

---

# Explanation-Augmented SVM: an Approach to Incorporating Domain Knowledge into SVM Learning

---

Qiang Sun  
Gerald DeJong

QIANGSUN@UIUC.EDU  
DEJONG@CS.UIUC.EDU

Computer Science Department, University of Illinois at Urbana-Champaign, 201 N. Goodwin, Urbana, IL 61801 USA

## Abstract

We introduce a novel approach to incorporating domain knowledge into Support Vector Machines to improve their example efficiency. Domain knowledge is used in an Explanation Based Learning fashion to build justifications or explanations for why the training examples are assigned their given class labels. Explanations bias the large margin classifier through the interaction of training examples and domain knowledge. We develop a new learning algorithm for this Explanation-Augmented SVM (EA-SVM). It naturally extends to imperfect knowledge, a stumbling block to conventional EBL. Experimental results confirm desirable properties predicted by the analysis and demonstrate the approach on three domains.

## 1. Introduction

Prior knowledge has the potential to greatly aid machine learning. A concept can be more confidently selected from a hypothesis space when evidence from the training set is combined with prior knowledge of the world.

There are two sorts of prior knowledge. The first, which we call *Solution Knowledge*, concerns the target of the learning itself and is specific to the learning task at hand. The structure of a Bayes net, for example, directly influences which distributions can be acquired by the learner. The kernel function of a support vector machine, the topology of a neural network, and the vocabulary of splits from which a decision tree is to be built all exemplify solution knowledge. The other sort, which we call *Domain Knowledge*, describes objects of the world. Some of these objects may participate directly or indirectly in the learning task, but domain knowledge is not task specific. For example, one may believe when classifying images of handwritten letters that the input pixels arise from strokes of a writing implement, and that

the same person and the same writing implement made all of the strokes of a particular image. The emergent patterns formed by handwritten images would indeed be very different in a world where this knowledge did not hold. But the knowledge is not directly associated with any particular classification task. It does not affect, for example, whether a sample image should be classified as an “H” or “N.”

Solution knowledge is easily integrated into the machine learning process as bias, but we believe its utility is limited compared to domain knowledge. Domain knowledge, while more difficult to employ, is generally more reliable and more easily articulated by a human expert. This is because the domain expert need not also possess expertise about the machine learning techniques or about the particular learning task at hand.

The main contribution of our research is to illustrate a new approach that combines Explanation-Based Learning (EBL) with statistical learning to dynamically integrate domain knowledge, examples, and the learning mechanism. Our version of EBL can be viewed as a process of inferentially transforming examples and domain knowledge into solution knowledge tailored to the learning mechanism and the learning task at hand. By insulating the domain expert from the eccentricities of a particular task, the expert is no longer forced to distort his expertise into the bias vocabulary of the learning mechanism. More importantly, the process can import pre-existing knowledge bases into the learning task.

In this paper, we demonstrate how to use EBL to incorporate domain knowledge into Support Vector Machines (SVM). An *explanation* in EBL explicitly specifies which properties of a training example are relevant and how those properties fit together to warrant the prescribed classification label. In conventional EBL an explanation is logically entailed; for us, it is only a statistical conjecture. In this study, we introduce a simple notion of *generalized* or *explained* examples to let SVMs treat them much as it treats conventional examples. In explained examples, only the important features are allowed to contribute to the kernel computation. Given an original example  $x$ , and a subset of important features  $e \subseteq x$  from an explanation, the explained example  $v$  is constructed thus:

---

Appearing in *Proceedings of the 22<sup>nd</sup> International Conference on Machine Learning*, Bonn, Germany, 2005. Copyright 2005 by the author(s)/owner(s).

$$\begin{cases} v_i = x_i, & \text{if } x_i \in e \\ v_i = '*', & \text{otherwise} \end{cases}$$

The special symbol “\*” indicates that this feature does not participate in the inner product evaluation. With numerical features one can simply use the value zero.

An explained example can be viewed as a generalization of an original example, in the sense that examples that satisfy the same explanations merit the same label for the same reasons and thus should be treated equivalently by the learner. Consider an SVM’s linear separator in its high-dimensional feature space (Vapnik, 1998). In the ideal case, an explanation is a lower dimensional linear surface to which the correct classifier *should be parallel*. To see why, consider the extensional definition of the explanation which is the set of all examples that satisfy the explanation’s requirements. Assuming the ideal case, the SVM’s feature space and linear separator are adequate to capture all of the relevant distinctions and relations. All of the examples that merit this label for the same reasons should be treated identically. In the high dimensional feature space they should have the same margin from the correct classifier. This means that they fall on a parallel linear surface. This surface will be of a lower dimension if there are any redundancies or irrelevancies in the high dimensional feature space with respect to this explanation. Such explanations constrain the correct classifier, and therefore, once discovered, can guide the learner.

But the situation will likely never be ideal. Our simple explanations can only express relevance or irrelevance of input features. In addition, there may be noise in the training examples resulting in spurious explanations. The space, despite its high dimension, may not adequately capture all of the relevant distinctions. The domain knowledge itself will almost certainly be flawed so that the explanations will vary in their veracity and in their information content. This, incidentally, is a major limitation of the conventional formalizations of EBL (Mitchell, 1997; Russell & Norvig, 2003). The present research does not fit into these conventional views. While our explanations also carry evidence about classification labels, they are not forced to carry the overwhelming evidence of a logical proof.

