Optimal Properties and Adaptive Tuning of Support Vector Machines (SVMs)

Grace Wahba
`http://www.stat.wisc.edu/˜wahba`
→ *TRLIST*

Based on work of Yi Lin
`http://www.stat.wisc.edu/˜yilin`
and joint work with
Yi Lin, Hao Zhang and Yoonkyun Lee

MSRI Workkshop on Nonlinear Estimation and Classification
March 20, 2001

Abstract

We review and compare the definition of SVM's in their penalty method formulation and analogous penalized likelihood estimates, when the training set consists of yes-no responses for membership in class $\mathcal{A}$, along with attribute vectors. SVM's return a yes-no response for a new attribute vector x, while penalized likelihood estimates return an estimate of the probability $p(x)$ of membership in class $\mathcal{A}$ for a new attribute vector x. We describe a version of the generalized approximate cross validation (GACV) for tuning or controlling the bias-variance tradeoff a. k. a. goodness of fit/complexity tradeoff for the SVM case. A result of Yi Lin (UW-Madison Statistics Dept TR 1014, 1999) that the (tuned) SVM in the balanced case is asymptotically estimating sign$[p(x) - 1/2]$ is noted. In that case it can be shown that the GACV is asymptotically tuning SVM's against the misclassification rate. The results are generalized to the unbalanced case where the fraction of members of class $\mathcal{A}$ in the training set is different than that in the general population, and the costs of misclassification for the two kinds of errors are different.

References: Available via the home pages of Yi Lin and Grace Wahba.

Y. Lin. Support vector machines and the Bayes rule in classification. Technical Report 1014, Department of Statistics, University of Wisconsin, Madison WI, 1999.

Y. Lin. On the support vector machine. Technical Report 1029, Department of Statistics, University of Wisconsin, Madison WI, 2000.

Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in nonstandard situations. Technical Report 1016, Department of Statistics, University of Wisconsin, Madison WI, 2000. To appear, *Machine Learning*.

Y. Lin, G. Wahba, H. Zhang, and Y. Lee. Statistical properties and adaptive tuning of support vector machines. Technical Report 1022, Department of Statistics, University of Wisconsin, Madison WI, 2000. To appear, *Machine Learning*.

G. Wahba. Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods-Support Vector Learning*, pages 69–88. MIT Press, 1999.

G. Wahba, Y. Lin, and H. Zhang. Generalized approximate cross validation for support vector machines. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 297–311. MIT Press, 2000.

X. Lin, G. Wahba, D. Xiang, F. Gao, R. Klein, and B. Klein. Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV. Technical Report 998, Department of Statistics, University of Wisconsin, Madison WI, 2000. Slightly revised version to appear, *Ann. Statist.*, **28**.

G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.

OUTLINE

1. Review of Optimal Classification.

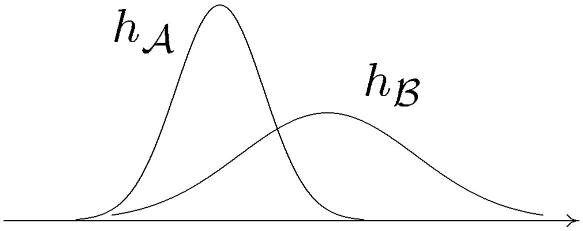
2. Comparison of penalized likelihood and SVM classifiers.

3. The standard case SVM – equal cost of misclassification and representative training set. GACV tuning for the standard case.

4. Yi Lin’s theorem: The (tuned) SVM is estimating the sign of the log-odds ratio and minimizing the expected misclassification rate.

5. Extension to the non-standard case:
Non-representative training set, unequal costs.

♣♣ Optimal Classification and the Neyman-Pearson Lemma:



$h_{\mathcal{A}}(\cdot), h_{\mathcal{B}}(\cdot)$ densities of x

for class $\mathcal{A}$ and class $\mathcal{B}$.

NOTATION:

$\pi_{\mathcal{A}} =$ prob. next observation (Y) is an $\mathcal{A}$

$\pi_{\mathcal{B}} = 1 - \pi_{\mathcal{A}} =$ prob. next observation is a $\mathcal{B}$

$$\begin{aligned} p(x) &= prob\{Y = \mathcal{A}|x\} \\ &= \frac{\pi_{\mathcal{A}} h_{\mathcal{A}}(x)}{\pi_{\mathcal{A}} h_{\mathcal{A}}(x) + \pi_{\mathcal{B}} h_{\mathcal{B}}(x)} \end{aligned}$$

Let $c_{\mathcal{A}} =$ cost to falsely call a $\mathcal{B}$ an $\mathcal{A}$

$c_{\mathcal{B}} =$ cost to falsely call an $\mathcal{A}$ a $\mathcal{B}$

Bayes classification rule: Let

$\phi(x)$: $\quad x \to \{{}^{\mathcal{A}}_{\mathcal{B}}\}$

Expected cost:

