

# Support Vector Machines

## Some Perspectives from Probability and Statistics

Imperial College Statistics Seminars, 16<sup>th</sup> November,  
2001

Robert Burbidge, Statistics,  
Imperial College

`r.burbidge@ic.ac.uk`

`http://stats.ma.ic.ac.uk/~rdb`

`http://www.cs.ucl.ac.uk/staff/r.burbidge`

# Overview

The Idea of the Support Vector Machine

Support Vector Classification

Support Vector Regression

Why Does It Work?

Summary

# The Idea of the Support Vector Machine

A learning system for classification and regression

—also, density estimation, non-linear principle components, time series prediction, outlier detection, clustering, . . .

Hypothesis space: linear functions in a high-dimensional space

—includes a wide range of well-known function classes from statistics and machine learning

Learning algorithm: quadratic program

—hence, no local optima; can also be formulated as a linear programme

Learning bias: statistical learning theory

—gives bounds on the expected performance; can also be motivated from a Bayesian perspective

Performance: better than most other systems for a wide range of applications

—handwritten digit recognition, bioinformatics, charmed quark detection, chemoinformatics, document classification, . . .

# Support Vector Classification

Binary Classification

Generalized Linear Discriminants

Consistency

'No Free Lunch'

Optimal Separating Hyperplane

Margin Maximization

Kernels

Non-Parametric Density Estimation

Support Vectors

More Kernels

Posterior Probabilities

# Binary Classification

Training data drawn i.i.d. from (unknown)  $p(\mathbf{x}, y)$ ,  
 $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)\}$ ,  $\mathbf{x}_i \in \mathbb{R}^d$ ,  $y_i \in \{-1, +1\}$ .

Given a new example  $\mathbf{x}$ , classify according to,

$$y = \operatorname{argmax}_{y \in \{-1, +1\}} p(y|\mathbf{x})$$

Equivalently, by Bayes' rule,

$$y = \operatorname{argmax}_{y \in \{-1, +1\}} p(\mathbf{x}|y)p(y)$$

One approach is to estimate the class-conditional densities  $p(\mathbf{x}|y)$  and the priors  $p(y)$  from the training data

# Generalized Linear Discriminants

For classification, only the relative probabilities are required

$$y = \text{sgn}(p(+1|\mathbf{x}) - p(-1|\mathbf{x})) = \text{sgn}(p(+1|\mathbf{x}) - \frac{1}{2})$$

Could simply try to estimate by,

$$y = g(\mathbf{x}) = \text{sgn}(f(\mathbf{x})) = \text{sgn}(\langle \mathbf{w}, \psi(\mathbf{x}) \rangle_{\mathcal{H}} + b)$$

This is a *perceptron*, i.e. one learns an hyper-plane in some high-dimensional space given by the mapping

$$\begin{aligned} \mathcal{X} &\rightarrow \mathcal{H}, \\ \mathbf{x} &\rightarrow \psi(\mathbf{x}). \end{aligned}$$

$f$  is the real-valued output (i.e. unthresholded), e.g. for logistic regression, estimate

$$f(\mathbf{x}) = \log \frac{p(+1|\mathbf{x})}{1 - p(+1|\mathbf{x})}$$

by some arbitrary

$$f(\mathbf{x}) = h(\mathbf{x}) + b.$$

If  $\mathcal{H}$  is finite dimensional then  $\langle \psi(\mathbf{x}), \psi(\mathbf{z}) \rangle = \psi(\mathbf{x})^T \psi(\mathbf{z})$ , the standard inner product in a Euclidean space. The  $\psi$  may be thought of as basis functions centred on the training data (c.f. potential functions).

If  $\mathcal{H}$  is countably-infinite dimensional then  $\langle \cdot, \cdot \rangle$  is the inner product in the  $l^2$  space, i.e.  $\langle \psi(\mathbf{x}), \psi(\mathbf{z}) \rangle = \sum_{i=1}^{\infty} \psi_i(\mathbf{x})\psi_i(\mathbf{z})$ . However, we shall later see that this  $l^2$  space can be thought of as Hilbert space composed of linear combinations of  $K(\mathbf{x}, \cdot) = \langle \psi(\mathbf{x}), \cdot \rangle$  with inner product

$$\langle K(\mathbf{x}, \cdot), K(\mathbf{z}, \cdot) \rangle = K(\mathbf{x}, \mathbf{z})$$

i.e. a *reproducing kernel Hilbert space* with kernel  $K$ .