The constructed explanations are treated as preferences or soft constraints rather than hard constraints on the correct classifier. We blend their effect on the SVM classifier with the conventionally-treated training set. We call the result an Explanation-Augmented Support Vector Machine or EA-SVM.

We show formally that even the simple notion of explanation advanced here leads to improved classification performance. The analysis of our formulation is analogous to (and quite compatible with) the treatment of soft margin SVMs. The analysis leads to three predictions: 1) explanations will help more in difficult learning problems than in easy ones; 2)

improvement will be graduated, with more accurate domain knowledge helping more than less accurate domain knowledge; and 3) even entirely inaccurate domain knowledge should not result in EA-SVM behavior that is unduly worse than the performance of a conventional SVM with training examples alone. Through a series of experiments, we demonstrate these properties empirically using three domains.

## 2. Explanation-Augmented SVM Classification

We first discuss the ideal case of perfect explanations and correctly labeled, separable data. Then we show that imperfect explanations can be realized using slack variables. These combine straightforwardly with standard slack variables from the treatment of non-separable data.

### 2.1 Perfect Explanations

Ideally, the learned classifier evaluates the original example and the generalized example to the same value:

$$w \cdot x_i + b = w \cdot v_i + b \text{ or equivalently } w \cdot (x_i - v_i) = 0$$

Geometrically, this requires the classifier hyper-plane to be parallel to the direction  $x_i - v_i$ . We call these *parallel constraints*. The SVM quadratic problem becomes:

$$\text{minimize : } \frac{1}{2} \|w\|^2$$

$$\text{subject: to : } y_i(w \cdot x_i + b) - 1 \geq 0, \forall i \text{ and } w \cdot x_i - w \cdot v_i = 0, \forall i$$

This is a problem similar to the standard SVM optimization problem and can be solved by the method of Langrange multipliers. The primal Langrangian is:

$$L_P \equiv \frac{1}{2} \|w\|^2 - \sum_i \alpha_i y_i (w \cdot x_i + b) + \sum_i \alpha_i - \sum_i \lambda_i (w \cdot x_i - w \cdot v_i)$$

Setting the derivatives w.r.t. the primal variables to zero, we have:

$$w = \sum_i \alpha_i y_i x_i + \sum_i \lambda_i (x_i - v_i) \quad \text{and} \quad \sum_i \alpha_i y_i = 0$$

Substituting into  $L_P$ , the dual problem becomes maximizing (w.r.t. the  $\alpha_i, \lambda_i$ ):

$$\begin{aligned} L_D \equiv & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j + \sum_{i,j} \alpha_i \lambda_j \lambda_i \cdot (x_j - v_j) \\ & - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j (x_i - v_i) \cdot (x_j - v_j) \end{aligned}$$

$$\text{subject to: } \alpha_i \geq 0, \quad \sum_i \alpha_i y_i = 0$$

This is a standard form of the quadratic programming (QP) problem w.r.t the variable vector composed by  $\alpha_i$ , and  $\lambda_i$ . The quadratic term is no longer the original  $n \times n$  kernel matrix but a  $2n \times 2n$  matrix in which we employ a single explanation for each training example. After computing this quadratic matrix, we can use a standard QP solver for the above optimization problem.

Notice that, if we fail to build an explanation for a particular example, all input features are treated as important, therefore,  $v_i=x_i$ , and  $v_i-x_i=0$ . In such case, the corresponding variable in the QP problem can be simply eliminated. With no explanations the problem reduces to a standard SVM.

## 2.2 Imperfect Explanations

If the domain knowledge is imperfect, the constraints cannot all be met. The explained examples should then be treated as a bias to be respected as much as possible. This is similar to the standard SVM algorithm for the non-separable case (Vapnik, 1998). New slack variables ( $\delta_i$ ) measure the difference between the evaluations of the original examples and the generalized examples:

$$\forall i \quad w \cdot x_i - w \cdot v_i \geq -\delta_i, \quad w \cdot x_i - w \cdot v_i \leq \delta_i, \quad \delta_i \geq 0$$

To penalize violations of the constraints, we change the objective function from  $\|w\|^2/2$  to  $\|w\|^2/2 + Q\sum_i \delta_i$ ; we call  $Q$  the *confidence parameter*. It reflects confidence in (or assessed quality of) the domain knowledge. It will be set automatically. A larger  $Q$  corresponds to better knowledge and a greater penalty for disagreeing with the explanations. Now our primal problem becomes:

$$\begin{aligned} \text{Minimize:} \quad & \frac{1}{2}\|w\|^2 + Q\sum_i \delta_i \\ \text{sub:} \quad & y_i(w \cdot x_i + b) - 1 \geq 0, \forall i; \quad w \cdot x_i - w \cdot v_i \geq -\delta_i, \forall i; \\ & w \cdot x_i - w \cdot v_i \leq \delta_i, \forall i; \quad \delta_i \geq 0, \forall i \end{aligned}$$

with the Lagrangian:

$$\begin{aligned} L_P \equiv & \frac{1}{2}\|w\|^2 + Q\sum_i \delta_i - \sum_i \alpha_i y_i (w \cdot x_i + b) + \sum_i \alpha_i \\ & - \sum_i \beta_i (w \cdot x_i - w \cdot v_i + \delta_i) - \sum_i \gamma_i (-w \cdot x_i + w \cdot v_i + \delta_i) - \sum_i \mu_i \delta_i \\ \text{sub:} \quad & \alpha_i \geq 0, \quad \sum_i \alpha_i y_i = 0, \quad \beta_i \geq 0, \quad \gamma_i \geq 0, \quad \mu_i \geq 0 \end{aligned}$$

