Deterministic Annealing for Semi-supervised Kernel Machines

Vikas Sindhwani VIKASS@CS.UCHICAGO.EDU
Department of Computer Science, University of Chicago, Chicago, IL 60637 USA

S. Sathiya Keerthi SELVARAK@YAHOO-INC.COM
Yahoo! Research, Media Studios North, Burbank, CA 91504 USA

Olivier Chapelle OLIVIER.CHAPELLE@TUEBINGEN.MPG.DE
MPI for Biological Cybernetics, Dept. Schölkopf, Spemannstraße 38 72076 Tübingen, Germany

Abstract

An intuitive approach to utilizing unlabeled data in kernel-based classification algorithms is to simply treat the unknown labels as additional optimization variables. For margin-based loss functions, one can view this approach as attempting to learn low-density separators. However, this is a hard optimization problem to solve in typical semi-supervised settings where unlabeled data is abundant. The popular Transductive SVM algorithm is a label-switching-retraining procedure that is known to be susceptible to local minima. In this paper, we present a global optimization framework for semi-supervised Kernel machines where an easier problem is parametrically deformed to the original hard problem and minimizers are smoothly tracked. Our approach is motivated from deterministic annealing techniques and involves a sequence of convex optimization problems that are exactly and efficiently solved. We present empirical results on several synthetic and real world datasets that demonstrate the effectiveness of our approach.

1. Introduction

In this paper, we consider the semi-supervised learning approach based on optimizing regularization objective functions jointly over a continuous space of functions and a discrete space of unknown labels. Given a binary classification problem with l labeled examples $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$ and u unlabeled examples $\{\mathbf{x}'_j\}_{j=1}^{u}$, we seek a real-valued function $f^\star$ and a labeling $\mathbf{y}'^\star = \{y'^\star_1 \ldots y'^\star_u\} \in \{-1,+1\}^u$ for the unlabeled data, by solving:

$$(f^\star, \mathbf{y}'^\star) = \underset{f \in \mathcal{H}_K, \mathbf{y}' \in \{-1,1\}^u}{\operatorname{argmin}} \mathcal{J}(f, \mathbf{y}') = \frac{\lambda}{2}\|f\|_K^2 + \frac{1}{l}\sum_{i=1}^{l} V\left(y_i f(\mathbf{x}_i)\right) + \frac{\lambda'}{u}\sum_{j=1}^{u} V\left(y'_j f(\mathbf{x}'_j)\right)$$

$$\text{subject to:} \quad \frac{1}{u}\sum_{j=1}^{u} \max(0, y'_j) = r \tag{1}$$

where $V : \mathbb{R} \to \mathbb{R}$ is a loss function, $\mathcal{H}_K$ is a Reproducing kernel Hilbert space (RKHS) of functions with kernel K, λ, λ' are real-valued parameters, and $\mathcal{J}$ denotes the objective function. The constraint incorporates prior knowledge about class ratios; r is the fraction of the number of unlabeled examples belonging to the positive class.

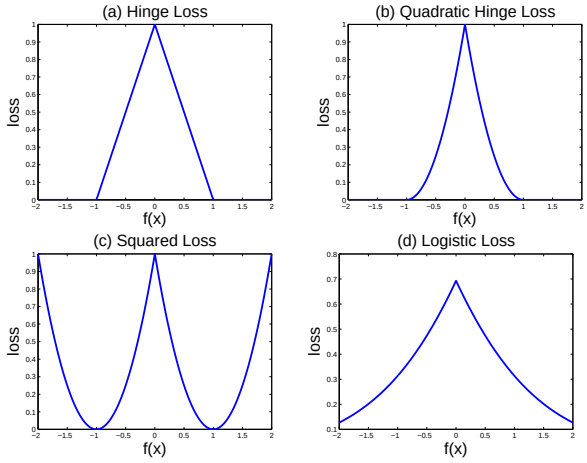
We define the effective loss function V' over an unlabeled example $\mathbf{x}$ as $V'(f(\mathbf{x})) = \min\left[V(f(\mathbf{x})), V(-f(\mathbf{x}))\right]$ corresponding to making an optimal choice for the unknown label of $\mathbf{x}$. Thus, one can formulate an equivalent continuous optimization problem over $\mathcal{H}_K$ alone, with V and V' as the loss functions over labeled and unlabeled examples respectively.

Figure 1 shows the shape of V' for common choices of V. Since small outputs are penalized, decision boundaries that pass through a low-density region in the input space are preferred. Thus, this may be seen as a general approach for semi-supervised learning based on the *cluster assumption*: the assumption that the true decision boundary does not cut data clusters.

The combinatorial nature of this problem due to discrete label variables, or equivalently, the non-convexity

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

Figure 1. Effective Loss function V'



of the effective loss function V' make Eqn. 1 a difficult problem. Currently available approaches (Vapnik & Sterin, 1977; Bennett & Demirez, 1999) for global optimization of Eqn 1 for SVMs are unrealistically slow for typical semi-supervised problems where unlabeled data is plentiful. On the other hand, many recent techniques (Joachims, 1999; Chapelle & Zien, 2005; Collobert et al., 2005; Gartner et al., 2005) are susceptible to local minima on difficult real world classification problems.

