Improved Generalization via Tolerant Training

W. Nick Street
Computer Science Department
Oklahoma State University
205 Mathematical Sciences, Stillwater, OK 74078
nstreet@cs.okstate.edu
405-744-6471

O. L. Mangasarian
Department of Computer Sciences
University of Wisconsin
1210 West Dayton Street, Madison, WI 53706
olvi@cs.wisc.edu

December 20, 1996

Abstract

Theoretical and computational justification is given for improved generalization when the training set is learned with less accuracy. The model used for this investigation is a simple linear one. It is shown that learning a training set with a tolerance τ improves generalization, over zero-tolerance training, for any testing set satisfying a certain closeness condition to the training set. These results, obtained via a mathematical programming formulation, are placed in the context of some well-known machine learning results. Computational confirmation of improved generalization is given for linear systems (including nine of the twelve real-world data sets tested), as well as for nonlinear systems such as neural networks for which no theoretical results are available at present. In particular, the tolerant training method improves generalization on noisy, sparse, and over-parameterized problems.

Keywords: Inductive learning, function approximation, generalization
Running head: Tolerant Training

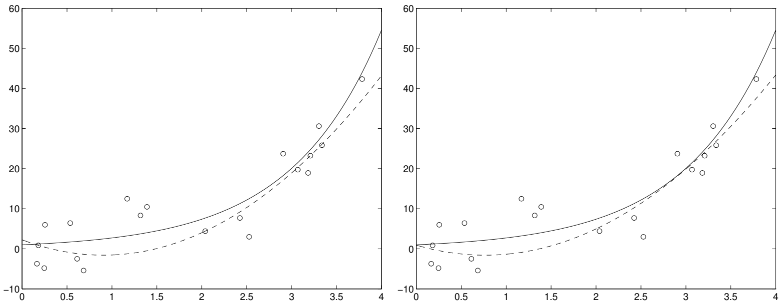
1 Introduction

It is a widely accepted heuristic in machine learning as well as in approximation theory that precisely learning a training set can lead to overfitting, and therefore to poor generalization on unseen data points. In this work we give a deterministic mathematical justification for this idea on a linear system, using the tools of mathematical programming. We also corroborate our results computationally on linear systems and show, again computationally, that they also hold for a much wider class of nonlinear problems, including neural networks.

Tolerant training is a general method for avoiding overfitting in high-variance learning situations that permits small errors in fitting the training data. In trying to fit noisy data, we attempt to obtain better generalization by assuming a "band" of errors around the measured observations. This is implemented with a parametric tolerance band of width τ surrounding the predictive surface. Points that fall into this band during training will not be considered to be in error; points outside the band have their error reduced by an amount equal to the width of the band. When $\tau = 0$, this is classical least error fit and one might expect poor generalization if the data are noisy. On the other hand, if a large tolerance τ is used, then an inaccurate fit would again result in poor generalization. Somewhere between $\tau = 0$ and large values of τ, one might expect an "optimal" τ that generalizes in a more satisfactory way. Indeed, our theoretical and computational results indicate precisely that this is true.

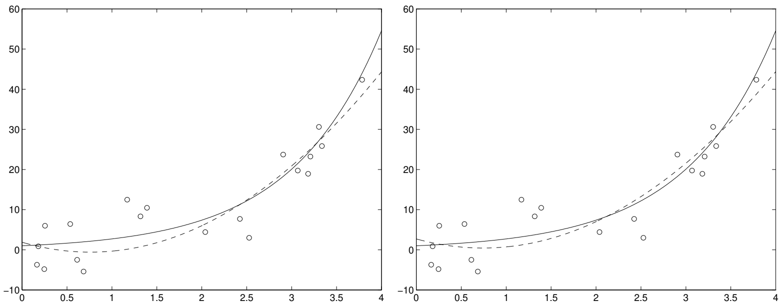
In order to better understand the circumstances in which tolerant training is useful, we view our theoretical results as a framework for generalization based upon the foundation of mathematical programming. Mathematical models of generalization [Vapnik, 1995, Wolpert, 1995] have grown from several diverse fields: for instance, statistics [Geman et al., 1992, Wahba, 1990], computational complexity [Valiant, 1984], and statistical physics [Tishby et al., 1989]. We believe that the methods and tools of mathematical optimization are an effective way to explore the theory of generalization, which in turn will lead to more effective applications of supervised learning.

To illustrate the idea of tolerant training, Figure 1 visually displays its effect on a simple linear approximation problem. The solid line in each figure represents the underlying function, the exponential e^x, on the interval $[0, 4]$. The training points, shown with open circles, have zero-mean Gaussian noise added to e^x for specific values of x. A quadratic curve (dotted line) is fit to the training set with a tolerant least-squares fit, as we gradually increase the fitting tolerance parameter τ. In this example, larger values of τ clearly allow a more faithful reconstruction of the underlying function, decreasing the dependence on the particular training points. Through theoretical and experimental results, we will show that this improvement occurs in a wide variety of learning situations.



(a) $\tau = 0$

(b) $\tau = 1$



(c) $\tau = 2$

(d) $\tau = 3$

Figure 1: A simple curve-fitting problem with improved solutions obtained via tolerant training. The solid line is the underlying function e^x. Twenty points sampled with Gaussian noise (N(0,4)) are shown as open dots. A quadratic curve ax^2+bx+c, shown with a dotted line, is fit to the points using least-squares fitting. As the tolerance parameter τ is increased, the least-squares fit improves significantly.

We outline the paper now. In Section 2 we obtain conditions for improved generalization for linear training and testing sets. The principal result of this section, Theorem 2, shows that there exists a training set tolerance $\bar{\tau}$ such that improved generalization, over zero-tolerance training, is obtained on any testing set sufficiently close to the training set in the sense of condition (23). Extension to a more general testing set is given in Theorem 3 with an improved generalization condition (32). These results are placed in a context of some familiar concepts from the machine learning literature in Section 3. Section 4 gives some computational results for both linear systems covered by our theoretical results, as well as other, nonlinear problems. Section 5 concludes the paper with some remarks and some future research directions.

