14), SSL significantly improves regression performance over SVR.

What if we change the number of order preferences p? In semi-supervised learning one expects a larger gain with more unlabeled data, or the number of order preferences p. We systematically varied p from 10 to 5000, keeping everything else the same as in Table 1. Figure 2(a) shows that it was indeed the case. A very small p sometimes hurts SSL, making it worse than SVR. But as p grows larger SSL rapidly improves, and levels off at around $p = 100$. This indicates that one needs only a moderate amount of order preferences to enjoy the benefit.

What if we change the labeled data size l? In semi-supervised learning the benefit of unlabeled data is expected to decrease with more labeled data. We fixed the number of order preferences $p = 1000$, and systematically varied l. As expected, Figure 2(b) shows that SSL is most useful when l is small, and the benefit diminishes as l grows.

How precise do the order preferences need to be? Extending (14), one can define order preferences as $f(x_i) - f(x_j) \geq \beta(y_i - y_j)$ where β controls how precise they are. As mentioned earlier, $\beta = 0$ only supplies order information, and a larger β estimates the differences. We varied β from 0 to 2 (over-estimate) for the experiments in Table 1. Figure 2(c) shows that with only the order ($\beta = 0$) SSL already outperformed SVR. With a conservative estimate of the differences ($0 < \beta < 1$) SSL was even better. The value $\beta = 1$ was not as good since $f(x_i) - f(x_j) \geq y_i - y_j$ would selectively penalize $f(x_i) = f(x_j) + y_i - y_j - \delta$ but not $f(x_i) = f(x_j) + y_i - y_j + \delta$ for any $\delta > 0$, thus introducing a bias. Finally over-estimating the differences ($\beta = 2$) was clearly bad. In summary, one wants to use a conservative estimate $0 \leq \beta < 1$. This is advantageous in practice, since one does not need to know the precise differences, and can err on the safe side.

4.3 Sentiment analysis in movie reviews

Finally, we experimented with sentiment analysis in movie reviews. Given a movie review text document x, we would like to predict $f(x)$, the rating (e.g., '4 stars') given to the movie by the reviewer. We assume that by looking at the wording of unlabeled reviews, one can determine that some movies will likely be rated higher than others (even though we do not know their actual ratings). These are incorporated as order preferences. We worked on the "scale dataset v1.0" with continuous ratings, available at `http://www.cs.cornell.edu/people/pabo/movie-review-data/` and first used in [30]. It contains four authors with 1770, 902, 1307, 1027 reviews respectively. For each author, we varied $l \in \{30, 60, 120\}$, and let $u = 500, p = 500$. The remaining reviews were test examples. Each experiment was repeated for 20 random trials. All reported results are test-set mean-absolute-error. Each review

Table 1: Benchmark data. All differences are statistically significant.

Dataset	Partition		Mean absolute error		Improvement
	dim	l/u/test	SVR	SSL	
Boston	13	20/200/286	4.780 ± 1.351	3.511 ± 0.376	27%
Abalone	8	30/1000/3147	1.856 ± 0.180	1.685 ± 0.102	9%
Computer	21	30/1000/7162	7.373 ± 3.445	5.364 ± 0.998	27%
California	8	60/1000/19580	58268 ± 4435	52120 ± 1843	11%
Census	16	60/1000/21724	24992 ± 1377	23241 ± 901	7%

document was represented as a word-presence vector, normalized to sum to 1. We used a linear kernel, set $\lambda_1 = 10^{-7}$ and $\lambda_2 = 1$.