The main contributions of this paper are summarized as follows: (1) We present a deterministic annealing framework for global optimization of objective functions of the form in Eqn 1. Our approach generates a sequence of optimization problems approaching the given problem with gradually increasing complexity. These objective functions are locally minimized; the solution for one problem is seeded to the next as the initial guess. This strategy falls in a class of homotopy optimization methods, e.g., see (Nocedal & Wright, 2000; Dunlavy & O'Leary, 2005), and can be also interpreted in terms of maximum entropy principles and deterministic variants of stochastic search techniques (Rose, 1998; Hofmann & Buhmann, 1997). A related class of techniques is the Graduated Non-Convexity method of (Blake & Zisserman, 1987). Some recent work on semi-supervised learning with similar motivation appears in (Chapelle et al., 2006). (2) We derive and evaluate an alternating convex optimization procedure within this framework. This method can utilize off-the-shelf optimization techniques for regularization algorithms. For example, it yields a large scale semi-supervised L_2-SVM for sparse, linear settings (Sindhwani & Keerthi, 2006) when implemented in conjunction with specialized primal methods (Keerthi & DeCoste, 2005). (3) We present an experimental study demonstrating the importance of the idea of annealing on semi-supervised tasks. On some difficult classification problems, our methods show significant improvements over competing algorithms. Whereas recent efforts for solving Eqn. 1 have largely focussed on margin loss functions, our experimental study shows that the classical squared loss can also be very effective for semi-supervised learning within this framework.

This paper is arranged as follows: In section 2 we outline homotopy methods and deterministic annealing for global optimization. Our algorithms are presented in sections 3 and their empirical performance is described in Section 4. Section 5 concludes this paper.

2. Tracking the Global Minimum

2.1. Homotopy Methods

The intuition for our framework for global optimization is simply stated in the following. Consider an unconstrained optimization problem: find $\mathbf{u}^\star = \text{argmin}_{\mathbf{u}\in\mathbb{R}^n} \mathcal{C}(\mathbf{u})$ where the objective function $\mathcal{C}(\mathbf{u})$ may have many possible local minima. Instead of directly dealing with such a problem, we first construct a related "easier" objective function $c(\mathbf{u})$. The minimizers of this function are either known, or easy to compute, for example due to convexity. We then gradually deform the easy problem to the original problem by specifying a smooth map, i.e a homotopy $\mathcal{J}(\mathbf{u}, T)$ parameterized by a real-valued variable T, so that $\mathcal{J}(\mathbf{u}, t_1) = c(\mathbf{u})$ and $\mathcal{J}(\mathbf{u}, t_2) = \mathcal{C}(\mathbf{u})$. Typically, one chooses a convex homotopy such as $\mathcal{J}(\mathbf{u}, T) = (1 - T)\ \mathcal{C}(\mathbf{u}) + T\ c(\mathbf{u})$ where $0 \leq T \leq 1$, or a global homotopy such as $\mathcal{J}(\mathbf{u}, T) = \mathcal{C}(\mathbf{u}) + Tc(\mathbf{u})$ where $T \in [0, \infty)$. T may be varied over an interval starting from t_1 and ending at t_2, in fixed additive steps or by a fixed multiplicative factor. We track local minimizers along the deformation path, at each point starting from the previously computed solution.

Clearly, whether or not this method succeeds in finding a global minimum of $\mathcal{C}(\mathbf{u})$ depends strongly on the choice of the map $\mathcal{J}(\mathbf{u}, T)$ and the manner in which T is varied. For general choices for these, one cannot guarantee a global minimum since there need not be a path in the variable T connecting the sequence of local minimizers to the global minimum, and even if there is one, there is no apriori guarantee that the minimum reached at $T = t_2$ is globally optimal. Moreover, in general, the optimal deformation protocol need not be monotonically increasing or decreasing in T. Inspite of these limitations, in typical applications, it is of-

ten more natural to construct $c(\mathbf{u})$ than to find good starting points. A good choice of the homotopy function and deformation protocol and can drastically reduce local minima problems in the starting and middle stages of the optimization process allowing the focus to be on the globally relevant features of the original objective function.

2.2. Deterministic Annealing

Deterministic annealing may be viewed as a homotopy method for dealing with combinatorial optimization problems. This approach involves two steps. In the first step, discrete variables are treated as random variables over which a space of probability distributions is defined. In the second step, the original problem is replaced by a continuous optimization problem of finding a distribution in this space that minimizes the expected value of the objective function. The latter optimization is performed by a homotopy method using negative of the entropy of a distribution as the "easy", convex function. Specifically, one solves:

$$\mathbf{p}^\star = \underset{\mathbf{p}\in\mathcal{P}}{\operatorname{argmin}}\, E_{\mathbf{p}}\mathcal{C}(\mathbf{u}) - TS(\mathbf{p}) \tag{2}$$

where $\mathbf{u} \in \{0,1\}^n$ are the discrete variables for the objective function $\mathcal{C}(\mathbf{u})$, $\mathcal{P}$ is a family of probability distributions over $\mathbf{u}$, $E_{\mathbf{p}}$ denotes expectation with respect to a distribution $\mathbf{p}$ and $S(\mathbf{p})$ denotes the entropy of $\mathbf{p}$. Note that if $T = 0$ and $\mathcal{P}$ contains all possible point-mass distributions on $\{0,1\}^n$, then the global minimizer $\mathbf{p}^\star$ puts all its mass on the global minimizer of $\mathcal{C}(\mathbf{u})$. Factorial distributions where the associated random variables are taken to be independent are one such class of distributions. With such a choice for $\mathcal{P}$, the first step of "relaxation" to continuous variables does not lose any optimality. The task of finding a minimizer close to the global minimizer in $\mathcal{P}$ is then left to the homotopy method in the second step.