A word about our notation. Vector notation will be used throughout with all vectors being column vectors unless transposed by a superscript T to a row vector. The n-dimensional real space will be denoted by R^n. For $x \in R^n$, x_i will denote the i^{th} component. $A \in R^{m\times n}$ will denote an $m \times n$ matrix and A_i will denote the i^{th} row and A_I will denote a submatrix of A with rows $i \in I \subset \{1, \ldots, m\}$. The vector e will denote a vector of ones of arbitrary dimension. The identity matrix of arbitrary dimension will be denoted by I. For $x \in R^n$, $\|x\|$ will denote an arbitrary norm, while $\|x\|_1 = \sum_{i=1}^n |x_i|$ and $\|x\|_\infty = \text{maximum}_{1\leq i\leq n}\{|x_i|\}$ denote the 1-norm and ∞-norms respectively, and $\|x\|_2$ will denote the 2-norm, $(x^Tx)^{\frac{1}{2}}$. The notation $a := b$ means that a is defined by b. For a vector $x \in R^n$, $(x_+)_i := \text{maximum}\{x_i, 0\}, i = 1, \ldots, n$. A function $f : R^n \mapsto R^m$ is Lipschitz continuous on $S \subset R^n$ if $\|f(y) - f(x)\| \leq K\|y - x\|$ for all x, y in S for some $K > 0$. If B is a subset of A then $A - B$ denotes the complement of B in A. For a minimization problem $\text{minimize}_{x\in X}\, \theta(x)$ where $\theta : R^n \mapsto R, X \subset R^n$, the solution set is defined by $\arg\min_{x\in X} \theta(x)$, that is,

$$\arg\min_{x\in X} \theta(x) := \{y | y \in X, \theta(y) \leq \theta(x), \forall x \in X\}.$$

2 Improved generalization for linear systems

In this section we consider a simple linear model defined by the training set $\{A, a\}$ and attempt to learn the linear relation

$$Ax = a. \tag{1}$$

Here A is a given $m \times n$ real matrix, a is a given $m \times 1$ real vector (hence, each pair (A_i, a_i) represents one training example) and x is an unknown vector $x \in R^n$ to be learned. The system (1) in general does not have a solution, and one resorts to minimizing some residual error function associated with it. Typically one solves the minimization problem

$$\underset{x \in R^n}{\text{minimize}} \qquad \|Ax - a\|, \tag{2}$$

where $\|\cdot\|$ is some norm on R^m. We shall use the 2-norm in our analysis for simplicity. In the present approach, solving the minimization problem (2) is referred to as *zero-tolerance training*, because any deviation from $0 \in R^m$ by $(Ax - a)$ is considered to be an error.

If the data set $\{A, a\}$ contains some errors, and possibly outliers, it seems unreasonable to attempt to enforce the relation (1) by means of the minimization problem (2). Instead we tolerate an error of magnitude τ, for some $\tau > 0$, in satisfying each of the equalities of (1). That is, we relax our original system of equalities (1) to the set of inequalities

$$-e\tau \leq Ax - a \leq e\tau, \tag{3}$$

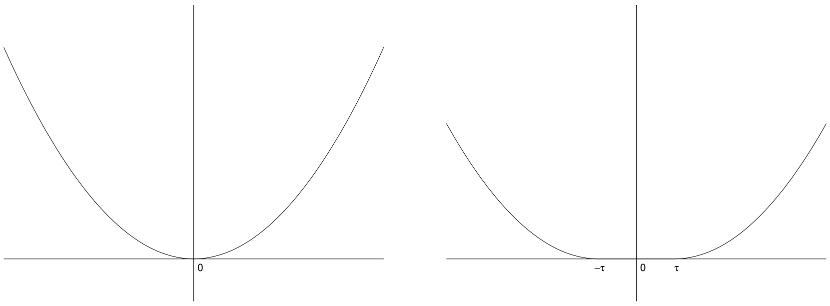
where e is a vector of ones in R^m. A plausible justification for (3) is that each component of the vector a may be known only within an accuracy of τ. We therefore attempt to enforce condition (3) by solving the following regularized quadratic program, for some nonnegative τ and a small positive ϵ:

$$\begin{array}{ll} \underset{x,y,z}{\text{minimize}} & \frac{1}{2}\|y\|_2^2 + \frac{1}{2}\|z\|_2^2 + \frac{\epsilon}{2}\|x\|_2^2 \\ \text{subject to} & \begin{array}{c} -z - e\tau \leq Ax - a \leq e\tau + y \\ y, z \geq 0. \end{array} \end{array} \tag{4}$$

Here, $y, z \in R^m$ are vectors of errors that will be made as small as possible by minimizing their Euclidean distance from the origin in R^m, and ϵ is a small fixed positive regularization constant that ensures the uniqueness of the x components of the solution. For $\tau = 0$, the quadratic program (4) reduces to the regularized least squares problem

$$\underset{x}{\text{minimize}} \qquad \frac{1}{2}\|Ax - a\|_2^2 + \frac{\epsilon}{2}\|x\|_2^2, \tag{5}$$

that we call the zero-tolerance training problem here. Figure 2 depicts the error for one component $A_i x - a_i$ as measured by the minimum of the quadratic program (4) with zero tolerance ($\tau = 0$) and a positive tolerance ($\tau > 0$).



(a) $\tau = 0$ (b) $\tau > 0$

Figure 2: Error measured by (4) with zero tolerance and positive tolerance.