Requiring that the gradient of  $L_P$  with respect to  $w$ ,  $b$  and  $\delta_i$  vanish yields:

$$\begin{aligned} \frac{\partial L_P}{\partial w} &= w - \sum_i \alpha_i y_i x_i - \sum_i (\beta_i - \gamma_i)(x_i - v_i) = 0; \\ \frac{\partial L_P}{\partial b} &= \sum_i \alpha_i y_i = 0; \quad \frac{\partial L_P}{\partial \mu_i} = Q - \beta_i - \gamma_i - \mu_i = 0 \end{aligned}$$

Substituting into the Lagrangian formulation  $L_P$ , with  $\lambda_i \equiv \beta_i - \gamma_i$  we obtain the dual:

$$\begin{aligned} \text{Maximize:} \quad L_D \equiv & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ & + \sum_{i,j} \alpha_i \lambda_j y_i x_i \cdot (x_j - v_j) - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j (x_i - v_i) \cdot (x_j - v_j) \\ \text{sub:} \quad & \alpha_i \geq 0, \quad \sum_i \alpha_i y_i = 0, \quad -Q \leq \lambda_i \leq Q \end{aligned}$$

This is a QP problem with the same solution as the perfect explanation case,  $w = \sum_i \alpha_i y_i x_i + \sum_i \lambda_i (x_i - v_i)$ , except the

$\lambda_i$  are now bounded by  $Q$ . If we have perfect explanations, then  $Q$  and the  $\lambda_i$  are unbounded, and the problem reduces to the ideal case. Conversely, if  $Q$  and the  $\lambda_i$  are 0, the problem ignores the explanations and reduces to a standard SVM.

This is easily combined with slack variables for non-separable data: slack variables  $\xi_i$  are introduced to penalize the errors:  $y_i(x_i \cdot w + b) \geq 1 - \xi_i, \forall i; \quad \xi_i \geq 0, \forall i$ ; the objective function now becomes:  $\|w\|^2/2 + Q\sum_i \delta_i + C\sum_i \xi_i$ . The resulting maximization is identical except that the first constraint  $\alpha_i \geq 0$  becomes  $0 \leq \alpha_i \leq C$ .

EA-SVMs can be solved by the standard SVM methods since the QP problem is convex:

**Theorem 1: The EA-SVM QP problem is convex.**

Proof: It is sufficient that the expression  $L_D$  of our EA-SVM is quadratic and positive semi-definite:

$$\begin{aligned} & \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ & - \sum_{i,j} \alpha_i \lambda_j y_i x_i \cdot (x_j - v_j) + \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j (x_i - v_i) \cdot (x_j - v_j) \\ & = \frac{1}{2} \langle \sum_i (\alpha_i y_i x_i) - \sum_j \lambda_j (x_j - v_j), \sum_i (\alpha_i y_i x_i) - \sum_j \lambda_j (x_j - v_j) \rangle \\ & = \frac{1}{2} \left\| \sum_i (\alpha_i y_i x_i) - \sum_j \lambda_j (x_j - v_j) \right\|^2 \\ & \geq 0 \end{aligned}$$

Where  $\langle \dots \rangle$  denotes the inner product. □

## 2.3 Setting the Confidence Parameter $Q$

In our implementation, the confidence parameter  $Q$  is set by cross validation. Note that  $Q$  bounds the extra variables  $\lambda$  in the solution. When  $Q$  is sufficiently small,  $\lambda$  are much smaller than  $\alpha$  and an EA-SVM solution will be similar to standard SVM solution. Conversely, if  $Q$  is sufficiently large,  $\lambda$  is much larger than  $\alpha$  and the explanations will dominate. Therefore, we choose the initial candidate set for  $Q$  with respect to value of  $\alpha$ . In our comparisons, we first run the standard SVM to determine the average value of  $\alpha$ . The candidate set for  $Q$  is a geometric series about this average value.

## 3. A Formal Analysis

We now provide a formal justification for our explanation-augmented SVM using the fat-shattering dimension. We also identify qualitative predictions to be empirically tested which will insure that the formalization addresses significant and observable phenomena.