$\mathcal{H}$  may also be uncountably-infinite dimensional.



# Consistency

The hyperplane  $(\mathbf{w}, b)$  should be chosen to minimize the expected risk; *empirical risk minimization* (ERM) aims to minimize the risk on the training data; by the law of large numbers

$$R_{\text{emp}}(f) \rightarrow R(f) \text{ as } l \rightarrow \infty$$

This does not imply that, for  $f^l$  minimizing the empirical risk and  $f^{\text{opt}}$  minimizing risk, the following holds

$$R_{\text{emp}}(f^l) \rightarrow R(f^{\text{opt}}), \quad R(f^l) \rightarrow R(f^{\text{opt}}).$$

If the above do hold, then ERM is said to be *consistent*

For consistency, it is necessary to limit the size of the hypothesis space

The first equation states: the empirical risk of a function tends to the true risk of that function.

The second equation states: the empirical risk of the ERM solution tends to the true risk of the optimal solution.

The third equation states: the true risk of the ERM solution tends to the true risk of the optimal solution.

# 'No Free Lunch'

Consider a machine that can implement *all* functions from  $\mathbb{R}^d$  to  $\{-1, +1\}$

Given a test set

$$\{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_l\}, \bar{\mathbf{x}}_j \in \mathbb{R}^d$$

such that

$$\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)\} \cap \{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_l\} = \emptyset,$$

for any function  $f$  there exists a function  $f^*$  such that

$$\begin{aligned} f^*(\mathbf{x}_i) &= f(\mathbf{x}_i) \forall i, \\ f^*(\bar{\mathbf{x}}_j) &\neq f(\bar{\mathbf{x}}_j) \forall j. \end{aligned}$$

This is an extreme example of *overfitting*

This example is taken from Schölkopf, Burges and Smola (1999).

On the training set the two hypotheses agree, whereas on the test set they give opposite answers.

It is necessary to restrict the size of the hypothesis space. Here follows a load of probabilistic arguments that I've left out. The next slide gives a woolly argument (that is intuitively pleasing).

# Optimal Separating Hyperplane

A *separating hyperplane* w.r.t. the training set  $S$  is a pair  $(\mathbf{w}, b)$  such that

$$(\forall (\mathbf{x}_i, y_i) \in S)(y_i(\langle \mathbf{w}, \psi(\mathbf{x}_i) \rangle_{\mathcal{H}} + b) > 0)$$

The *margin* of a separating hyperplane is the distance from  $\{\psi(\mathbf{x}) \mid \langle \mathbf{w}, \psi(\mathbf{x}) \rangle_{\mathcal{H}} + b = 0\}$  to the image  $\psi(\mathbf{x})$  of the nearest training example

ERM is consistent iff the margin is 'large'

The *optimal separating hyperplane* is that separating hyperplane for which the margin is maximal

A *canonical* hyperplane is one for which  $\min_{\mathbf{x}_i \in \mathcal{R}^d} |\langle \mathbf{w}, \psi(\mathbf{x}_i) \rangle_{\mathcal{H}} + b| = 1$

The margin of a canonical hyperplane is

$$\min_{\mathbf{x}_i \in \mathcal{R}^d} \left| \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \mathbf{x}_i \right\rangle + \frac{b}{\|\mathbf{w}\|} \right| = \frac{1}{\|\mathbf{w}\|}.$$

The first equation states: all of the training examples are correctly classified.

If we fix the margin beforehand (usually at unity), then we have restricted the size of the hypothesis space (by setting an upper limit on  $\|\mathbf{w}\|$ ). If we then find a separating hyperplane, we can expect it to do well, since it is less likely that we found such an hyperplane by chance. These arguments are made rigorous by means of VC-theory and Chernoff bounds.

The definition of canonical hyperplane removes the scaling freedom (so that the solution is unique (except in pathological cases)).

The O.S.H. is not necessarily optimal w.r.t. the risk.