The choice of entropy in the homotopy is well-motivated in various other ways. If T is non-zero and $\mathcal{P}$ is unrestricted, the minimizer in Eqn. 2 is given by the Gibbs distribution $p^\star_{gibbs}(\mathbf{u}) = \frac{\exp(-\mathcal{C}(\mathbf{u})/T)}{\sum_{\{0,1\}^n}\exp(-\mathcal{C}(\mathbf{u})/T)}$. As $T \mapsto 0$, the Gibbs distribution begins to concentrate its mass on the global minimizer of $\mathcal{C}(\mathbf{u})$. Therefore, a stochastic optimization strategy, simulated annealing (Kirkpatrick et al., 1983), samples candidate solutions from a Markov process whose stationary distribution is the Gibbs distribution, while gradually lowering T. In deterministic annealing, one attempts to find a distribution in $\mathcal{P}$ that is closest to the Gibbs distribution in the sense of KL-divergence, resulting in an optimization problem that is equivalent to Eqn. 2 (Bilbro et al., 1991). Finally, one can also interpret this approach in terms of maximum entropy inference (Rose, 1998).

3. Semi-supervised Kernel Machines

We now apply deterministic annealing for solving Eqn. 1 which involves a mix of discrete and continuous variables. The discussion above motivates a continuous objective function,

$$\mathcal{J}_T(f,\mathbf{p}) = E_{\mathbf{p}}\mathcal{J}(f,\mathbf{y}') - TS(\mathbf{p}) \tag{3}$$

defined by taking the expectation of $\mathcal{J}(f,\mathbf{y}')$ (Eqn. 1) with respect to a distribution $\mathbf{p}$ on $\mathbf{y}'$ and including entropy of $\mathbf{p}$ as a homotopy term. Thus, we have:

$$\begin{aligned}\mathcal{J}_T(f,\mathbf{p}) = \frac{\lambda}{2}\|f\|_K^2 + \frac{1}{l}\sum_{i=1}^{l} V\left(y_i f(x_i)\right)\\ +\frac{\lambda'}{u}\sum_{j=1}^{u}\left[p_j V\left(f(\mathbf{x}'_j)\right) + (1-p_j)V\left(-f(\mathbf{x}'_j)\right)\right]\\ +\frac{T}{u}\sum_{j=1}^{u}\left[p_j\ \log\ p_j + (1-p_j)\ \log\ (1-p_j)\right]\end{aligned} \tag{4}$$

where $\mathbf{p} = (p_1 \ldots p_u)$ and p_i may be interpreted as the probability that $y'_i = 1$.

This objective function for a fixed T is minimized under the following class balancing constraint, in place of the balance constraint in Eqn. 1:

$$\frac{1}{u}\sum_{j=1}^{u} p_j = r \tag{5}$$

As in the formulation of (Joachims, 1999), r is treated as a user-provided parameter. It may also be estimated from the labeled examples.

The solution to the optimization problem above $(f^\star_T, \mathbf{p}^\star_T) = \operatorname{argmin}_{f\in\mathcal{H}_K, \mathbf{p}\in[0,1]^u} \mathcal{J}_T(f,\mathbf{p})$ is tracked as the parameter T is lowered to 0. The final solution is given as $f^\star = \lim_{T\to 0} f^\star_T$. In practice, we monitor the value of the objective function in the optimization path and return the solution corresponding to the minimum value achieved.

3.1. Optimization

For any fixed value of T, the problem in Eqns. 4, 5 is optimized by alternating the minimization over $f \in \mathcal{H}_K$ and $\mathbf{p} \in [0,1]^u$ respectively. Fixing $\mathbf{p}$, the minimization over f can be done by standard techniques for solving weighted regularization problems. Fixing f, the minimization over $\mathbf{p}$ can also be done easily as described below. While the original problem is non-convex, keeping one block of variables fixed yields a

convex optimization problem over the other block of variables. Both these convex problems can be solved exactly and efficiently. An additional advantage of such block optimization is that it allows efficient algorithms for training kernel classifiers to be used directly within the deterministic annealing procedure. We note that a similar alternating optimization scheme was proposed in (Gartner et al., 2005) in the context of semi-supervised logistic regression. We now provide some details.