If we let $(x(\tau), y(\tau), z(\tau))$ denote a solution of (4) for a fixed τ, that is

$$\begin{aligned}(x(\tau), y(\tau), z(\tau)) := \arg\min_{x,y,z}\{ & \frac{1}{2}\|y\|_2^2 + \frac{1}{2}\|z\|_2^2 + \frac{\epsilon}{2}\|x\|_2^2 \,\Big|\\ & -z - e\tau \le Ax - a \le e\tau + y, y \ge 0, z \ge 0\},\end{aligned} \tag{6}$$

then the principal question we wish to answer is this: How well does $x(\tau)$ generalize? To answer this question, we consider the following simple perturbation of our linear system (1) as our *testing set*:

$$Ax = a + p, \tag{7}$$

where p is a completely arbitrary perturbation vector in R^m. In particular for a given *fixed* p we consider the error in satisfying (7) by the $x(\tau)$ learned by solving the quadratic program (4), that is,

$$f(\tau) := \frac{1}{2}\|Ax(\tau) - a - p\|_2^2. \tag{8}$$

We would like to know when $f(0)$ is *not* a local minimum of $f(\tau)$ on $\{\tau | \tau \geq 0\}$. This would then tell us that, for sufficiently small $\tau > 0$, $f(\tau)$ will decrease, thus improving generalization. Theorem 2 answers this question by giving a precise condition on p that shows when tolerant training, achieved by solving (4) for some $\tau > 0$, improves the generalization error (8) on a testing set of the form of (7). To establish this result, we will show that $f(\tau)$ decreases as τ increases near 0, under an appropriate condition on p. To show this, we first characterize $x(\tau)$ for $\tau \in [0, \tilde{\tau}]$, for some $\tilde{\tau} > 0$, as a linear function in τ.

Lemma 1 *(Linearity of $x(\tau)$ on $[0, \tilde{\tau}]$)*

There exists a $\tilde{\tau} > 0$ such that for $\tau \in [0, \tilde{\tau}]$, $(x(\tau), y(\tau), z(\tau))$ solves (4) and

$$x(\tau) = x(0) + x'(0)\tau \tag{9}$$

where $x(0)$ solves (5) and $x'(0)$ is some vector in R^n.

Proof The quadratic program (4) is equivalent to the problem

$$\underset{x}{\text{minimize}} \qquad \frac{1}{2}\left\|(Ax - a - e\tau)_+\right\|_2^2 + \frac{1}{2}\left\|(-Ax + a - e\tau)_+\right\|_2^2 + \frac{\epsilon}{2}\left\|x\right\|_2^2. \tag{10}$$

The objective function of (10) is a differentiable, strongly convex function. Hence a necessary and sufficient condition for (10) to have a unique minimum solution $x(\tau)$ is that the gradient of the objective function vanish, that is,

$$A^T(Ax(\tau) - a - e\tau)_+ - A^T(-Ax(\tau) + a - e\tau)_+ + \epsilon x(\tau) = 0. \tag{11}$$

Let $\{I(\tau), II(\tau), III(\tau)\}$ be a partition of the row index set $\{1, \ldots, m\}$ of A defined as follows:

$$\begin{aligned}
I(\tau) :=& \{i : A_i x(\tau) - a_i - \tau \geq 0, A_i x(\tau) - a_i + \tau \geq 0\} \\
=& \{i : A_i x(\tau) - a_i - \tau \geq 0\} \\
II(\tau) :=& \{i : -A_i x(\tau) + a_i - \tau \geq 0, -A_i x(\tau) + a_i + \tau \geq 0\} \\
=& \{i : -A_i x(\tau) + a_i - \tau \geq 0\} \\
III(\tau) :=& \{i : -A_i x(\tau) + a_i + \tau > 0, A_i x(\tau) - a_i + \tau > 0\} \\
=& \{1, \ldots, m\} - \{I(\tau) \cup II(\tau)\}.
\end{aligned} \tag{12}$$

Equation (11) can then be written as

$$\epsilon x(\tau) + \begin{bmatrix} A_{I(\tau)} \\ A_{II(\tau)} \\ A_{III(\tau)} \end{bmatrix}^T \begin{bmatrix} A_{I(\tau)}x(\tau) - a_{I(\tau)} - e_{I(\tau)}\tau \\ A_{II(\tau)}x(\tau) - a_{II(\tau)} + e_{II(\tau)}\tau \\ 0_{III(\tau)} \end{bmatrix} = 0. \tag{13}$$

Let τ now approach zero. Because of the finite number of possible partitions $\{I(\tau), II(\tau), III(\tau)\}$, one fixed partition will repeat infinitely often on a sequence $\{\tau_j\} \downarrow 0$. Denote that partition by $\{I, II, III\}$. Hence (13) gives

$$(\epsilon I + A_I^T A_I + A_{II}^T A_{II})x(\tau) = (A_I^T a_I + A_{II}^T a_{II}) + (A_I^T e_I - A_{II}^T e_{II})\tau, \qquad \tau \in \{\tau_j\} \downarrow 0. \tag{14}$$

Thus

$$x(\tau) = (\epsilon I + A_I^T A_I + A_{II}^T A_{II})^{-1}((A_I^T a_I + A_{II}^T a_{II}) + (A_I^T e_I - A_{II}^T e_{II})\tau), \quad \tau \in \{\tau_j\} \downarrow 0. \tag{15}$$

Since $x(\tau)$ is continuous, in fact Lipschitz continuous on $\{\tau | \tau \geq 0\}$ [Robinson, 1981], it follows that (15) holds in the limit for $\tau = 0$ and hence (9) holds for $\tau \in \{\tau_j\} \downarrow 0$ as well as for $\tau = 0$ with the vectors $x(0)$ and $x'(0)$ of (9) given by:

$$\begin{aligned} x(0) &= (\epsilon I + A_I^T A_I + A_{II}^T A_{II})^{-1}(A_I^T a_I + A_{II}^T a_{II}), \\ x'(0) &= (\epsilon I + A_I^T A_I + A_{II}^T A_{II})^{-1}(A_I^T e_I - A_{II}^T e_{II}). \end{aligned} \tag{16}$$