Maximizing the margin is in practice achieved by fixing the (functional) margin at unity and minimizing  $\|\mathbf{w}\|_q$ , c.f. weight decay in neural networks.

If the margin norm is taken to be  $p$ , then the weight norm minimized is  $q : 1/p + 1/q = 1$ . In particular,  $p = 1, \infty$  lead resp. to  $q = \infty, 1$ , which results in a linear programming problem. The standard SVM takes  $p = q = 2$ .

# Margin Maximization

Introduce *slack variables*  $\xi_i$  to allow for training errors; then the O.S.H. (given an error weight of  $C$ ) is found by solving the following Q.P.

$$\text{Minimize}_{\mathbf{w}, b, \xi} \Phi(\mathbf{w}, \xi) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i$$

subject to

$$y_i(\langle \mathbf{w}, \psi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, l.$$

This is equivalent to maximizing the dual

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle_{\mathcal{H}},$$

subject to

$$\sum_{i=1}^l y_i \alpha_i = 0, \quad C \geq \alpha_i \geq 0, \quad i = 1, \dots, l.$$



O.S.H. = optimal separating hyperplane  
 $C$  is a predetermined regularization constant  
Q.P. = quadratic program

The first term is a regularizer and the second term is ER, this is known as regularized learning or structural risk minimization (see later).

The ER term can be split into two terms in the case of different misclassification costs, or unbalanced training data (i.e. empirical prior different to true prior).

The constraints enforce separability of the training data. Note that both errors  $\xi \geq 1$  and *margin errors*  $0 < \xi < 1$  are penalized. This error function is actually the closest convex error function to the misclassification rate itself (see later). It is also possible to penalize the errors quadratically.

The  $\alpha_i$  are the dual variables (Lagrange multipliers), and indicate how much influence each training point  $\mathbf{x}_i$  has on the solution. Typically, many  $\alpha_i$  are zero, resulting in a sparse solution.  $C$  limits the influence any single point may have.

This is a convex Q.P. with linear constraints, hence there are no local optima. The Q.P. is large, new algorithms have been developed to tackle it, empirically  $O(l^2)$ .

The important thing to note here is that the training data only occur in the Q.P. as inner products  $\langle \psi(\mathbf{x}), \psi(\mathbf{z}) \rangle$ .

# Kernels

At the solution to the Q.P. the weight vector is given by

$$\mathbf{w} = \sum_{i=1}^l \alpha_i^* y_i \psi(\mathbf{x}_i)$$

which yields a decision function

$$g(\mathbf{x}) = \text{sgn} \left( \sum_{i=1}^l \alpha_i^* y_i \langle \psi(\mathbf{x}), \psi(\mathbf{x}_i) \rangle + b^* \right).$$

The data  $\mathbf{x}_i$  and test point  $\mathbf{x}$  occur in the Q.P. and decision function only as inner products in  $\mathcal{H}$ ; hence, define a *kernel*

$$K(\mathbf{x}, \mathbf{z}) = \langle \psi(\mathbf{x}), \psi(\mathbf{z}) \rangle_{\mathcal{H}}$$

The mapping to  $\mathcal{H}$  need not be explicitly carried out;  $K$  can be chosen to mimic well-known classifiers

The  $\alpha_i^*$  and  $b^*$  are the solution to the Q.P. Note that  $b^*$  is not guaranteed to be Bayes optimal (except asymptotically), due to the inductive bias (see e.g. Friedman, 'boundary bias').

If  $K$  is a continuous symmetric kernel for a positive integral operator then  $K(\mathbf{x}, \mathbf{z})$  is an inner product in some Hilbert space  $\mathcal{H}$  (Mercer, 1909) (technically,  $\mathcal{H}$  is the RKHS defined by  $K$ ).

The high-dimensional images of the data need not be calculated or stored, this allows infinite-dimensional  $\mathcal{H}$ .