Optimizing f for fixed $\mathbf{p}$

By the Representer theorem, the minimizer over $f \in \mathcal{H}_K$ of the objective function in Eqn. 4 for fixed $\mathbf{p}$ is given as:

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{u} \alpha_{l+j} K(\mathbf{x}, \mathbf{x}'_j) \quad (6)$$

The coefficients $\boldsymbol{\alpha} = (\alpha_1 \dots \alpha_{l+u})$ can be computed by solving a finite dimensional optimization problem that arises by substituting this expression in Eqn. 4 where the norm $\|f\|_K^2 = \boldsymbol{\alpha}^T K \boldsymbol{\alpha}$. The resulting objective function in $\boldsymbol{\alpha}$ is denoted as $\mathcal{J}_T(\boldsymbol{\alpha}, \mathbf{p})$. Below we explicitly write down the solutions for two common choices of loss functions. One can also solve for $\boldsymbol{\alpha}$ using other optimization techniques and for other choices of loss functions.

Regularized Least Squares (RLS)
For Regularized Least squares (RLS), $V(t) = (1-t)^2/2$. Setting $\nabla_{\boldsymbol{\alpha}} \mathcal{J}_T(\boldsymbol{\alpha}, \mathbf{p}) = 0$ and solving for $\boldsymbol{\alpha}$ we obtain:

$$\boldsymbol{\alpha} = (G + \lambda C)^{-1} Y \quad (7)$$

where $Y = [y_1 \dots y_l, (2p_1 - 1) \dots (2p_u - 1)]^T$, G is the gram matrix over the $l+u$ points, C is a diagonal matrix whose first l diagonal entries are l and remaining u diagonal entries are u/λ'. In Eqn. 7, note that the matrix $(G + \lambda C)^{-1}$ is independent of $\mathbf{p}$ and therefore needs to be computed only once. Subsequent updates in the iteration only involve matrix vector products. Thus, if the inverse $(G + \lambda C)^{-1}$ can be formed, the updates for $\boldsymbol{\alpha}$ in the deterministic annealing iterations are very cheap.

SVM with Quadratic Hinge Loss
The quadratic hinge loss function is given by $V(t) = \max(0, 1-t)^2/2$. We apply the primal finite newton methods from (Keerthi & DeCoste, 2005; Chapelle, 2006) to solve Eqn. 4 with this loss. A sequence of candidate solutions $\{\boldsymbol{\alpha}^{(k)}\}$ is generated as follows. For any $\boldsymbol{\alpha}^{(k)}$ in the sequence, denote the output, as given by Eqn. 6, on any example $\mathbf{x}$ as $f^{(k)}(\mathbf{x})$ and define the following index sets: $i_0 = \{i : y_i f^{(k)}(\mathbf{x}_i) < 1\}$, $i_1 = \{j : f^{(k)}(\mathbf{x}'_j) \leq -1\}$, $i_2 = \{j : f^{(k)}(\mathbf{x}'_j) \geq 1\}$, and $i_3 = \{j : |f^{(k)}(\mathbf{x}'_j)| < 1\}$.

Consider the following objective function in the variable $\boldsymbol{\alpha}$:

$$\mathcal{J}^{(k)}(\boldsymbol{\alpha}) = \frac{\lambda}{2}\boldsymbol{\alpha}^T K \boldsymbol{\alpha} + \frac{1}{l}\sum_{i_0} V_s\left(y_i f(x_i)\right)$$
$$+\frac{\lambda'}{u}\left[\sum_{i_1} p_j V_s\left(f(\mathbf{x}'_j)\right) + \sum_{i_2}(1-p_j)V_s\left(-f(\mathbf{x}'_j)\right)\right.$$
$$\left. + \sum_{i_3} p_j V_s\left(f(\mathbf{x}'_j)\right) + (1-p_j)V_s\left(-f(\mathbf{x}'_j)\right)\right]$$

where $V_s(t) = (1-t)^2/2$. This objective function is a local quadratic approximation of the objective function Eqn. 4 and simply involves squared loss terms. Denote $\bar{\boldsymbol{\alpha}} = \text{argmin}_{\boldsymbol{\alpha}} \mathcal{J}^{(k)}(\boldsymbol{\alpha})$. This can be computed by solving a linear system that arises by setting $\nabla_{\boldsymbol{\alpha}} \mathcal{J}^{(k)}(\boldsymbol{\alpha}) = 0$. Finally, obtain $\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} + \delta^\star(\bar{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^{(k)})$ where the step length $\delta^\star$ is obtained by performing an exact line search by solving the one-dimensional problem $\delta^\star = \text{argmin}_{\delta>0} \mathcal{J}_T(\boldsymbol{\alpha} + \delta(\bar{\boldsymbol{\alpha}^{(k)}} - \boldsymbol{\alpha}^{(k)}), \mathbf{p})$. This can be done using efficient recursive updates as outlined in (Keerthi & DeCoste, 2005). From the arguments in (Keerthi & DeCoste, 2005), it can be shown that the sequence $\{\boldsymbol{\alpha}^k\}$ starting from any initial point converges in a finite number of steps to the minimizer (in $\boldsymbol{\alpha}$) of $J_T(\boldsymbol{\alpha}, \mathbf{p})$ for a given fixed $\mathbf{p}$. By starting the optimization from the solution of the previous DA iteration ("seeding"), the convergence can be very fast.