$E\,\{c_{\mathcal{A}}[1-p(x)]\ I(\phi(x)=\mathcal{A})\}$

get a $\mathcal{B}$ and call it an $\mathcal{A}$

$+E\,\{c_{\mathcal{B}}[p(x)]\ \ I(\phi(x)=\mathcal{B})\}$

get an $\mathcal{A}$ and call it $\mathcal{B}$

Optimum (Bayes) classifier:

$$\phi_{\text{OPT}}(x) = \begin{cases} \mathcal{A} & \text{if } \frac{p(x)}{1-p(x)} > \frac{c_{\mathcal{A}}}{c_{\mathcal{B}}}, \\ \mathcal{B} & \text{otherwise.} \end{cases}$$

To estimate $p(x)$, alternatively let $f(x) = \log p(x)/(1-p(x))$, the log odds ratio a.k.a. the logit. “Standard” case: Training set

$$\{y_i, x_i\} \qquad \begin{array}{l} y_i \in \{\mathcal{A}, \mathcal{B}\} \\ x_i \in \mathcal{T}, \text{ some index set} \end{array}.$$

Relative frequency of $\mathcal{A}$’s in the training set is about the same as in the general population.

Penalized log likelihood estimation:

Estimate f by penalized likelihood. If $c_{\mathcal{A}}/c_{\mathcal{B}} = 1$, then the optimal classifier is

$$\begin{array}{l} f(x) > 0 \text{ (equivalently, } p(x) - \frac{1}{2} > 0) \rightarrow \mathcal{A} \\ f(x) < 0 \text{ (equivalently, } p(x) - \frac{1}{2} < 0) \rightarrow \mathcal{B} \end{array}$$

♣♣ Penalized log likelihood estimation of the logit $f = \log[p/(1-p)]$.

$$y = \begin{matrix} 1 & = \mathcal{A} \\ 0 & = \mathcal{B} \end{matrix} \text{ (important)}$$

The probability distribution function (likelihood) for $y \mid p$ is: $\mathcal{L} = p^y(1-p)^{1-y} = \begin{cases} p & \text{if } y = 1 \\ (1-p) & \text{if } y = 0 \end{cases}$
and the negative log likelihood is

$$-\log \mathcal{L} = -\log[p^y(1-p)^{1-y}]$$
$$= -y \log p - (1-y)\log(1-p).$$

Using $p = e^f/(1+e^f)$ gives
$-\log \mathcal{L} = -yf + \log(1+e^f)$

♣♣ Penalized log likelihood estimation of f (continued) (special case).

$$\{y_i, x_i\},\ \ y_i = \begin{array}{c} 1 \\ 0 \end{array}\ \ , x_i \in \mathcal{T}$$

Find $f(x) = d + h(x)$ with $h \in \mathcal{H}_K$ to min

$$\frac{1}{n} \sum_{i=1}^{n} \left[-y_i f(x_i) + \log(1 + e^{f(x_i)}) \right] + \lambda \|h\|^2_{\mathcal{H}_K}$$

where $\mathcal{H}_K$ is the reproducing kernel Hilbert space (RKHS) with reproducing kernel

$$K(s,t),\ \ s,t, \in \mathcal{T}.$$

Theorem: [KW71]

$$f_\lambda(x) = d + \sum_{i=1}^{n} c_i K(x, x_i).$$

♣♣ Penalized log likelihood estimation of f (continued)

$$f_\lambda(x) = d + \sum_{i=1}^{n} c_i K(x, x_i)$$

Find $d, c = (c_1, \ldots, c_n) = c_\lambda$ to minimize

$$\frac{1}{n}\sum_{i=1}^{n}\left[-y_i f(x_i) + \log(1 + e^{f(x_i)})\right] + \lambda\|h\|^2_{\mathcal{H}_K}.$$

Here

$$\|h\|^2_{\mathcal{H}_K} \equiv \sum_{i,j=1}^{n} c_i c_j \; K(x_i, x_j).$$

Given λ, this is a nice strictly convex optimization problem. Choose λ by GACV [LWXGKK+00]. Target for GACV is to minimize the Comparative Kullback-Liebler (CKL) distance of the estimate from the true distribution:

$$R(\lambda) = E_{f_{true}} \sum_{i=1}^{n} -y_{new.i} f_\lambda(x_i) + \log(1 + e^{f_\lambda(x_i)}).$$

♣♣ Support Vector Machines

$$y = \begin{matrix} +1 = \mathcal{A} \\ -1 = \mathcal{B} \end{matrix} \text{ (note different coding)}$$

Find $f(x) = d + h(x)$ with $h \in \mathcal{H}_K$ to min

$$\frac{1}{n}\sum_{i=1}^{n}(1 - y_i f(x_i))_+ + \lambda \|h\|^2_{\mathcal{H}_K} \qquad (**)$$

where $(\tau)_+ = \tau, \tau > 0, = 0$ otherwise.