As a proxy for expert knowledge, we used a completely separate "snippet dataset" also located at the above URL. The snippet dataset is very different from the scale dataset: it contains single punch line sentences (snippets) instead of full reviews; the snippets have binary positive/negative labels instead of continuous ratings; it comes from different authors on different movies. We trained a standard binary, linear-kernel SVM classifier g on the *snippet* data using SVM$^{\text{light}}$. We then applied g on random pairs of unlabeled movie reviews x_i, x_j in the *scale* dataset. The order of the continuous margin output $g(x_i), g(x_j)$ serves as our proxy for expert knowledge. Since this is a very crude and noisy estimate, we created an order preference $(i, j, 0, 1)$ only if $g(x_i) - g(x_j) > 0.25$, where 0.25 is an arbitrary threshold. Note we set $d = 0$ since we do not know the difference in rating. Table 4.3 presents the results of our sentiment analysis experiments. As expected, SSL is most useful when l is small, and the gain over SVR gradually diminishes with larger l. SSL leads to improvements in all cases, and the differences are significant (*) with paired t-tests at the 0.05 level in about half of the cases[1]. We expect better order preferences from advanced natural language processing (e.g., parsing) to bring larger improvements.

5 Conclusions

We presented a general semi-supervised learning framework. As a special case we proposed a novel semi-supervised regression algorithm with order preferences, formulated as a linear program. It can be easily extended beyond regression, e.g., to ordinal classification [31]. We believe the real power of the general framework (4) lies in its ability to incorporate arbitrary, higher-order regularization terms. Future work on this will expand the frontier of semi-supervised learning.

Acknowledgments

We thank Grace Wahba for discussions on the representer theorem; Olvi Mangasarian, Edward Wild and Michael Ferris on optimization; and Wei Chu on the benchmark datasets.

References

[1] Olivier Chapelle, Alexander Zien, and Bernhard Schölkopf, editors. *Semi-supervised learning.* MIT Press, 2006.

[2] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf.

[3] Matthias Seeger. Learning with labeled and unlabeled data. Technical report, University of Edinburgh, 2001.

[4] Zhi-Hua Zhou and Ming Li. Semi-supervised regression with co-training. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

[5] Ulf Brefeld, Thomas Gaertner, Tobias Scheffer, and Stefan Wrobel. Efficient co-regularized least squares regression. In *ICML06, 23rd International Conference on Machine Learning*, Pittsburgh, USA, 2006.

[6] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML05, 22nd International Conference on Machine Learning*, 2005.

[7] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from examples. Technical Report TR-2004-06, University of Chicago, 2004.

[8] Vladimir Vapnik. *The Nature of Statistical Learning Theory.* Springer, second edition, 1995.

[9] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proc. 16th International Conf. on Machine Learning*, pages 200–209. Morgan Kaufmann, San Francisco, CA, 1999.

[10] Ronan Collobert, Fabian Sinz, Jason Weston, and Leon Bottou. Large scale transductive SVMs. *The Journal of Machine Learning Research*, 7(Aug):1687–1712, 2006.

[1] As a sanity check, we also experimented with *wrong* order preferences by intentionally flipping all preferences $(i, j, 0, 1)$ into $(j, i, 0, 1)$. As expected, SSL with wrong orders became *worse* than SVR by 1% – 13% for different authors at $l = 120$.

Table 2: Movie review sentiment analysis mean-absolute-error for each author.

Dataset	l/u/test	SVR	SSL	Improvement
Author (a)	30/500/1240	0.1383 ± 0.0072	0.1362 ± 0.0028	1.5%
	60/500/1210	0.1323 ± 0.0042	0.1311 ± 0.0025	0.9%
	120/500/1150	0.1224 ± 0.0042	0.1219 ± 0.0024	0.4%
Author (b)	30/500/372	0.1645 ± 0.0146	0.1540 ± 0.0046	* 6.4%
	60/500/342	0.1514 ± 0.0063	0.1496 ± 0.0046	* 1.2%
	120/500/282	0.1431 ± 0.0063	0.1416 ± 0.0062	* 1.0%
Author (c)	30/500/777	0.1405 ± 0.0163	0.1357 ± 0.0070	3.4%
	60/500/747	0.1268 ± 0.0072	0.1258 ± 0.0038	0.8%
	120/500/687	0.1150 ± 0.0048	0.1138 ± 0.0047	1.0%
Author (d)	30/500/497	0.1433 ± 0.0151	0.1350 ± 0.0052	* 5.8%
	60/500/467	0.1366 ± 0.0104	0.1293 ± 0.0037	* 5.3%
	120/500/407	0.1256 ± 0.0092	0.1226 ± 0.0038	2.4%

[11] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1998.