Large Scale Implementations
In the case of linear kernels, instead of using Eqn. 6 we can directly write $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ where updates for the weight vector $\mathbf{w}$ are obtained by the finite Newton procedure outlined above. For large scale problems such as text classification where $(l+u)$ as well as the dimension of $\mathbf{x}$ are possibly large and the data matrix consisting of the $\mathbf{x}_i$ has only a small fraction of nonzero elements, effective conjugate gradient schemes can be used to implement the updates for $\mathbf{w}$. The result is an impressively fast algorithm for such problems. See (Sindhwani & Keerthi, 2006) for full details.

Optimizing $\mathbf{p}$ for fixed f

For the latter problem of optimizing $\mathbf{p}$ for a fixed f, we construct the Lagrangian: $\mathcal{L} = \mathcal{J}_T(f, \mathbf{p}) - \nu(\frac{1}{u}\sum_{j=1}^{u} p_j - r)$. Solving $\partial\mathcal{L}/\partial p_j = 0$, we get:

$$p_j = \frac{1}{1 + e^{\frac{g_j - \nu}{T}}} \quad (8)$$

where $g_j = \lambda'[V(f(\mathbf{x}'_j)) - V(-f(\mathbf{x}'_j))]$. Substituting this expression in the balance constraint in Eqn. 5, we get a one-dimensional non-linear equation in ν: $\frac{1}{u}\sum_{j=1}^{u} \frac{1}{1+e^{\frac{g_i-\nu}{T}}} = r$. The root is computed exactly by using a hybrid combination of Newton-Raphson iterations and the bisection method together with a carefully set initial value.

For a fixed T, the alternate minimization of $f \in \mathcal{H}_K$ and $\mathbf{p}$ proceeds until some stopping criterion is satisfied. A natural criterion is the KL-divergence between values of $\mathbf{p}$ in consecutive iterations. The parameter T is decreased in an outer loop until the total entropy falls below a threshold. Table 1 outlines the steps for the algorithm with default parameters. In the rest of this paper, we will abbreviate our method as DA (loss) where loss is l_1 for hinge loss, l_2 for quadratic hinge loss and *sqr* for squared loss.

Table 1. Semi-supervised Learning with Deterministic Annealing.

Inputs:	$\{\mathbf{x}_i, y_i\}_{i=1}^{l}, \{\mathbf{x}'_j\}_{j=1}^{u}, \lambda, \lambda', r$
Initialize:	Set $\mathbf{p} = (r, \ldots, r) \in \mathbb{R}^u$ $\mathbf{q} = \mathbf{p}$
	Set $T = 10$ $R = 1.5$ $\epsilon = 10^{-6}$
loop1	while $S(\mathbf{p}) > \epsilon$ (S denotes entropy)
loop2	while $KL(\mathbf{p}, \mathbf{q}) > \epsilon$
	(KL denotes KL-divergence)
	Update $\boldsymbol{\alpha}$ by solving Eqn. 4
	for fixed $\mathbf{p}$
	Set $\mathbf{q} = \mathbf{p}$
	Set $\mathbf{p}$ according to Eqn. 8
	end loop1
	$T = T/R$
	end loop2
	Return $f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{u} \alpha_{l+j} K(\mathbf{x}, \mathbf{x}'_j)$

3.2. Annealing Behaviour of Loss functions

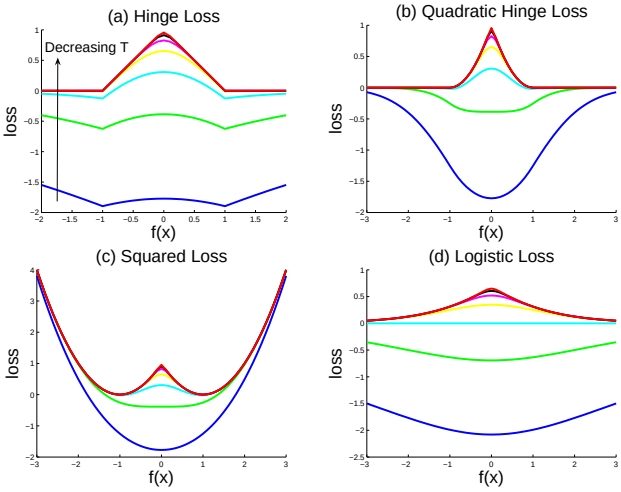
We can develop a good intuition for the working of our method by ignoring the balancing constraint in Eqn. 5 and putting together the loss terms in Eqn. 4 for a single unlabeled example $\mathbf{x}'_j$:

$$\Phi_T(f(\mathbf{x}'_j), p_j) = p_j V(f(\mathbf{x}'_j)) + (1-p_j)V(-f(\mathbf{x}'_j)) + T[p_j \log p_j + (1-p_j)\log(1-p_j)]$$

Keeping f fixed, the optimal value of p_j, say $p_j^\star$, is given by Eqn. 8 (with $\nu = 0$). The effective loss function then becomes $V'_T(f(\mathbf{x}'_j)) = \Phi_T(f(\mathbf{x}'_j), p_j^\star)$.

In Figure 2, we plot V'_T as a function of $f(\mathbf{x}'_j)$ for different settings of T. The sub-plots show the behavior of V'_T for common choices of V.