### 3.1 EA-SVM with Hard Constraints

The fat-shattering dimension measures the expressiveness of a hypothesis space. The parallel constraints from our

explanations should further restrict the expressiveness yielding an easier learning problem. Consider a class of linear functions  $F$  of norm less than or equal to  $B$  restricted to the examples in the sphere of radius  $R$  about the origin. The fat-shattering dimension of  $F$  is bounded by  $\text{fat}_F(\gamma) \leq (BR/\gamma)^2$ . See Bartlett and Shawe-Taylor (1999) for details. To this we add parallel constraints:

**Theorem 2 *Fat-shattering of linear functions with parallel constraints.*** Consider a Hilbert space and one class of linear functions  $F$  of norm less than or equal to  $B$  satisfying the following constraints:  $\forall i, f(x_i) = f(v_i)$ , where  $v_i$  is the explained example of  $x_i$ , let  $R_V$  denote the radius of the ball that contains all  $v_i$ , then the fat shattering dimension of  $F$  can be bounded by  $\text{fat}_F(\gamma) \leq (BR_V/\gamma)^2$ .

**Proof:** For every  $f \in F$ , where  $f = w \cdot x + b$ , we define a linear function  $g \in G: V \mapsto \{0,1\}$  where  $g(v) = w \cdot v + b$ . The norm of functions  $g \in G$  is the same as  $f \in F$ , but they are restricted to the examples in the sphere of radius  $R_V$ . Bartlett and Shawe-Taylor tell us the fat-shattering dimension of  $G$  is bounded by  $\text{fat}_G(\gamma) \leq (BR_V/\gamma)^2$ .

Next we show that  $F$  has the same fat-shattering dimension as  $G$ . First observe that  $V \subseteq X$ , therefore if a set of points  $V^m = \{v_1, v_2, \dots, v_m\}$  is shattered by  $G$ , they are also shattered by  $F$ . Therefore  $\text{fat}_F(\gamma) \geq \text{fat}_G(\gamma)$ . Now consider the explanations as a many-to-one mapping  $m: X \mapsto V$ , then  $f(x) = g(m(x))$ . Therefore if a set of points  $X^m = \{x_1, x_2, \dots, x_m\}$  is shattered by  $F$ , then the set  $V^m = \{m(x_1), m(x_2), \dots, m(x_m)\}$  is shattered by  $G$ . Therefore  $\text{fat}_F(\gamma) \leq \text{fat}_G(\gamma)$ ;  $\text{fat}_F(\gamma) \geq \text{fat}_G(\gamma)$  and  $\text{fat}_F(\gamma) \leq \text{fat}_G(\gamma)$  imply  $\text{fat}_F(\gamma) = \text{fat}_G(\gamma)$ .  $\square$

Applying the Theorem 3.12 in Shawe-Taylor et al. (1998) to linear classifiers with parallel constraints immediately yields the following theorem:

**Theorem 3 *Generalization error bound on linear classifiers with parallel constraints.*** Let  $S = \{x_1, x_2, \dots, x_m\}$  be a training set of size  $m$  drawn from a fixed but unknown distribution over the input space  $X$ . Let  $v_i$  be the explained example of  $x_i$ , and let  $R_V$  denote the radius of the sphere containing all  $v_i$ . Then with probability  $1 - \delta$ , the generalization error of a linear classifier  $(u, b)$  on  $X$  with  $\|u\| = 1$  that correctly classifies all examples in  $S$  with margin  $\gamma > 0$ , and satisfies the parallel constraints  $\forall i, f(x_i) = f(v_i)$ , is bounded by:

$$\epsilon(m, h, \delta) = 2(h \log_2(8em/h) \log_2(32m) + \log_2(8m/\delta))/m,$$

$$\text{where } h = \lfloor 64.5R_V^2/\gamma^2 \rfloor \leq em.$$

This bound is of the same form as the bound for standard SVM (Shawe-Taylor et al., 1998), with  $R_V$  playing the role

of  $R$ .  $R$  measures the radius of the example sphere in the original space, while  $R_V$  measure the radius of the explained example sphere. Therefore, the explanations have most to offer when the ratio  $R_V/R$  is small. This is the case when the learning problem is difficult but the domain knowledge is informative.

This yields our first two testable qualitative predictions: 1) Holding the domain knowledge constant, explanation-augmentation should benefit difficult learning problems more than easier ones. 2) Over the same learning problems, better knowledge should result in better EA-SVM performance.

### 3.2 EA-SVM with Soft Constraints

When the constraints cannot all be satisfied, we want a classifier that is as consistent with the constraints as possible. We follow the analysis of Shawe-Taylor and Cristianini (2002) for soft margins. The input space  $X$  is mapped to a higher dimensional space so that the parallel constraints can be satisfied.

Following Shawe-Taylor and Cristianini's work, we use the notion of  $L_f(X)$  to represent the set of real valued functions  $f$  on  $X$  with inner product of  $f, g \in L_f(X)$  as  $\langle f, g \rangle = \sum_{x \in \text{supp}(f)} f(x)g(x)$ . We use  $f_0$  to denote a special function such that  $\langle f_0, g \rangle \equiv 0$ .

Now we define a new inner product space  $X \times L_f(X)$ . For any fixed  $\Delta > 0$ , we embed  $X$  into  $X \times L_f(X)$  with  $\tau_\Delta: x \mapsto (x, \Delta f_0)$ , and embed  $V$  into  $X \times L_f(X)$  with  $\tau_\Delta: v \mapsto (v, \Delta \delta_v)$ , where  $\delta_v \in L_f(X)$  is defined by  $\delta_v(z) = 1$ , if  $z = v$ , and  $\delta_v(z) = 0$ , otherwise.