Then

$$f_\lambda(x) = d + \sum_{i=1}^{n} c_i K(x, x_i). \qquad (*)$$

Substitute (*) into (**), choose λ, given λ, find c and d. The classifier is

$$f_\lambda(x) > 0 \to \mathcal{A}$$

$$f_\lambda(x) < 0 \to \mathcal{B}$$

♣♣ Comparison of the penalized log likelihood estimate f_λ of the log odds ratio $\log p/(1-p)$ and f_λ, the SVM classifier:

Suspicion: They are related...

Let us relabel y in the likelihood –

$$\tilde{y} = \begin{cases} +1 & \text{if } \mathcal{A}, \\ -1 & \text{if } \mathcal{B}. \end{cases}$$

Then

$$-yf + \log(1+e^f) \to \log(1+e^{-\tilde{y}f})$$

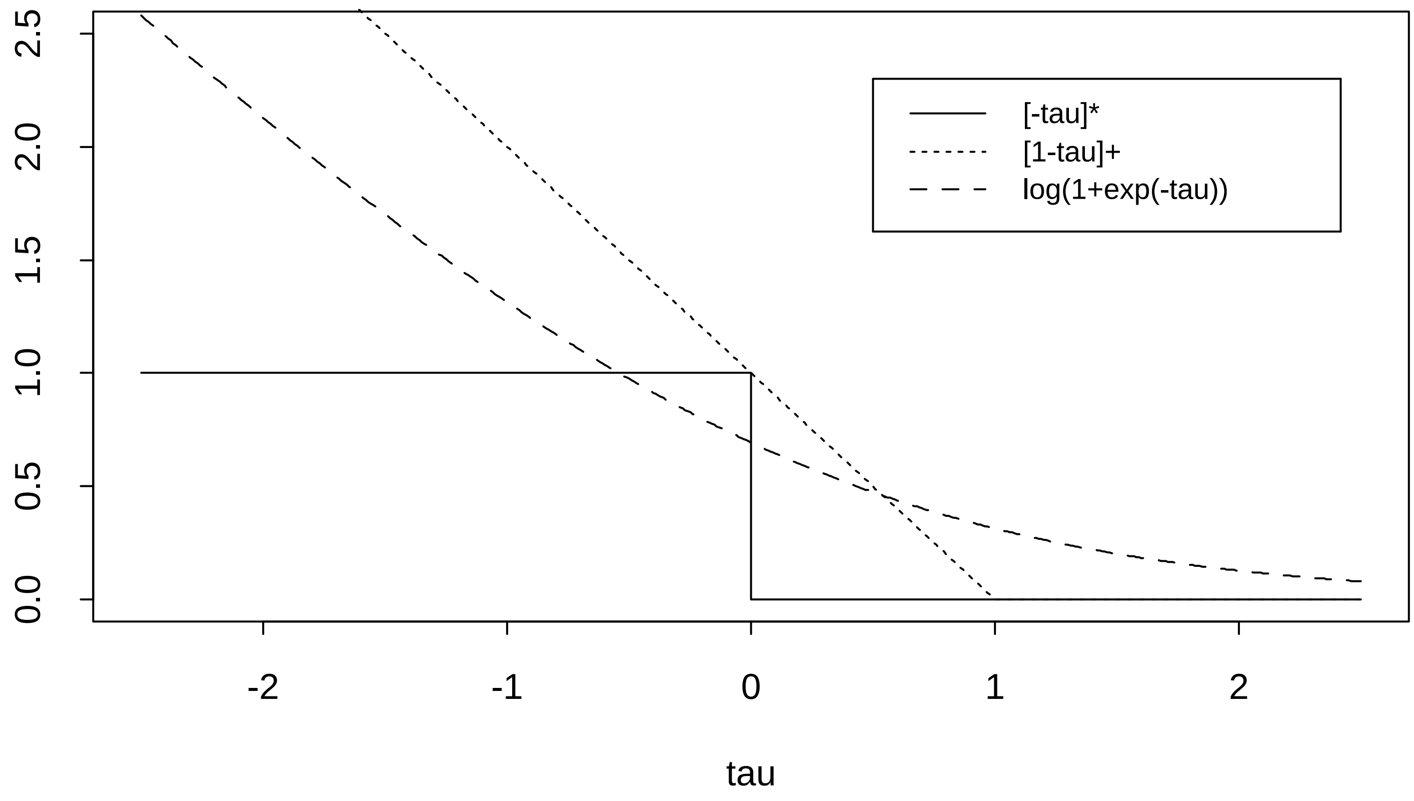
Figure 1 compares

$$\log(1+e^{-yf}), \quad (1-yf)_+ \text{ and } (-yf)_*$$

where

$$(\tau)_* = \begin{cases} 1 & \text{if } \tau > 0, \\ 0 & \text{otherwise.} \end{cases}$$

($(-yf)_*$ is the misclassification counter).



Adapted from [Wahba99]. Comparison of $(-\tau)_*$, $(1-\tau)_+$ and $log_e(1+e^{-\tau})$. Bin Yu observed at the talk that $log_2(1+e^{-\tau})$ goes through 1 at $\tau = 0$. Any strictly convex function that goes through 1 at $\tau = 0$ will be an upper bound on the missclassification function and will be a looser bound than some SVM function.