We can verify that the partition $\{I, II, III\}$ satisfies (12) for $\tau = 0$ once $x(0) + x'(0)\tau$ is substituted for $x(\tau)$ in (12) as follows:

$$\begin{aligned} I(\tau) =& \{i | A_i x(0) - a_i + \tau(A_i x'(0) - 1) \geq 0\} \\ II(\tau) =& \{i | -A_i x(0) + a_i + \tau(-A_i x'(0) - 1) \geq 0\} \\ III(\tau) =& \{1, \ldots, m\} - \{I(\tau) \cup II(\tau)\}. \end{aligned} \tag{17}$$

Since these relations hold for $\{\tau_j\} \downarrow 0$, it follows that:

$$A_i x(0) - a_i \geq 0 \text{ for } i \in I$$

$$-A_i x(0) + a_i \geq 0 \text{ for } i \in II,$$

and hence (2) holds for $\tau = 0$. Define $\tilde{\tau} := \tau_0$, the initial term of the sequence $\{\tau_j\} \downarrow 0$. Now since (12) and (13), which are linear in $x(\tau)$ and are satisfied by $x(\tau) = x(0) + x'(0)\tau$ at $\tau = 0$ and $\tau = \tilde{\tau}$, for the fixed partition $\{I, II, III\}$, it follows that they are also satisfied for the fixed partition $\{I, II, III\}$ at $\tau = \lambda\tilde{\tau}, 0 \leq \lambda \leq 1$, by

$$s(\tau) := x(0) + \lambda x'(0)\tilde{\tau}, \qquad 0 \leq \lambda \leq 1, \tag{18}$$

which is equivalent to (9) holding on $\tau \in [0, \tilde{\tau}]$, because $s(\tau)$ satisfies (13) or equivalently (11) which implies that it solves (10).

□

We immediately note that since $x(0)$ of (9) solves the regularized least squares problem (5) it follows that

$$x(0) \in \arg\min_x \frac{1}{2} \|Ax - a\|_2^2 + \frac{\epsilon}{2} \|x\|_2^2, \tag{19}$$

and consequently

$$\epsilon x(0) + A^T(Ax(0) - a) = 0. \tag{20}$$

Hence

$$x(0) = (\epsilon I + A^T A)^{-1} A^T a. \tag{21}$$

We note that the argument used in Lemma 1 can also be applied to any $\hat{\tau} > 0$ to establish the linearity of the solution $x(\tau)$ on $[\hat{\tau}, \hat{\tau} + \delta]$ for some $\delta > 0$. Hence $x(\tau)$ is piecewise linear and continuous on $\{\tau | \tau \geq 0\}$. It follows then that the training error function $f(\tau)$ of (8) is continuous and piecewise-quadratic for such $x(\tau)$ for $\tau \geq 0$. We also note that the minimum value of (4) or equivalently of (10) approaches zero for $\tau \geq \hat{\tau}$ for some $\hat{\tau} \geq 0$ as ϵ approaches zero ([Mangasarian, 1986], Theorem 2.5). In fact,

$$\hat{\tau} := \underset{x}{\text{minimum}} \|Ax - a\|_\infty . \tag{22}$$

Hence we shall restrict our attention to the interval $\tau \in [0, \hat{\tau}]$. It follows that the continuous, piecewise quadratic testing error function $f(\tau)$ attains its minimum on this interval at a point other than $\tau = 0$ whenever condition (23) below is satisfied. We state our principal result now.

Theorem 2 *(Improved generalization with positive tolerance for testing model $Ax = a + p$)*

The testing set error function $f(\tau)$ of (8) has a strict local maximum at $\tau = 0$ and a global minimum on $[0, \hat{\tau}]$, where $\hat{\tau}$ is defined by (22), at some $\bar{\tau} > 0$, whenever

$$(\epsilon x(0) + A^T p)^T (x(\tau) - x(0)) > 0, \tag{23}$$

for some $\tau \in (0, \tilde{\tau}]$, where $\tilde{\tau}$ is defined in Lemma 1.

Proof For $\tau \in [0, \tilde{\tau}]$ where $\tilde{\tau}$ is defined in Lemma 1, we have:

$$\begin{aligned} f(\tau) - f(0) &= \frac{1}{2}\|A(x(0) + x'(0)\tau) - a - p\|_2^2 - \frac{1}{2}\|Ax(0) - a - p\|_2^2 \\ &\qquad\qquad \text{(By Lemma 1)} \\ &= \frac{1}{2}\|Ax'(0)\tau\|_2^2 + \tau x'(0)^T A^T (Ax(0) - a - p) \\ &\qquad\qquad \text{(By (20))} \\ &= \tau \left[\frac{\tau}{2}\|Ax'(0)\|_2^2 - x'(0)^T (A^T p + \epsilon x(0))\right] \\ &< 0 \qquad \text{for all sufficiently small } \tau > 0 \text{ and } x'(0)^T (A^T p + \epsilon x(0)) > 0. \end{aligned} \tag{24}$$

But by (9) and (23)

$$x'(0)^T (A^T p + \epsilon x(0)) = \frac{(x(\tau) - x(0))^T}{\tau} (A^T p + \epsilon x(0)) > 0 \tag{25}$$

Hence by (23) – (25) $f(\tau) < f(0)$ for all sufficiently small $\tau \geq 0$ and $f(0)$ is a strict local maximum of $f(\tau)$ on $\{\tau | \tau > 0\}$. Since $f(\tau)$ is continuous on $[0, \hat{\tau}]$, it follows that it attains its global minimum on $[0, \hat{\tau}]$ at some $\bar{\tau} > 0$.

□

Ignoring the regularization term $\epsilon x(0)$, since ϵ can be made arbitrarily small, condition (23) can be replaced by

$$p^T A(x(\tau) - x(0)) > 0. \tag{26}$$

We interpret this condition as follows. When the difference of the maps $A(x(\tau))$ and $A(x(0))$ is not drastically different from the perturbation direction p (i.e., it makes an acute angle with it), the training with a nonzero tolerance τ will always improve the testing results.