[12] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularized approach to semi-supervised learning with multiple views. In *Proc. of the 22nd ICML Workshop on Learning with Multiple Views*, August 2005.

[13] George Kimeldorf and Grace Wahba. Some results on Tchebychean spline functions. *Journal of Mathematics Analysis and Applications*, 33:82–95, 1971.

[14] Bernhard Schölkopf, Ralf Herbrich, and Alexander J. Smola. A generalized representer theorem. In *Proceedings of the Fourteenth Annual Conference on Computational Learning Theory*, 2001.

[15] Sameer Agarwal, Kristin Branson, and Serge Belongie. Higher order learning with graphs. In *ICML06, 23rd International Conference on Machine Learning*, Pittsburgh, USA, 2006.

[16] Olivier Chapelle, Mingmin Chi, and Alexander Zien. A continuation method for semi-supervised SVMs. In *ICML06, 23rd International Conference on Machine Learning*, Pittsburgh, USA, 2006.

[17] O. Dekel, C. Manning, and Y. Singer. Loglinear models for label-ranking. In *Advances in Neural Information Processing Systems (NIPS) 16*, 2004.

[18] O. L. Mangasarian, J. W. Shavlik, and E. W. Wild. Knowledge-based kernel approximation. *Journal of Machine Learning Research*, 5:1127–1141, 2004.

[19] Ralf Herbrich, Klaus Obermayer, and Thore Graepel. Large margin rank boundaries for ordinal regression. In A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000.

[20] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *ICML-05, 22nd International Conference on Machine Learning*, 2005.

[21] Shipeng Yu, Kai Yu, Volker Tresp, and Hans-Peter Kriegel. Collaborative ordinal regression. In *ICML-06, 23nd International Conference on Machine Learning*, 2006.

[22] Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6(July):1019–1041, 2005.

[23] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of KDD '02, the ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, 2002.

[24] Alex Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004.

[25] Olvi Mangasarian. Generalized support vector machines. In A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 135–146. MIT Press, 2000.

[26] Paul Bradley and Olvi Mangasarian. Feature selection via concave minimization and support vector machines. In *ICML98, 15th International Conference on Machine Learning*, California, 1998.

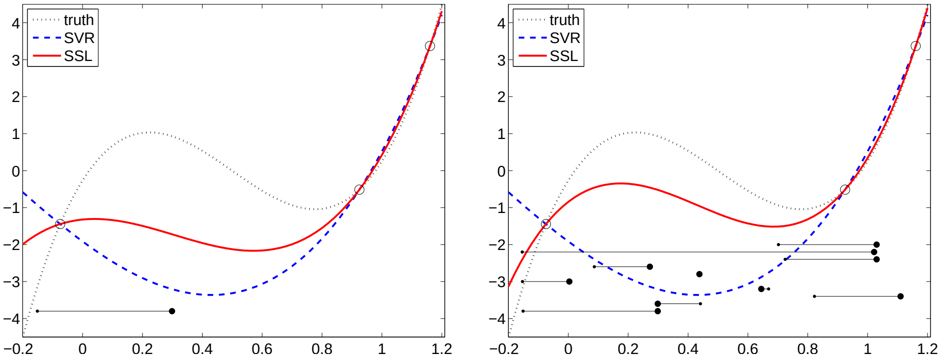
[27] Jinbo Bi, Kristin Bennett, Mark Embrechts, Curt Breneman, and Minghu Song. Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research*, 3:1229–1243, 2003.