We augment a linear classifier  $(u, b)$  on  $X$  to a  $(\hat{u}, b)$  on  $X \times L_f(X)$ , where  $\hat{u} = (u, \sum_i (u \cdot (x_i - v_i) \delta_{v_i}) / \Delta)$ . After some algebra, we observe that the augmented classifier  $(\hat{u}, b)$  on  $X \times L_f(X)$  classifies  $\tau_\Delta(x)$  the same as  $(u, b)$  classifiers  $x$ :  $\hat{u} \cdot \tau_\Delta(x) + b = u \cdot x + b$ . Also  $(\hat{u}, b)$  satisfies parallel constraints defined by  $\tau_\Delta(x_i)$  and  $\tau_\Delta(v_i)$ :  $\hat{u} \cdot (\tau_\Delta(x_i) - \tau_\Delta(v_i)) = 0$ . Therefore, Theorem 2 provides a bound on its fat-shattering dimension, and Theorem 3 yields the bound on its error rate, which is the same as the error rate of the original classifier  $(u, b)$ .

To examine the fat-shattering dimension of the augmented classifier  $(\hat{u}, b)$ , we first observe that the additional component in  $\hat{u}$  increases the square of the norm of the classifier to  $\|\hat{u}\|^2 = \|u\|^2 + D^2/\Delta^2$ , where  $D \equiv \sqrt{\sum_i (u \cdot (x_i - v_i))^2}$ . Also the explained examples are

embedded in  $X \times L_f(X)$  by the mapping  $\tau_\Delta : v \mapsto (v, \Delta \delta_v)$ , which makes  $\|\tau_\Delta(v)\|^2 = \|v\|^2 + \Delta^2 \langle \delta_v, \delta_v \rangle = \|v\|^2 + \Delta^2$ .

Taking these adjustments into account, Theorems 2 and 3 yield the following result:

**Theorem 4 Generalization error bound of linear classifiers with soft constraints.** Fix  $\Delta > 0, b \in R$ . Randomly draw training set  $S = \{x_1, x_2, \dots, x_m\}$  of size  $m$  with a fixed but unknown probability distribution on the input space  $X$ . Let  $v_i$  be the explained example of  $x_i$ , and  $R_V$  denote the radius of the sphere containing all  $v_i$ . Then with probability  $1 - \delta$ , the generalization error of a linear classifier  $(u, b)$  on  $X$  with  $\|u\| = 1$  that correctly classifies all examples in  $S$  with margin  $\gamma > 0$  is bounded by:

$$\varepsilon(m, h, \delta) = 2(h \log_2(8em/h) \log_2(32m) + \log_2(8m/\delta))/m, \quad \text{where}$$

$$h = \left\lfloor \frac{64.5(R_V^2 + \Delta^2)(1 + D^2/\Delta^2)}{\gamma^2} \right\rfloor \leq em, \quad \text{and } D \equiv \sqrt{\sum_i (u \cdot (x_i - v_i))^2}.$$

### 3.3 Finding Linear Classifier with Soft Constraints in the Expanded Space

The preceding analysis provides a way to transform an optimization problem with non-satisfiable constraints into one with satisfiable constraints. The mapping  $\tau_\Delta$  used in the transformation implicitly defines a kernel as follows:

$$k(x, x') = \langle \tau_\Delta(x), \tau_\Delta(x') \rangle = \langle (x, \Delta f_0), (x', \Delta f_0) \rangle = x \cdot x'$$

$$k(v, v') = \langle \tau_\Delta(v), \tau_\Delta(v') \rangle = v \cdot v' + \Delta^2 \delta_v(v')$$

By using these kernels in the expanded space  $X \times L_f(X)$ , the optimization problems becomes:

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) + \sum_{i,j} \alpha_i \lambda_j y_i k(x_i, (x_j - v_j))$$

$$- \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j k((x_i - v_i), (x_j - v_j))$$

$$= \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j + \sum_{i,j} \alpha_i \lambda_j y_i x_i \cdot (x_j - v_j)$$

$$- \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j (x_i - v_i) \cdot (x_j - v_j) - \frac{1}{2} \Delta^2 \sum_i \lambda_i^2$$

Now we show that the above is exactly the dual QP problem that one would obtain by solving the following optimization problem with  $Q = 1/(2\Delta^2)$ :

$$\text{Minimize : } \frac{1}{2} \|w\|^2 + Q \sum_i \delta_i^2$$

$$\text{sub : } y_i(w \cdot x_i + b) - 1 \geq 0, \forall i; \quad w \cdot x_i - w \cdot v_i = \delta_i$$

To see this, we apply the Lagrangian variables to obtain the dual problem:

$$\text{Maximize : } L_D \equiv \sum_i \alpha_i \sum_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

$$+ \sum_{i,j} \alpha_i \lambda_j y_i x_i \cdot (x_j - v_j) - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j (x_i - v_i) \cdot (x_j - v_j) - \sum_i \lambda_i^2 / (4Q)$$

The algorithm we described in the section 2 actually

solves a closely related optimization problem, where we use 1-norm instead of 2-norm in the extra penalty term.