We conclude this section by extending Theorem 2 to a more general testing model

$$Cx = c \tag{27}$$

where $C \in R^{k \times n}$ and $c \in R^k$ are chosen arbitrarily. To do this we define a corresponding error function $g(\tau)$ to (8) which measures the error in satisfying (27) by the $x(\tau)$ learned by solving the regularized quadratic program (4). We thus have the error

$$g(\tau) := \frac{1}{2} \|Cx(\tau) - c\|_2^2. \tag{28}$$

For τ in $[0, \tilde{\tau}]$, where $\tilde{\tau}$ is defined in Lemma 1 we have:

$$\begin{aligned} g(\tau) - g(0) &= \frac{1}{2} \|C(x(0) + x'(0)\tau) - c\|_2^2 - \frac{1}{2} \|Cx(0) - c\|_2^2 \\ &= \tau \left[\frac{\tau}{2} \|Cx'(0)\|_2^2 + x'(0)^T C^T (Cx(0) - c) \right] \\ &< 0 \qquad \text{for all sufficiently small } \tau > 0 \text{ and } x'(0)^T C^T (Cx(0) - c) < 0. \end{aligned}$$

By (9) we have that for $\tau \in [0, \tilde{\tau}]$:

$$\begin{aligned} x'(0)^T C^T (Cx(0) - c) &= \frac{(x(\tau) - x(0))^T}{\tau} C^T (Cx(0) - c) \\ &= \frac{1}{\tau} (r(\tau) - r(0))^T r(0), \end{aligned} \tag{29}$$

where $r(\tau)$ is the residual vector for the testing model (27), that is,

$$r(\tau) := Cx(\tau) - c. \tag{30}$$

Hence for $\tau \in [0, \tilde{\tau}]$,

$$g(\tau) < g(0) \qquad \text{whenever} \;\; \|r(0)\|_2^2 > r(0)^T r(\tau). \tag{31}$$

This leads to the following improved generalization theorem for the more general testing model (27). We skip details of the proof that are very similar to the proof of Theorem 2.

Theorem 3 *(Improved generalization with positive tolerance for testing model $Cx = c$)* *Let $x(\tau)$ be defined by the tolerant training of $Ax = a$ by the regularized quadratic program (6) with tolerance $\tau \geq 0$. Let $g(\tau)$ denote the error generated by $x(\tau)$ in the testing model $Cx = c$, defined by (28). The zero-tolerance error $g(0)$ is a local maximum of $g(\tau)$ over the set $\{\tau : \tau \geq 0\}$ whenever*

$$\|r(0)\|_2^2 > r(\tau)^T r(0) \text{ for some } \tau \in (0, \tilde{\tau}], \tag{32}$$

where $r(\tau)$ is defined by (30) and $\tilde{\tau}$ in Lemma 1. Furthermore, the testing set error function $g(\tau)$ of (28) has a global minimum on $[0, \hat{\tau}]$, where $\hat{\tau}$ is defined by (22), at some $\bar{\tau} > 0$, whenever condition (32) holds and $\hat{\tau} > 0$.

This concludes our theoretical treatment of tolerant training. We now give some machine learning interpretations of our results as well as their computational validation.

3 Machine learning context

In this section we examine the mathematical results from the previous section in light of two well-known results from the machine learning literature: the conservation of generalization laws, and the bias/variance tradeoff.

The conservation of generalization theorem [Schaffer, 1993, Schaffer, 1994, Wolpert, 1992a, Wolpert, 1992b] (a.k.a., "no free lunch theorem") states that all learning algorithms are created equal. In the context of classification, this means that the generalization performance of any learning system, when averaged across all possible learning situations, is 0.5. That is, the classifier will be right exactly half the time when tested on examples not in the training set. Therefore, improved generalization in a particular learning situation will be exactly offset by worse performance in one or more different situations. The result extends trivially to more general function approximation problems such as those examined here. This seems to suggest that the search for good *general purpose* learning systems is futile, since no algorithm can perform at a better-than-average level under all conditions. Similarly, extensions or variations of existing learning systems cannot be expected to improve upon the underlying method in the general case.

Our main result from the previous section states that generalization performance on the perturbed (underlying) system $Ax = a + p$ is improved when $\epsilon x(0) + A^T p$ makes an acute angle with the vector $x(\tau) - x(0)$. Since tolerant training is simply a variation of least-error fitting, the conservation of generalization law would lead us to believe that this condition holds in half of all possible learning situations, which is in fact the case. Fix $x(\tau)$ and $x(0)$ and consider all possible values of the vector p. The plane $(\epsilon x(0) + A^T p)^T (x(\tau) - x(0)) = 0$ divides R^m into two halfspaces, and therefore essentially half of the possible p vectors will satisfy $(\epsilon x(0) + A^T p)^T (x(\tau) - x(0)) > 0$.

Tolerant training, then, is simply an overtraining avoidance technique, and should therefore be viewed in the same light as other overfitting avoidance techniques. In the field of artificial neural networks, such techniques include early stopping [Lang et al., 1990], soft weight sharing [Nowlan and Hinton, 1991], and various methods for deleting weights and nodes, such as optimal brain damage [Le Cun et al., 1990]. Other methods for overtraining avoidance in pattern recognition, learning and regression include node pruning in decision trees [Breiman et al., 1984, Quinlan, 1993a], k-nearest-neighbors [Hart, 1967], smoothing splines [Wahba, 1990], and feature selection [Draper and Smith, 1966]. The true test of tolerant training, and other overtraining avoidance techniques, is whether it improves generalization on a wide variety of common learning problems, particularly those in which overtraining is likely to occur. Our computational results, including those in the next section, indicate that it does.

We now examine the error-reduction capabilities of tolerant training in the context of the bias/variance tradeoff [Geman et al., 1992]. Generalization error can be decomposed into two components: *variance*, which refers to the variability of the learned concept depending on the particular training data used, and *bias*, which measures the limitations of the representational power of the learning system. Typically, reducing one of these types of error drives up the other type. For instance, one can reduce the bias of a neural network by adding more hidden units, thereby increasing the number of functions the network can accurately represent. However, the resulting net is a more highly-parameterized modes, and will be therefore be more sensitive to the peculiarities of the training set and thus have higher variance.