[28] Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. In *Neural Information Processing Systems 16*, 2004.

[29] Thorsten Joachims. Making large-scale svm learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

[30] Bo Pang and Lillian Lee. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the Association for Computational Linguistics*, 2005.

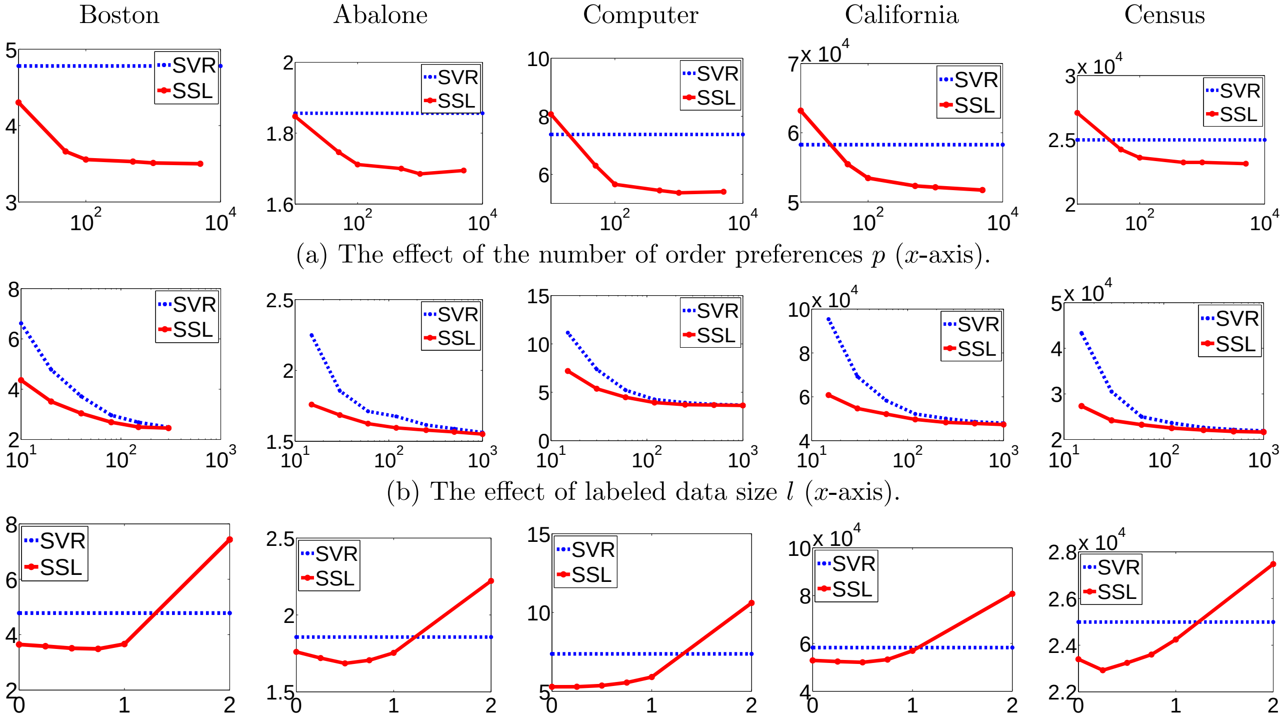
[31] Wei Chu and S. Sathiya Keerthi. New approaches to support vector ordinal regression. In *ICML05, 22nd International Conference on Machine Learning*, pages 145–152, Bonn, Germany, 2005.



(a) one order preference

(b) ten order preferences

Figure 1: A toy example comparing SVR and SSL, showing the benefit of order preferences.



(a) The effect of the number of order preferences p (x-axis).

(b) The effect of labeled data size l (x-axis).

(c) The effect of the difference scaling factor β (x-axis).

Figure 2: The effect of various parameters on SSL on the Benchmark data. y-axis is test-set mean-absolute-error.