## SVC — Recap

Support Vector Classification consists of

- Choosing an hypothesis from a function class defined by  $K$
- Minimizing a regularized risk

But: the kernel also implicitly defines the type of regularization (see later)

# Non-Parametric Density Estimation

Histogram

Kernel Estimators

Series Expansions

## Histogram

Partition  $\mathbb{R}^d$  into  $\Gamma_1, \dots, \Gamma_s$  and define  $g_k(\mathbf{x}) = 1_{[\mathbf{x} \in \Gamma_k]}$  then

$$\hat{p}(\mathbf{x} | + 1) = \frac{1}{l+1} \sum_{k=1}^s \sum_{i=1}^{l+1} g_k(\mathbf{x}_i) g_k(\mathbf{x})$$

so that

$$g(\mathbf{x}) = \text{sgn} \left( \frac{1}{l} \sum_{i=1}^l y_i K(\mathbf{x}_i, \mathbf{x}) \right)$$

where  $K(\mathbf{x}, \mathbf{z}) = \sum_{k=1}^s g_k(\mathbf{x}) g_k(\mathbf{z})$  is the inner product in the group indicator space  $\{0, 1\}^s$

This assumes that we are taking the priors to be  $\hat{p}(c) = \frac{l_c}{l}$ ,  $c \in \{-1, +1\}$ , if not, then the summand could be weighted by some  $\alpha_i$  for alternative estimators of the prior.



# Kernel Estimators

$$\hat{p}(\mathbf{x} | + 1) = \frac{1}{l+1} \sum_{i=1}^{l+1} K(\mathbf{x} - \mathbf{x}_i)$$

so that

$$g(\mathbf{x}) = \text{sgn} \left( \frac{1}{l} \sum_{i=1}^l y_i K(\mathbf{x} - \mathbf{x}_i) \right)$$

with, for example,

$$K(\mathbf{x} - \mathbf{z}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\|\mathbf{x}-\mathbf{z}\|^2/\sigma^2}$$

so that  $f(\mathbf{x})$  is a radial basis function network classifier with centres on the data points; this corresponds to an implicit mapping

$$\begin{aligned} \mathcal{X} &\rightarrow \mathcal{H}, \\ \mathbf{x} &\rightarrow \psi(\mathbf{x}) \end{aligned}$$

into a countably infinite dimensional space

For specific  $K$  this formulation also leads to Parzen windows

For RBFs,  $K$  defines a regularizer on the smoothness in the Fourier domain (Poggio and Girosi).

It is possible, but messy, to formulate  $k - nn$  in this way, but the resultant kernel is not pos. def.

# Series Expansions (e.g. Fourier, wavelets)

Estimate the (1-dimensional) density by a (truncated) orthogonal series expansion

$$\begin{aligned}\hat{p}(x | + 1) &= \sum_{k=1}^s a_k \psi_k(x) \\ &= \frac{1}{l+1} \sum_{k=1}^s \sum_{i=1}^{l+1} \psi_k(x_i) \psi_k(x)\end{aligned}$$

so that

$$g(x) = \text{sgn} \left( \frac{1}{l} \sum_{i=1}^l y_i K(x_i, x) \right)$$

where  $K(x, z) = \sum_{k=1}^s \psi_k(x) \psi_k(z) = \langle \psi(x), \psi(z) \rangle_{\mathcal{H}}$ , and  $H$  is the wavelet, Fourier, etc./ domain.

Could generalize to higher dimensions by  $K(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^d \sum_{k=1}^s \psi_i(x_k^j) \psi_i(z_k^j)$ , but would become computationally intensive

## Recap

Classification by non-parametric density estimation frequently results in a classifier of the form

$$g(\mathbf{x}) = \text{sgn} \left( \frac{1}{l} \sum_{i=1}^l y_i K(\mathbf{x}_i, \mathbf{x}) + b \right)$$

where  $K(\mathbf{x}, \mathbf{z}) = \langle \psi(\mathbf{x}), \psi(\mathbf{z}) \rangle_{\mathcal{H}}$  is an inner product in some feature space ( $b = 0$  in the previous examples)

Equivalently, this can be thought of as a generalized linear discriminant, with basis functions  $\psi$ , centred on the training data

# Support Vectors

The SVM learns a classifier of the form

$$f(\mathbf{x}) = \text{sgn} \left( \frac{1}{l} \sum_{i=1}^l y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \right)$$

That is, a hyperplane in some feature space, implicitly defined by  $K$

Note that the training data (equiv. basis functions) are now weighted by the Lagrange multipliers  $\alpha_i$ ; only those points  $\mathbf{x}_i$  that are misclassified, or are closest to the hyperplane, have  $\alpha_i \neq 0$ ; these points are termed *support vectors*

c.f. potential functions, condensed  $k$ -NN, edited  $k$ -NN

This is a direct result of the explicit regularization (i.e. margin maximization)

The SVs 'support' the hyperplane, in the sense that, if each exerts a force  $y_i\alpha_i$  on the hyperplane, then the forces sum to zero, and the torques sum to zero.