Tolerant training is a variance reduction method. Increasing the tolerance parameter τ frees the learner from trying to exactly fit each of the training examples, reducing the variance in much the same way as increasing the value of the parameter k in k-nearest-neighbors regression. As τ increases from zero, then, the bias error increases while the variance error decreases; if the variance error is changing faster, then a local minimum of the error is available. Eventually, the error band controlled by τ is so wide that the variance error goes to zero, that is, the training set is ignored.

4 Computational results

We turn now to some computational experiments that corroborate the theoretical results presented above and also indicate the wider applicability of the tolerant training approach.

4.1 Demonstration of Theorem 2

First we present a computational corroboration of Theorem 2. We use the first training model described in Section 2, learning the vector x with a least-squared fit so that $Ax \approx a$ and testing on the perturbed system $Ax = a + p$. In the learning situation depicted by the bottom curve of Figure 3, $(\epsilon x(0) + A^T p)$ makes an acute angle with $(x(\tau) - x(0))$ in the neighborhood of $\tau = 0$. Hence, as τ increases, the test error on $Ax = a + p$ decreases. In the upper curve, we show a perturbation for which the angle is reversed: the testing model is $Ax = a - p$, using the same perturbation p. Here the testing error goes up as τ increases away from zero.

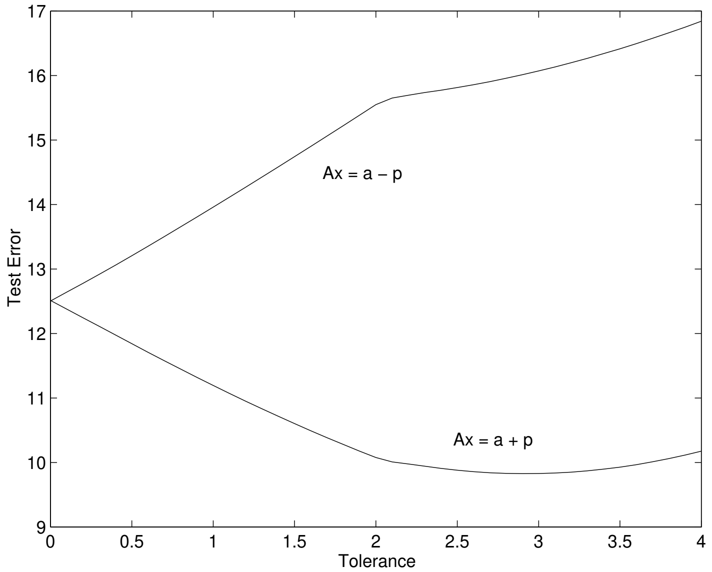


Figure 3: A computational example of Theorem 2. The bottom curve shows a situation in which $(\epsilon x(0) + A^T p)^T (x(\tau) - x(0)) > 0$, therefore the testing error decreases by use of tolerant training in the neighborhood of $\tau = 0$. Conversely, the top curve demonstrates $(\epsilon x(0) + A^T p)^T A(x(\tau) - x(0)) < 0$, and tolerant training is of no value.

4.2 Real data sets

Experiments were performed on a number of data sets collected from Statlib [Meyer, 1989] and the UCI machine learning repository [Murphy and Aha, 1994]. Each of these problems has a number of linear input features (both real-valued and integer-valued) and a single linear output. Simple least-error regression was performed to fit a linear model to each data set, using both a 1-norm fit (least absolute error) and a 2-norm fit (least squared error). The regressions were then repeated using tolerant training. The τ parameter was selected dynamically for each problem based on the mean absolute error of the zero-tolerance training case. This error was divided by 2, 5, 10 and 20, and each resulting value was used as τ to generate a tolerant model. These models were evaluated using a tuning or validation set [Lang et al., 1990], and the value that performed best was then used for a final model that

Data Set	Tolerance	Mean Absolute Error	Mean Squared Error
Cancer Recurrence (m = 47, n = 32)	$\tau = 0$	83.42 ± 40.53	660.1 ± 431.6
[Mangasarian et al., 1995]	$\tau > 0$	36.80 ± 15.40	585.5 ± 250.9
Servo (m = 167, n = 4)	$\tau = 0$	1.001 ± 0.351	2.944 ± 1.873
[Quinlan, 1993b]	$\tau > 0$	1.012 ± 0.344	2.934 ± 1.872
Pollution (m = 60, n = 15)	$\tau = 0$	62.75 ± 32.55	5447 ± 8250
[McDonald and Schwing, 1973]	$\tau > 0$	45.93 ± 19.15	3661 ± 3118
Strikes (m = 35, n = 4)	$\tau = 0$	191.9 ± 117.7	55035 ± 63589
[Western, 1996]	$\tau > 0$	192.2 ± 130.9	51487 ± 63264
Ships (m = 34, n = 3)	$\tau = 0$	9.794 ± 8.509	152.3 ± 256.3
[McCullagh and Nelder, 1989]	$\tau > 0$	8.306 ± 5.728	128.3 ± 220.4
Housing (m = 50, n = 13)	$\tau = 0$	2.781 ± 1.859	43.10 ± 76.36
[Harrison and Rubinfeld, 1978]	$\tau > 0$	2.875 ± 1.885	14.87 ± 27.05

Table 1: Cross-validated estimates of error and standard deviation of error performing regression with and without tolerant training. In accordance with the theoretical results, m is the number of training examples and n is the dimensionality of the problem.

was evaluated on a testing set. Error rates and standard deviations of the errors were estimated using tenfold cross-validation [Stone, 1974] and are shown in Table 4.2.

In *nine of the twelve* cases we examined, tolerant training improved the estimated error rate, often dramatically. The standard deviation of the observed error was also lowered in most cases. The tolerant training method appears to lower the error most in the cases that have a small number of examples relative to the problem dimensionality, a point that we will examine further in a later section.