### 3.4 More on the Confidence Parameter $Q$

The bound in the Theorem 4 depends on a parameter  $\Delta$  that we use to define the mapping. For different  $\Delta$ , the bound stated in Theorem 4 holds, but the tightness of the bound varies. Note that the minimum of the expression for  $h$  (ignoring the constant and suppressing the denominator  $\gamma^2$ ) is  $(R + D)^2$  attained when  $\Delta = \sqrt{RD}$ . Therefore, we need to not only search for the hyper-plane, but also adjust  $\Delta$  to minimize the error bound.

One way to adjust  $\Delta$  is to choose a discrete set of values for it, and evaluate the best hyper-plane we could find given each value. This is exactly the cross-validation algorithm we used in our algorithm. To verify this, notice, as shown in section 4.3, that the parameter  $Q$  used in our algorithm and  $\Delta$  from the analysis are related:  $C = 1/(2\Delta^2)$ . Changing  $Q$  in our algorithm is just mapping examples to the expanded space using a different  $\Delta$ .

Since  $Q = 1/(2\Delta^2)$ , if the optimal  $Q$  (found by cross validation) is large, the optimal  $\Delta$  minimizing  $h$  in Theorem 4 is small. Since this optimal value is obtained when  $\Delta = \sqrt{RD}$ , it suggests that the linear classifier has a small  $D$  with respect to explanations. According to Theorem 4, such a classifier is likely to have a low error rate. Thus,  $Q$  is a measure of the domain knowledge quality and, from cross validation, it is unlikely that poor knowledge will be confused with good knowledge. This is our third qualitative prediction.

## 4. Empirical Results

We devised four empirical investigations to validate the EA-SVM method and test the predicted qualitative behaviors.

The first three experiments employ the domain of distinguishing handwritten Chinese characters. Experiment 1 demonstrates the relative advantage of the EA-SVM over an otherwise-identical conventional SVM. Experiment 2 tests the first prediction, demonstrating that explanation-augmentation helps more as problems become more difficult. Experiment 3 confirms the two predictions that EA-SVM performance improves with the quality of the domain knowledge and that even extremely misleading domain knowledge does not harm EA-SVM performance significantly measured against an identical conventional SVM. Experiment 4 demonstrates that the EA-SVM method is broadly applicable. EA-SVM improvement is shown on protein super-family classification and categorization of Reuters news articles.

We implemented the SVM and EA-SVM with Matlab using PR\_LOQO as the QP solver. The confidence parameter  $Q$  in the algorithm is set automatically by 5-fold cross-validation. The candidate set for  $Q$ , as explained in section 3.3, is chosen according to the average SVM  $\alpha$ . The soft SVM slack variable misclassification penalty,  $C$ , is set to 0.1 for all experiments.

Our primary domain concerns classifying pairs of handwritten Chinese characters. With greater than 3,000 commonly used characters the most complete database we could find (Saito et al 1985), contains just 200 examples for each character. We expect knowledge can be greatly helpful for such domains with limited training data.

Domain knowledge is specified at the level of *stroke* descriptions. For each character, we provide a prototype that describes how many strokes are in the character, whether the strokes are horizontal, vertical or slanted, from this we derive whether strokes will be connected, crossed, etc. We approximate each stroke with a line segment, and use Hough transformation to determine the significant lines in the images. This gives us with some approximate information on associating pixels with strokes. More details on building such association appear in (Sun & DeJong, 2005) which describes how specialized kernel functions can be learned using prior knowledge. With such association, building explanations is simply to select those pixels that realize the set of strokes that are essential to distinguish two training characters. The explained examples are also images with some pixels (automatically) removed. In this work, human experts provide the knowledge of which strokes are important for classification. This information can be derived from a vector font and we are currently pursuing this line.

We chose 10 characters from 3 related groups, as shown in Figure 1. The characters in the same group are highly similar, while characters between groups have little in common. This yields 45 classification problems of varying difficulties. Both learners use a conventional 3<sup>rd</sup> degree polynomial kernel function of pixel intensities. The performance is evaluated using 5-fold cross validation.

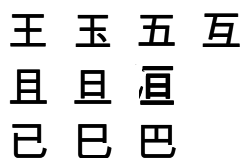


Figure 1. The Chinese characters used in the experiments

**Experiment 1: Does Explanation-Augmentation Help?**

In our first experiment, we compared EA-SVM with three conventional SVMs, which are trained on original examples, explained example and original+explained

examples respectively. The last two control conditions can viewed as naïve approaches to use explained example. The main thrust of EBL generally, and this research particularly, concerns the advantage of *interaction* between training examples and knowledge. A better control would be one that directly inputs knowledge instead of explanations to SVM training examples. Yet, it is not clear how to design such control condition; domain knowledge is the kind of knowledge that cannot be directly used by a statistical learner. The control conditions used in this experiments, nevertheless, illustrates the advantage of EA-SVM.

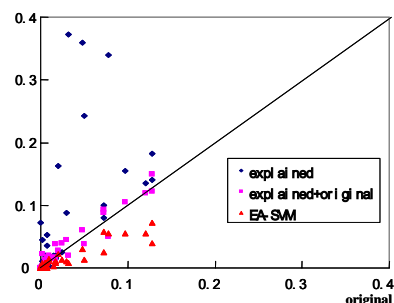


Figure 2. Comparison of SVM and EA-SVM performance on handwritten test recognition tasks.