Figure 2. Annealing behavior of loss functions parameterized by T.



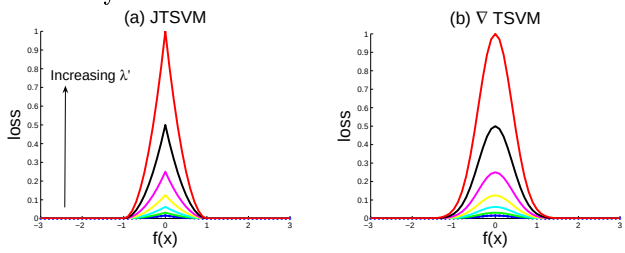
As T is decreased from high to low values, we see interestingly different behavior for different loss functions with respect to their shape in the "inner" interval $[-1, 1]$ (within the margin) and "outer" interval (outside the margin).

We see that at high values of T, the hinge loss has a sharp upward slope in the outer interval and is almost constant in the inner interval. The other loss functions are unimodal with a minimum at the decision boundary. As T is decreased, the effective loss V'_T gradually deforms into the effective loss V' in the original objective function in Eqn. 1 (also see Figure 1).

The Transductive SVM implementations of (Joachims, 1999; Chapelle & Zien, 2005) also solve a sequence of optimization problems with gradually increasing values of λ'. We refer to these implementations as JTSVM and ∇TSVM respectively. The respective effective loss functions are shown in Figure 3. We see that in all stages of the optimization, unlabeled examples in the outer interval do not influence the decision boundary. Other approaches for Transductive SVM, e.g., (Gartner et al., 2005; Collobert et al., 2005) do not discuss such an annealing component.

To examine the effectiveness of different annealing strategies and loss functions, we performed experiments on two toy datasets, 2MOONS and 2CIRCLES, with highly non-linear cluster structures. These datasets are shown in Figure 4 (a particular labeling is shown). For 10 random choices of 2 labeled examples, we recorded the number of times JTSVM, ∇TSVM and DA produced a decision boundary perfectly classifying the unlabeled data. For JTSVM and DA we report results for Hinge loss (l_1) and quadratic hinge

Figure 3. Loss functions for JTSVM and ∇TSVM parameterized by λ'.



loss (l_2).

The experiment was conducted with RBF kernels. In Table 2, we report the best performance for each method over a grid of parameters. We see that DA out-performs ∇TSVM which performs better than JTSVM. In our experiments, DA with Hinge loss succeeded in every trial for both 2CIRCLES and 2MOONS. On the other hand JTSVM failed everytime on 2MOONS and succeeded once on 2CIRCLES.

Figure 4.

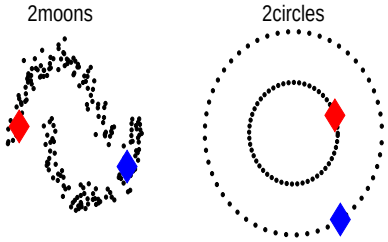


Table 2. Number of successes out of 10 trials.

Dataset → Algorithm ↓	2MOONS	2CIRCLES
JTSVM (l_2)	0	1
JTSVM (l_1)	0	1
∇TSVM	3	2
DA (l_2)	6	3
DA (l_1)	**10**	**10**

We believe that as the deformation proceeds, the interplay between the geometric structure of the data and the inner and outer intervals of the effective loss function is a key issue for global optimization in semi-supervised kernel machines based on Eqn. 1. In the early stages of the optimization, an ideal effective loss function should make the decision boundary sufficiently sensitive to unlabeled examples that are "far-away" so that it begins to align with the global geometry of data clusters. When only local adjustments need to be done, the remaining optimization can succeed under weaker deformations. An interesting direction for future work is the design of general homotopies using functions other than the entropy in Eqn. 4.

Table 3. Datasets with d features, l labeled examples, u unlabeled examples, v validation examples, t test examples.

DATASET	d	l	u	v	t
USPS2	241	93	1000	32	375
COIL6	241	90	1008	30	372
PC-MAC	7511	37	1410	13	486
AUT-AVN	20707	≥45	≤35543	-	35587
ESET2	617	33	1305	12	1350

4. Empirical Results

We present an experimental study on a collection of 5 datasets listed in Table 3. USPS2 is a subset of the USPS dataset with images of digits 2 and 5. COIL6 is a 6-class dataset derived from a collection of images of objects viewed from different angles. PC-MAC is a subset of the 20-newsgroup text dataset posing the task of categorizing newsgroup documents into two topics: *mac* or *windows*. AUT-AVN is a large-scale text dataset concerning binary categorization of UseNet articles in *auto* or *aviation* classes. Finally, ESET2 is a subset of the ISOLET dataset consisting of acoustic features of isolated spoken utterances of 9 confusible letters $\{B, C, D, E, G, P, T, V, Z\}$. We considered the binary classification task of separating the first 4 letters from the rest.

EXPERIMENTAL PROTOCOL

Datasets were split into subsets of labeled, unlabeled, validation and test examples. The sizes of these subsets are recorded in Table 3. Results presented in Tables 4, 5 are averaged over 10 random splits. For each method compared, we set $\lambda' = 1$ and recorded results on each split over the parameter grid defined by $\lambda = 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$ and widths for RBF kernels in the range $\sigma = 1/8, 1/4, 1/2, 1, 2, 4, 8$ (relative to a default value based on pairwise distances between examples in the dataset).