4.3 Artificial neural networks

This experiment explores the effect of tolerant training in a learning situation very different from the ones described previously. Here the tolerance is added to the hidden units of an artificial neural network classifier. Tests were performed on a relatively easily classifiable breast cancer diagnosis data set [Mangasarian et al., 1995, Wolberg et al., 1994, Murphy and Aha, 1994] that has thirty real-valued input features and two output classes.

The samples were then corrupted by changing the classification of various percentages of the samples. Neural networks were trained to convergence, using various levels of tolerance, on these corrupted data sets, and then tested on the original examples, which play the role of the underlying function. Note that this differs from the previous experiments in two ways: first, the neural network is a highly nonlinear model, and second, the problem is a classification task rather than function approximation (and hence, the output values and added noise are discrete).

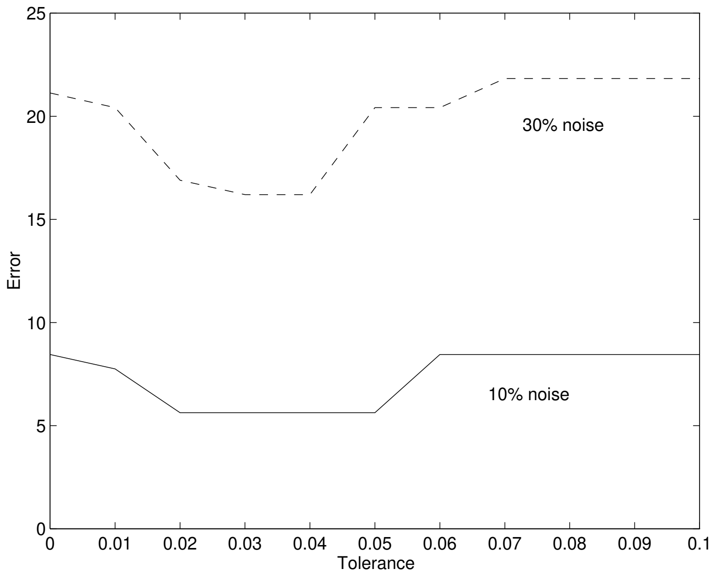


Figure 4: Tolerant training in artificial neural networks. In this experiment a neural network, with tolerance added to the hidden nodes, is trained on a real-world classification task. Classification noise is added to the training set, with the error measured on the original data. These results indicate that tolerant training can improve a learning system's ability to recover the underlying function even when the learning task lies far outside the limits of our theoretical analysis.

Results of one such experiment are shown in Figure 4. Again we see that, in every case, *some positive value of the tolerance parameter τ does a better job of recovering the underlying function than traditional, zero-tolerance training.*

Polynomial Degree	Tolerant Training		Regular Training	
	Error	Standard Deviation	Error	Standard Deviation
2	1.884	0.228	2.056	0.338
3	1.575	0.348	1.754	0.437

Table 2: Examples of improved generalization error using tolerant training, fitting e^x on $[0, 4]$ by a polynomial.

4.4 Best situations for tolerant training

In this set of experiments we fit the exponential curve e^x on the interval $[0, 4]$ with a polynomial, using training points with additive Gaussian noise. A best 1-norm fit is used rather than the 2-norm error used in the theoretical analysis. The results are then compared to the actual curve e^x. Some results are shown in Table 2. A reasonable heuristic was chosen to set the tolerance parameter τ to the mean absolute training error of the result obtained with zero-tolerance (i.e., standard least-error fit) training. These results are typical of the many different variations of this problem that were examined. The only cases in which tolerant training resulted in no improvement were those uniquely suited for least-error fitting, e.g., minimizing the squared error with added Gaussian noise. Even in these cases, tolerant training did not significantly raise the error.

To determine the types of problems to which tolerant training is best suited, the characteristics of the training data and the fitting polynomial were varied. In the results that follow, we observed the $x(\tau)$ which best fit the testing set, and computed its percentage improvement over $x(0)$, then averaged the "% Improvement" result over 100 random runs. We were therefore estimating the fitting improvement made *available* via tolerant training. In no instance do we attempt to find the "correct" value for τ. Therefore, the following are best-case results intended solely to compare different learning situations and determine those situations in which tolerant training offers the most potential improvement.

In the first experiment, the number of training points was varied from 10 to 100, with the noise variance held at four and a quadratic curve used as the concept representation. The higher number of points reduced the learning improvement made available by tolerant training from 25% to 5%.

Next we varied the amount of Gaussian noise added to the training points, with the variance of the noise varying from 0 to 10. As expected, more noise creates a higher-variance learning situation, and therefore tolerant training makes a greater improvement available, varying from 10% to 25%. The potentially surprising result in this experiment is that even in the no-noise case – i.e., the training points were sampled directly from the underlying function e^x – tolerant training offered a significant improvement in the quadratic's ability to fit the exponential.

We also tested the utility of tolerant training when fitting with different polynomials, up to degree 5. Hence the learning system is given significantly more expressive power than is necessary to fit the underlying function e^x. For these higher-degree polynomials, banded approximation offered a dramatic improvement in the fit, in the 30 to 35 percent range.

These experimental results show that, as a variance reduction procedure, tolerant training performs best in high-variance situations, which are pervasive in real-world learning tasks. In particular, tolerant training was shown to be most useful when:

- The number of training examples is small relative to the dimensionality of the problem,
- The examples contain features which are noisy and/or redundant, and
- The model is highly parameterized relative to the complexity of the underlying concept.

The burgeoning research in neural networks has resulted in many situations involving high-dimensional problems being approached with highly-parameterized models. Tolerant training can reduce the dependence of generalization performance upon finding precisely the right representation for the problem and the learning system.