Figure 2 shows the comparison of the average error rates in all forty-five tasks over a randomly selected training set of 320 examples; the remaining examples are used as the test set. The results are shown as a scatter plot where the horizontal axis is the conventional SVM trained on original examples, so that points falling below the 45 degree line correspond to learning problems for which EA-SVM or control SVMs outperforms the standard SVM.

The results in figure 2 show that the success of EA-SVM is due to the appropriate use of explained examples. Using only explained examples is similar to setting confidence parameter to infinity, while using both equally corresponds roughly to setting the confidence parameter to 1. EA-SVM uses cross-validation to automatically set confidence parameter using training examples. This gives us the robustness. We will briefly discuss this later.

**Experiment 2: Do Difficult Problems Benefit More?**

In Figure 2, explanations seem to be more helpful in difficult classification tasks. To further investigate this, we divided the 45 classification tasks into two difficulty conditions. In the easy condition, characters to be distinguished are drawn from different groups (33 tasks). In the difficult condition, characters from the same group must be distinguished (12 tasks). Figure 3 shows the average learning curves of these two conditions. The improvement on difficult problems is greater than easy ones at all training levels. EA-SVM and SVM performance on the easy tasks is almost indistinguishable.

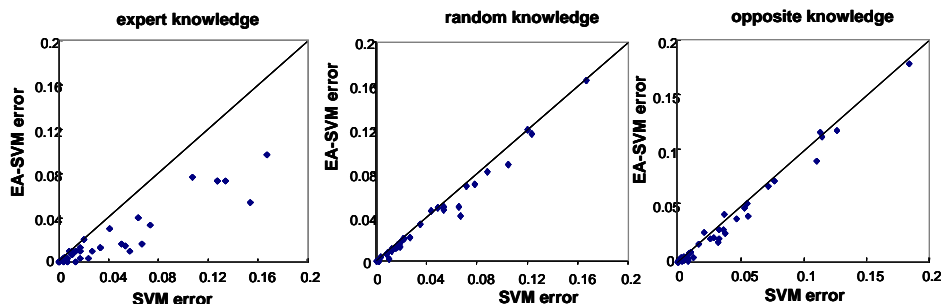


Figure 4. Effect of knowledge quality on performance of EA-SVM.

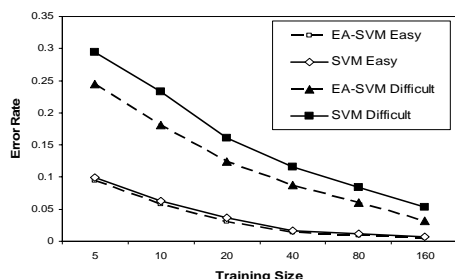


Figure 3. EA-SVM / SVM performance on easy and difficult tasks.

Next we applied Kendall’s Tau (Conover, 1980) to perform nonparametric test for the agreement between two rankings: problem difficulty and improvement afforded by EA-SVM. The Kendall’s Tau measure over these two variables is 0.798, which suggests a high correlation. More importantly, the hypothesis is accepted with greater than 99.99% confidence. We repeated the test with a number of other measures of problem difficulty and improvement. All tests show that the hypothesis is accepted at or above the 99.93% level.

### Experiment 3: The Effect of Knowledge Quality

Good but imperfect knowledge improves EA-SVM behavior over conventional SVMs. But does behavior degrade gracefully with poorer knowledge? To answer this question, we built two additional kinds of explanations: 1) in the random condition, random sets of pixels set to 0 with no regard to the original domain knowledge; 2) in the opposite condition, the complement of the original explanation is used so that important pixels are now unimportant and vice versa. Figure 4 shows the scatter plots of errors made by standard SVM and EA-SVM in these conditions. Random explanations almost never harm the performance, with some marginal improvement in certain tasks probably due to chance. It is clear that even opposite explanations do not significantly harm performance.

**Experiment 4: Performance on Protein Super-Family and Text Categorization.** How general is our approach? Is there some fortuitous match between the workings of

EA-SVMs and the task of distinguishing handwritten characters? To address this question, we examined two additional learning domains: protein super-family prediction, and topic categorization of text articles. In each domain we adopt the most standard kernel from the literature in order to exercise our approach under different kernel functions. We also adopt the accepted scoring criteria for each domain. Domain knowledge for these two domains is taken directly from available databases (the Prosite motif database and WordNet).

**Domain 1:** Proteins are assigned to functional and structural classes called *super-families* based on their amino acid sequences. We use the Structural Classification of Proteins (SCOP), a database of known 3D structures of proteins, as the data set. It contains 54 super-families and 7329 example protein sequences. We adopt the same test and training set splits, and use the same mismatch kernel function as (Leslie & Kuang, 2003). The performance of the classifiers is presented using the Receiver Operating Characteristic (ROC) score.

Our (rather flawed) domain knowledge is that a protein’s super-family is determined only by motif sequences. In reality this seems to be only a part of the answer. To build an explanation for a training example, we use its sequence to search for known motifs in the Prosite motif database. The search is performed using the tools offered by Biology WorkBench (<http://workbench.sdsc.edu>). Only motif sequence in the training examples are participated in kernel computation. This is similar to removing pixels from handwritten character recognition.