To focus our study on quality of optimization and degree of sensitivity to local minima, we chose to construct stratified splits so that each algorithm compared was provided an accurate estimate of class ratios. Parameters were chosen with respect to performance on the validation set for each split. Since model selection in semi-supervised settings with very few labels can often be unreliable and is largely considered to be an open issue, in Tables 4, 5 we also record the minimum of the mean error rate achieved over the pa-

Table 4. Comparison between SVM, JTSVM,∇TSVM and DA (all with quadratic hinge loss (l_2)). For each method, the top row shows mean error rates with model selection; the bottom row shows best mean error rates. u/t denotes error rates on unlabeled and test sets. Also recorded in performance of DA with squared loss (*sqr*).

	USPS2	COIL6	PC-MAC	ESET2
	u/t	u/t	u/t	u/t
SVM	8.0/8.2	22.9/23.5	21.1/20.0	20.9/21.8
	7.5/7.8	21.5/21.9	18.9/17.9	19.4/19.7
JTSVM	8.8/8.0	21.4/22.8	14.1/11.9	**10.4/10.5**
	7.6/7.2	19.9/21.2	10.4/7.0	9.2/8.9
∇TSVM	7.5/7.7	25.0/24.9	**7.6/6.9**	12.2/12.7
	6.9/7.1	21.4/21.6	5.4/**4.5**	8.7/9.1
DA	**6.5/6.6**	**15.6/16.4**	11.8/9.4	10.8/10.7
(l_2)	**6.4/6.3**	**13.6/15.0**	**5.3**/4.8	**8.1/8.5**
DA	7.3/7.1	16.5/16.7	11.8/9.4	11.6/11.3
(*sqr*)	5.7/6.3	13.8/15.2	5.4/4.7	9.0/9.4

rameter grid. This experimental setup neutralizes any undue advantage a method might receive due to different sensitivities to parameters, class imbalance issues and shortcomings of the model selection protocol.

Comparing DA, JTSVM and ∇TSVM

Table 4 presents a comparison between DA, JTSVM and ∇TSVM. The baseline results for SVM using only labeled examples are also provided. Being a gradient descent technique, ∇TSVM requires loss functions to be differentiable; the implementation in (Chapelle & Zien, 2005) uses the l_2 loss function over labeled examples and an exponential loss function over unlabeled examples. The results in Table 4 for DA and JTSVM were also obtained using the l_2 loss. Thus, these methods attempt to minimize very similar objective functions over the same range of parameters.

We see that DA is the best performing method on USPS2VS5 and COIL6. The performance improvement with DA is particularly striking on the COIL6 dataset where the TSVM and ∇TSVM performance falls below the SVM baseline. Since this dataset consists of images of 100 objects randomly grouped into 6 classes, each class is expected to be composed of several clusters. Gaps in the data space for points within the same class probably result in many local minima. The same observations do not hold to the same extent for the ESET2 dataset where the two classes are composed of acoustic sub-clusters. Here, all methods seem to perform similarly though DA returns the best mean performance over the parameter grid. On the PC-MAC text dataset, DA and ∇TSVM out-perform JTSVM. The difference between DA and ∇TSVM is found to be minor in terms of best performance achieved on this dataset. We also observe that these methods also yield good out-of-sample extension on the test set.

Large Scale Text Classification

For the large scale dataset AUT-AVN, we used the primal linear methods of (Keerthi & DeCoste, 2005) for implementing DA and JTSVM (Sindhwani & Keerthi, 2006). Learning curves with respect to varying amount of labeled data were generated and averaged over 10 random data splits keeping fixed parameter settings of $\lambda = 0.001$ and $\lambda' = 10$.

In the top sub-plot of Figure 5, we show the minimum value of the objective function achieved using DA and JTSVM with quadratic hinge loss on the AUT-AVN dataset as a function of number of labeled examples. We see that DA performs significantly better optimization over the entire range of amount of labeled data. The middle plot shows the corresponding test error rate plots. Whereas, DA shows useful improvements over JTSVM in terms of generalization performance, these improvements are not as much as one might expect given that DA finds significantly better solutions. We conjecture that on many textual problems the objective function indeed has many local minima, but lower values of it need not necessarily correspond to significantly lower values in generalization performance.

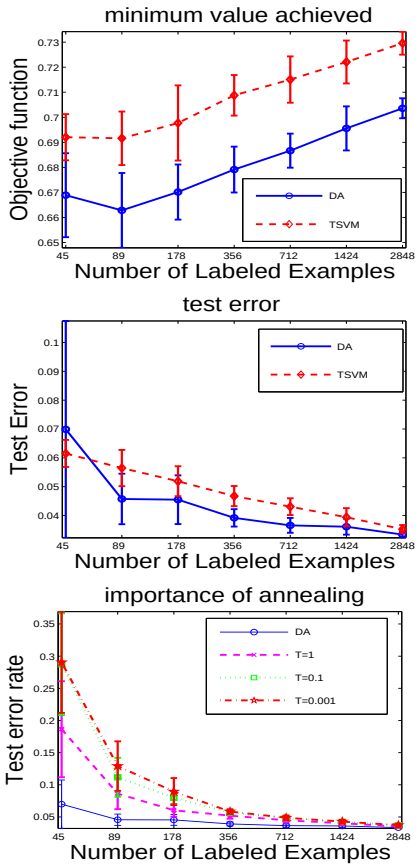
Table 5. Importance of Annealing: DA versus fixed T (no annealing) optimization. For each method, the top row shows mean error rates with model selection; the bottom row shows best mean error rates. u/t denotes error rates on unlabeled and test sets.