## More Kernels

$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle$  — linear discriminant

$K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^d$  — polynomial

$K(\mathbf{x}, \mathbf{z}) = \frac{1}{1 + e^{v\langle \mathbf{x}, \mathbf{z} \rangle - c}}$  — two-layer MLP

User defined kernels incorporating domain knowledge (e.g. derived from hidden Markov models on protein sequences)

Kernels motivated by regularization theory or priors on the function class

Within the framework of the SVM it is possible to define kernels for specific tasks, or to mimic well-known techniques.



# Posterior Probabilities

Estimate the logit  $f(\mathbf{x}) = \log \frac{p(+1|\mathbf{x})}{p(-1|\mathbf{x})}$  by

$$f(\mathbf{x}) = h(\mathbf{x}) + b$$

and minimize a penalized log likelihood

$$\frac{1}{l} \sum_{i=1}^l \log \left( 1 + e^{-y_i f(\mathbf{x}_i)} \right) + \lambda \|h\|_{\mathcal{H}}^2$$

However, the sparsity of the representation is lost.

Alternatively, fit a sigmoid to the real-valued output

$$\hat{p}(+1|f) = \frac{1}{1 + e^{-af+b}}$$

(this assumes the output of the SVM is proportional to the log odds)

## Fitting the sigmoid:

- maximum likelihood (training set, hold-out, or, cross-validation)
- regularization (i.e. a prior on out-of-sample data, or on  $(a, b)$ )
- create out-of-sample data as training data plus noise (as in Parzen windows)
- re-use training data with non-binary ('Bayesian') targets and take the MAP sigmoid

See Platt(2000) for more details. Sollich (NIPS99, ICANN99) derives probabilities from the perspective of the kernel as the covariance of a Gaussian Process. See later.

# Gaussian Processes

The SVM solution is that  $(\mathbf{w}, b)$  which minimizes

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^l |1 - y_i(\langle \mathbf{w}, \psi(\mathbf{x}) \rangle_{\mathcal{H}} + b)|_+$$

The first term gives a prior  $p(\mathbf{w}) \sim \exp\left(-\frac{1}{2}\|\mathbf{w}\|^2\right)$ .

The values  $a(\mathbf{x}) = \langle \mathbf{w}, \psi(\mathbf{x}) \rangle$  have a joint Gaussian distribution with covariances  $\langle a(\mathbf{x})a(\mathbf{z}) \rangle = \langle \langle \psi(\mathbf{x}), \mathbf{w} \rangle \langle \mathbf{w}, \psi(\mathbf{z}) \rangle \rangle = \langle \psi(\mathbf{x}), \psi(\mathbf{z}) \rangle$

The SVM prior is a Gaussian process prior over  $a$  with covariance function  $K(\mathbf{x}, \mathbf{z})$  (and zero mean).

The second term can be interpreted as the negative log-likelihood of the observed data.

See Sollich (NIPS99,ICANN99) for more details.

The components of  $\mathbf{w}$  are uncorrelated with unit variance. The prior on  $b$  is flat (uninformative, improper). Later work motivates a broad Gaussian as more reasonable.