5 Future work and conclusions

In summary, we have presented tolerant training, an easily implemented, robust method for overfitting avoidance in inductive learning. A theoretical justification for the technique was given for least-squares fitting on simple linear systems. This justification serves as the foundation for identifying exactly what types of learning situations are amenable to tolerant training. Our experimental results demonstrate that the utility of the technique extends well beyond the cases described in the theoretical results. They further show that, as a variance reduction procedure, tolerant training performs best in high-variance situations, which are pervasive in "real-world" learning tasks. In particular, tolerant training was shown to be useful when:

- The number of training examples is small relative to the dimensionality of the problem,
- The examples contain features which are noisy and/or redundant, and
- The model is highly parameterized relative to the complexity of the underlying concept.

The explosive growth in neural network research has resulted in many situations involving high-dimensional problems being approached with highly-parameterized models. Techniques like tolerant training can reduce the dependence of generalization performance upon finding precisely the right representation for the problem and the learning system. The next step in the exploration of the utility of tolerant training is an automatic method for choosing the tolerance parameter τ. Determining a useful τ for a particular learning problem turns tolerant training into a useful learning tool. In this paper, heuristic choices and a simple tuning set were used to choose a τ related to the observed zero-tolerance training error. This connection will be explored further, as well a more general method for determining τ based on intrinsic properties of the data set.

Acknowledgements

We wish to acknowledge useful discussions with our colleagues Robert R. Meyer, Stephen M. Robinson, Chunhui Chen, and Michael V. Solodov. This material is based on research supported by Air Force Office of Scientific Research Grant F–49620–94–1–0036 and National Science Foundation Grants CCR–9322479 and DCA-9024618.

References

[Breiman et al., 1984] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth, Inc., Pacific Grove, CA.

[Draper and Smith, 1966] Draper, N. R. and Smith, H. (1966). *Applied Regression Analysis.* John Wiley and Sons, New York.

[Geman et al., 1992] Geman, S., Bienestock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58.

[Harrison and Rubinfeld, 1978] Harrison, D. and Rubinfeld, D. L. (1978). Hedonic prices and the demand for clean air. *J. Environ. Economics and Management*, 5:81–102.

[Hart, 1967] Hart, P. (1967). The condensed nearest neighbor rule. *Transactions on Information Theory*, IT-14:515–516.

[Lang et al., 1990] Lang, K., Waibel, A., and Hinton, G. (1990). A time-delay neural network architecture for isolated word recognition. *Neural Networks*, 3:23–43.

[Le Cun et al., 1990] Le Cun, Y., Denker, J. S., and Solla, S. A. (1990). Optimal brain damage. In Touretzky, D. S., editor, *Advances in Neural Information Processing Systems*, volume 2, pages 598–605, San Mateo, CA. Morgan Kaufmann.

[Mangasarian, 1986] Mangasarian, O. (1986). Some applications of penalty functions in mathematical programming. In Conti, R., De Giorgi, E., and Giannessi, F., editors, *Optimization and Related Fields*, pages 307–329. Springer–Verlag, Heidelberg. Lecture Notes in Mathematics 1190.

[Mangasarian et al., 1995] Mangasarian, O. L., Street, W. N., and Wolberg, W. H. (1995). Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

[McCullagh and Nelder, 1989] McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*, chapter 6, page 137. Chapman and Hall, London, 2 edition.

[McDonald and Schwing, 1973] McDonald, G. C. and Schwing, R. C. (1973). Instabilities of regression estimates relating air pollution to mortality. *Technometrics*, 15:463–482.

[Meyer, 1989] Meyer, M. (1989). StatLib. Carnegie Mellon University Satistics Department, http://lib.stat.cmu.edu.

[Murphy and Aha, 1994] Murphy, P. M. and Aha, D. W. (1994). UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[Nowlan and Hinton, 1991] Nowlan, S. J. and Hinton, G. E. (1991). Simplifying neural networks by soft weight-sharing. In Moody, J., Hanson, S., and Lippmann, R., editors, *Advances in Neural Information Processing Systems*, volume 4, San Mateo, CA. Morgan Kaufmann.

[Quinlan, 1993a] Quinlan, J. R. (1993a). *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA.

[Quinlan, 1993b] Quinlan, J. R. (1993b). Combining instance-based and model-based learning. In Utgoff, P. E., editor, *Proceedings of the 10th International Conference on Machine Learning*, San Mateo. Morgan Kaufmann.

[Robinson, 1981] Robinson, S. M. (1981). Some continuity properties of polyhedral multifunctions. *Mathematical Programming Study*, 14:206–214.

[Schaffer, 1993] Schaffer, C. (1993). Overfitting avoidance as bias. *Machine Learning*, 10:153–178.

[Schaffer, 1994] Schaffer, C. (1994). A conservation law for generalization performance. In *Proceedings of the 11th International Conference on Machine Learning*, San Mateo, CA. Morgan Kaufmann.

[Stone, 1974] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society (Series B)*, 36:111–147.

[Tishby et al., 1989] Tishby, N., Solla, S., and Levin, E. (1989). Consistent inference of probabilities in layered networks: Predictions and generalization. In *IJCNN International Joint Conference on Neural Networks*, volume II, pages 403–409, New York. IEEE.

[Valiant, 1984] Valiant, L. (1984). A theory of the learnable. *Communications of the ACM*, 27:1134–1142.

[Vapnik, 1995] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* Springer-Verlag, New York.

[Wahba, 1990] Wahba, G. (1990). *Spline Models for Observational Data.* SIAM, Philadelphia.

[Western, 1996] Western, B. (1996). Vague theory and model uncertainty in macrosociology. *Sociological Methodology*, to appear.

[Wolberg et al., 1994] Wolberg, W. H., Street, W. N., and Mangasarian, O. L. (1994). Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates. *Cancer Letters*, 77:163–171.

[Wolpert, 1992a] Wolpert, D. H. (1992a). On overfitting as bias. Technical Report TR 92-03-5001, The Santa Fe Institute.

[Wolpert, 1992b] Wolpert, D. H. (1992b). On the connection between in-sample testing and generalization error. *Complex Systems*, 6:47–94.

[Wolpert, 1995] Wolpert, D. H., editor (1995). *The Mathematics of Generalization*, Reading, MA. Addison-Wesley.