	USPS2	COIL6	PC-MAC	ESET2
	u/t	u/t	u/t	u/t
DA-	**6.5/6.6**	**15.6/16.4**	11.8/9.4	10.8/10.7
	6.4/6.3	**13.6/15.0**	**5.3**/4.8	8.1/8.5
T=0.1	8.5/8.4	23.9/23.3	12.6/9.9	**8.1/8.2**
	6.6/6.8	20.0/21.0	5.7/**4.7**	**7.8/8.0**
T=0.01	8.9/8.2	22.2/23.3	17.1/12.7	12.9/12.0
	7.6/7.0	20.1/21.3	7.1/5.7	8.1/8.5
T=0.001	9.0/8.1	23.6/24.4	18.6/13.0	13.5/12.5
	7.9/7.2	20.3/21.5	9.1/7.3	8.8/8.8

Importance of Annealing

In Table 5, we show the results obtained with DA for three fixed values of T with no annealing. We see that in most cases, fixed T optimization performs worse than optimization with annealing (gradually decreasing T from high to low values). On COIL, the performance drop is very significant implying that annealing may be critical for hard problems. Cases where fixed T optimization out-performed optimization with anneal-

Figure 5. AUT-AVN: Large Scale Text Classification



ing are shown in bold. However, even in these cases, annealing actually achieved a lower value of the objective function but this did not correspond to lower error rates. In the bottom plot of Figure 5, we see the learning curves for fixed T optimization and DA. On this dataset, annealing gives significantly better results.

Performance with Squared Loss

In Table 4 we see that results obtained with the squared loss are also highly competitive with other methods on real world semi-supervised tasks. This is not surprising given the success of the regularized least squares algorithm for classification problems.

5. Conclusion

We have proposed a framework based on deterministic annealing for global optimization in semi-supervised kernel machines. This framework leads to efficient algorithms that out-perform competing methods for Transductive SVMs. We have demonstrated the importance of annealing for semi-supervised learning.

Acknowledgements

Vikas Sindhwani would like to thank Partha Niyogi for discussions and support.

References

Bennett, K. & Demirez, A. (1999). *Semi-supervised Support Vector Machines* NIPS 13.

Bilbro, G., Snyder, W. E & Mann, R. (1991). *Mean-field approximation minimizes relative entropy*, Journal of Optical Society of America A, vol. 8, No 2.

Blake, A. & Zisserman, A. (1987). *Visual Reconstruction*, MIT Press.

Collobert, R., Weston, J. & Bottou, L. (2005). *Trading Convexity for Scalability*, NIPS 19.

Chapelle, O. (2006). *Training a Support Vector Machine in the Primal*, Tech. report, MPI, Tuebingen.

Chapelle, O., Chi, M. & Zien, A. (2006). *A Continuation Method for Semi-Supervised SVMs*, submitted to ICML 2006.

Chapelle, O. & Zien, A. (2005). *Semi-Supervised Classification by Low Density Separation*, AI & Statistics.

Dunlavy, D. M & O'Leary, D. P. (2005). *Homotopy Optimization methods for Global Optimization*, Tech. Report, CS-TR-4773, Univ. of Maryland.

Gartner, T., Le, Q. V., Burton, S., Smola, A. J., & Vishwanathan, V. (2005). *Large Scale Multiclass Transduction*, NIPS 19.

Hofmann, T. & Buhmann, J. M. (1997). *Pairwise Data Clustering by Deterministic Annealing*, IEEE TPAMI, No. 1, pp. 1–14.

Joachims T. (1999). *Transductive Inference for Text Classification using Support Vector Machines*, ICML.

Keerthi, S. S., & DeCoste, D. (2005) *A Modified Finite Newton Method for Fast Solution of Large Scale Linear SVMs*, JMLR, vol. 6, pp. 341–361

Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). *Optimization by Simulated annealing*, Science, vol. 220, pp. 671-680.

Nocedal, J. & Wright, S. J. (2000). *Numerical Optimzation*, Springer.

Rose, K. (1998). *Deterministic annealing for clustering, compression, classification, regression, and related optimization problems*, Proc. of IEEE, vol. 80, no. 11, pp. 2210–2239

Sindhwani, V. & Keerthi, S. S. (2006). *Large Scale Linear Semi-supervised SVMs*, ACM SIGIR.

Vapnik, V. & Sterin, A. (1977). *On structural risk minimization or overall risk in a problem of pattern recognition*, Automation and Remote Control, vol. 10(3), pp. 1495-1503.