# SVC — Summary

SVC chooses a classifier from the space  $\mathcal{H}$  defined by the choice of kernel

Overfitting is avoided by maximizing the margin

The kernel also defines the type of regularization

Kernels can be chosen to mimic many well-known function classes from machine learning and statistics

Posterior probabilities can be obtained by post-processing

The SVM prior is a Gaussian process prior

# Support Vector Regression

$\epsilon$ -Insensitive Loss

The Optimization Problem

Ridge Regression

Gaussian Processes

## $\epsilon$ -Insensitive Loss

To carry the idea of a margin over to regression Vapnik introduced his (linear)  $\epsilon$ -insensitive loss function

$$V(y, f(\mathbf{x})) = |y - f(\mathbf{x})|_{\epsilon} = \max(0, |y - f(\mathbf{x})| - \epsilon)$$

Similarly, the quadratic  $\epsilon$ -insensitive loss function is

$$V(y, f(\mathbf{x})) = |y - f(\mathbf{x})|_{\epsilon}^2.$$

The underlying function is approximated to within  $\epsilon$ , any training points lying within the  $\epsilon$  tube do not appear in the solution

This leads to a sparser solution

This allows the concept of the margin to carry over to regression, and hence leads to a sparse solution. SVs lie on the  $\epsilon$ -tube, or outside of it.

Non-zero  $\epsilon$  corresponds to an additional weight decay.



# The Optimization Problem

Similar to the classification case, minimize:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \hat{\xi}_i)$$

subject to

$$\begin{aligned}(\langle \mathbf{x}, \mathbf{x}_i \rangle_{\mathcal{H}} + b) - y_i &\leq \epsilon + \xi_i, \\ y_i - (\langle \mathbf{x}, \mathbf{x}_i \rangle_{\mathcal{H}} + b) &\leq \epsilon + \hat{\xi}_i, \\ \xi_i, \hat{\xi}_i &\geq 0, \quad i = 1, 2, \dots, l.\end{aligned}$$

The  $\xi_i, \hat{\xi}_i$  measure the errors 'above' and 'below' the  $\epsilon$ -tube (hence  $\xi_i \hat{\xi}_i = 0$ )

This is solved in the dual as before to give  $\alpha_i$  and  $\hat{\alpha}_i$  so that

$$f(\mathbf{x}) = \sum_{i=1}^l (\hat{\alpha}_i - \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + b,$$

(Note,  $\alpha_i \hat{\alpha}_i = 0$ .)

# Ridge Regression

The case of  $\epsilon = 0$  with quadratic loss is equivalent to ridge regression

Minimize:

$$\lambda \|\mathbf{w}\|^2 + \sum_{i=1}^l \xi_i^2,$$

subject to:  $y_i - \langle \mathbf{w}, \mathbf{x} \rangle = \xi_i, \quad i = 1, \dots, l.$

Which can be formulated as a Lagrangian and solved to give:

$$f(\mathbf{x}) = \mathbf{y}'(\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{k},$$

where  $\mathbf{k}$  has entries  $k_i = \langle \mathbf{x}, \mathbf{x}_i \rangle_{\mathcal{H}}$ .

# Gaussian Processes

$$p(t, \mathbf{t} | \mathbf{x}, S) \propto p(\mathbf{y} | \mathbf{t}) p(t, \mathbf{t} | \mathbf{x}, \mathbf{X})$$

$$p(\mathbf{y} | \mathbf{t}) \propto \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{t}\|^2\right)$$

$$p(t, \mathbf{t} | \mathbf{x}, \mathbf{X})$$

$$= p_{f \sim \mathcal{D}}[(f(\mathbf{x}), f(\mathbf{x}_1), \dots, f(\mathbf{x}_l)) = (t, t_1, \dots, t_l)]$$

$$\propto \exp\left(-\frac{1}{2} \hat{\mathbf{t}}^T \hat{\Sigma}^{-1} \hat{\mathbf{t}}\right)$$

where  $\hat{\mathbf{t}} = (t, t_1, \dots, t_l)$  and  $\hat{\Sigma}_{00} = K(\mathbf{x}, \mathbf{x})$ ,  $\hat{\Sigma}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ ,  $i, j = 1, \dots, l$ , and  $\hat{\Sigma}_{0i} = \hat{\Sigma}_{i0} = K(\mathbf{x}, \mathbf{x}_i)$ .

The distribution of  $t$  is the predictive distribution. It is Gaussian with mean  $f(\mathbf{x})$  and variance  $V(\mathbf{x})$

$$f(\mathbf{x}) = \mathbf{y}^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}$$

$$V(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}) - \mathbf{k}^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}$$

The solution is the ridge regression solution.