Figure 5A shows the scatter plot of ROC scores comparing SVM and EA-SVM performance in all 54 classification problems of the SCOP database. The higher the score, the better the performance. The points above the equal-line indicate that EA-SVM out-performs SVM in almost all cases.

**Domain 2:** The Reuters-21578 data set assigns category labels to Reuters news articles. To obtain a training and test set, we use the Modified Apte (“ModApte”) split, which leads to a corpus of 9603 training documents and 3299 test documents.

Our domain knowledge, provided by WordNet, is that words semantically related to the label of a category are likely to be more informative than others about an article's category. Words that are one-distance away in the WordNet from synonyms of topic words are taken as important; an explained example is the bag of important words occurring in the training example. Only those words will participate in kernel function computation. Again, this is similar to selecting important pixels in the handwritten character images.

Our experimental setup follows (Christianini et al., 2001). Learning is performed on the top 5 Reuters categories: "earn", "acquire", "money", "grain", and "crude." The kernel function is defined as a linear function over the article's bag-of-words. For evaluation of classifiers, we used the F1 performance measure.

The results are shown in Figure 5B. Again, a higher score indicates better performance; the points above the equal-line indicate that EA-SVM consistently out-performs the SVM control. EA-SVM is sometimes significantly better and never worse than SVM.

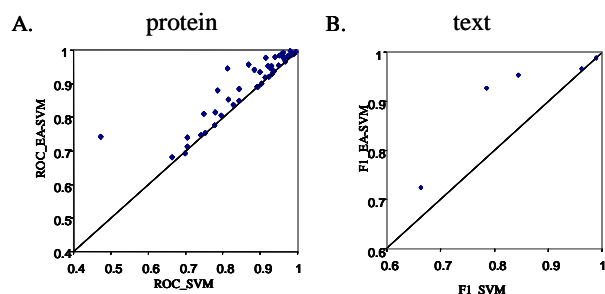


Figure 5. EA-SVM improves classification performance in the protein and text categorization domains.

## 5. Conclusion

The Explanation Based approach can be naturally extended to state-of-the-art statistical machine learners. Indeed it has much to offer by tapping additional source of information. The inferential interaction between domain knowledge and training examples sets it apart from other SVM approaches to prior knowledge (e.g., Decoste & Schoelkopf, 2002 and Fung & Shavlik, 2003). In our view, the purpose of domain knowledge is to introduce a high-level high-information vocabulary of pre-existing abstract features (such as "strokes" for handwritten characters). The explanation process, guided by the training examples, relates these high-level organizing features to input features. In this way, the classification patterns need not emerge purely empirically. Our experiments demonstrate significant improvements even though the explanations are simple and the domain knowledge is approximate and not specifically engineered for the task. This work takes a first step refocusing machine learning on the principled incorporation of prior domain knowledge.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Award NSF IIS 04-13161. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

- Bartlett, P. & Shawe-Taylor, J. (1999). Generalization performance of support vector machines and other pattern classifiers. In Schölkopf, B., Burges, C., and Smola, A. J., editors, *Advances in Kernel Methods --- Support Vector Learning*, pages 43-54, Cambridge, MA.
- Christianini, N., Shawe-Taylor, & J. Lodhi, H. (2001). Latent Semantic Kernels. *Journal of Intelligent Information Systems (JIIS)* Vol. 18(2): 127-152
- Conover, W.J., (1980) *Practical Non-Parametric Statistics*, 2nd edn. John Wiley and Sons, New York.
- Decoste D. & Schoelkopf B. (2002). Training Invariant Support Vector Machines. *Machine Learning* 46: 161-190.
- Fung, G., Mangasarian, O. & Shavlik, J. (2003). Knowledge-Based Support Vector Classifiers. *NIPS 2002*. 521-528.
- Leslie, C. & Kuang, R. (2003). Fast Kernels for Inexact String Matching. *16th Annual Conference on Learning Theory*. 114-128.
- Mitchell, T. (1997). *Machine Learning*. New York: McGraw-Hill.
- Russell S, Norvig P. (2003). *Artificial Intelligence: A Modern Approach*, Second Edition, Englewood Cliffs, New Jersey: Prentice Hall.
- Saito, T., Yamada, H. & Yamamoto K. (1985). On the data base ETL 9 of handprinted characters in JIS Chinese characters and its analysis. *IEICE Transactions*, J68-D(4):757--764.
- Shawe-Taylor, Bartlett, J., P. L., Williamson, R. C., & Anthony, M. Structural risk minimization over data-independent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926-1940, 1998.
- Shawe-Taylor, J. & Cristianini, N. (2002) On the Generalisation of Soft Margin Algorithms. *IEEE Transactions on Information Theory* 48(10): 2721-2735
- Sun Q., & DeJong, G. (2005) Feature Kernel Functions: Improving SVMs Using High-level Knowledge, (to appear) *IEEE Computer Vision and Pattern Recognition*
- Vapnik, V. N. (1998). *Statistical Learning Theory*. New York: Wiley.