This suggests that, when quadratically penalizing errors, the assumption is one of Gaussian noise with variance  $1/C$ .

Moreover, the G.P. gives an estimate of the reliability of the prediction.

It can also be used to estimate the evidence in favour of a particular kernel and aid principled model selection.

# SVR — Summary

The  $\epsilon$ -insensitive loss is introduced to maintain the idea of the margin, and sparsity of the solution (c.f. also robust regression)

The optimization is again a Q.P. with no local optima

The regularization is the same as for ridge regression, non-zero  $\epsilon$  results in an extra weight decay term

There is a close relationship to Gaussian processes, which can provide reliability estimates and evidence for the model (kernel)

# Why Does It Work?

Statistical Learning Theory

Regularization Networks

Penalized Log-Likelihood

# Statistical Learning Theory

Define a sequence of hypothesis spaces

$$\mathcal{H}_1 \subset \mathcal{H}_2 \subset \dots \subset \mathcal{H}_m$$

with increasing capacity (VC-dimension)

$$h_1 \leq h_2 \leq \dots \leq h_M$$

by defining  $\mathcal{H}_m = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq A_m\}$  where  $A_1 < A_2 < \dots < A_M$ .

Minimize the empirical risk on each subset and choose that solution  $f_m^l$  which minimizes an upper bound on the true risk

$$R(f) < R_{\text{emp}}(f) + \Phi\left(\sqrt{\frac{h}{l}}, \eta\right)$$

which holds with probability  $1 - \eta$ .

This is achieved by minimizing

$$R_{\text{reg}}(f) = R_{\text{emp}}(f) + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2$$

The selection of the hypothesis space  $\mathcal{H}_m$  corresponds to choosing  $\lambda$ , which is done by, e.g. cross-validation



# Regularization Networks

Regularization networks minimize

$$R_{\text{reg}}(f) = R_{\text{emp}}(f) + \frac{\lambda}{2} \|Pf\|^2$$

where  $P$  is a regularization operator

SVR has a kernel  $K$  such that

$$K(\mathbf{x}, \mathbf{z}) = \langle (PK)(\mathbf{x}, \cdot), (PK)(\mathbf{z}, \cdot) \rangle.$$

$K$  is taken to be a Green's function of  $P^*P$ .

Thus the kernel defines the function class and the regularization.

The loss functions also differ.

In the general case the sparsity is lost.

# Penalized Log-Likelihood

Estimate the logit  $f(\mathbf{x}) = \log \frac{p(+1|\mathbf{x})}{p(-1|\mathbf{x})}$  by

$$f(\mathbf{x}) = h(\mathbf{x}) + b, \quad h \in \mathcal{H}$$

Minimize:

$$\frac{1}{l} \sum_{i=1}^l \log \left( 1 + e^{-y_i f(\mathbf{x}_i)} \right) + \lambda \|h\|_{\mathcal{H}}^2$$

Has solution (Kimeldorf and Wahba, 1971):

$$f_{\lambda}(\mathbf{x}) = \sum_{i=1}^l c_i K(\mathbf{x}, \mathbf{x}_i) + b$$

The aim is to minimize the misclassification rate; the SVC loss function is the closest convex function to the misclassification rate.

(Lin, 2000) The minimizer of  $E_{\mathcal{D}}(1 - yf(\mathbf{x}))_+$  is  $\text{sgn}(p(+1|\mathbf{x}))$ ; thus, if  $\mathcal{H}$  is rich enough then SVC is implementing the Bayes rule (for the  $\lambda$  optimizing the GCKL)

Caveat:  $f(\mathbf{x}) = \langle \mathbf{w}, \psi(\mathbf{x}) \rangle + b$  does not approximate  $p(+1|\mathbf{x}) - \frac{1}{2}$ .

# Summary

The SVM is tool for classification and regression

The solution is found by minimizing a regularized loss

The kernel defines both the function class and the type of regularization

The kernel can be chosen to mimic many well-known techniques from machine learning and statistics

The kernel can be interpreted as the covariance of a Gaussian process — this can give reliability estimates and evidence for the model

The true risk can be bounded without the need for hold-out data — thus model selection is

easier (there are very few free parameters in any case)

The choice of loss function leads to sparse models

# References