Classification: Support Vector Machine

Xiang Zhou

School of Data Science
Department of Mathematics
City University of Hong Kong

SVM: support vector machine

Mainly developed and was dominantly hot in computer science / pattern recognition

- Vapnik 1995: Geometric Viewpoint + Primal-Dual for Quadratic Programming (+ Kernel trick, new def of metric)
- Major developments throughout 1990's
- Has good generalization properties;
- One of the most important and successful developments before deep learning.

Method	main properties
maximal margin classifier	only for linear separable dataset
support vector classifier	slack variable, linear classifier
support vector machine	kernel trick, nonlinear classifier

Table: Development of SVM

1

2

[1] Sollich 2002: Bayesian Viewpoint of SVM

Linear Separation for binary classification

Binary classification problem: dataset $\{x_i, y_i\}$ where $y_i \in \mathcal{Y} = \{-1, 1\}$.

- Logistic regression assumes: the log odd, $\log h(x) = \log \mathbb{P}(Y = +|X = x)$, is linear in x. The decision boundary is the level set of $h(x) = \beta \cdot x = 0.5$
- The LDA's the discriminant function $\delta(x)$ is also linear in x.
- SVM is also a linear classifier, with a strong geometric intuition.

Summary:

- The logistic regression = sigmoid activation function + linear feature assumption + maximum likelihood
- The linear discriminant analysis (LDA) = Bayes classifier + Gaussian mixture + common variance assumption
- The support vector machine (SVM) = linear classifier + max margin + slack variable (+ kernel trick)

Note the notations different from logistic regressions:

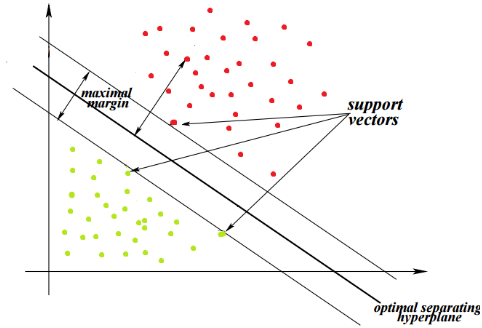
- $\mathcal{Y} = \{-1, 1\}$ (or $\{+, -\}$), not $\{0, 1\}$
- the discriminant function is generally denoted by f. The classifier $G(x) := \text{sign} f(x) \in \{-1, 1\}$. Then decision boundary is $f(x) = 0$, not $h(x) = 0.5$.

This set of notation is convenient because if y belong to $\{+, -\}$

$$\boxed{\text{sign} f(x) = y \iff y f(x) > 0.}$$

Note $\text{sign} f(x) = \text{sign}(\lambda f(x))$ for any $\lambda > 0$.

optimal separating hyperplane



Exercise

A linear function is $f(x) = w \cdot x + b$. *Given a point* x^*, *show the signed distance between* x^* *and the hyperplane* $f(x) = 0$ *is*

$$f(x^*)/\|w\|$$

The positive sign of $f(x^*)$ *means that* x^* *on the same side of the hyperplane as the normal direction vector* w.

Given one data example (x_i, y_i), if f correctly classifies x_i, then $\text{sign} f(x_i) = y_i$. Define

$$M_i := y_i \frac{f(x_i)}{\|w\|}$$

which is the (signed) **margin** of x_i and y_i to the separating hyperplane.

- f correctly classifies $x_i \iff M_i > 0$.
- A larger (positive) value of M_i indicates a large distance to the decision boundary: a large confidence of correctness in classification for (x_i, y_i).

Definition (margin)

Given the dataset $(x_i, y_i), i = 1, \ldots, n$ and a linear function $f(x) = w \cdot x + b$, then the margin [a] of the dataset $(x_i, y_i), i = 1, \ldots, n$ to the hyperplane $f(x) = 0$ is

$$M = \min_{1 \le i \le n} M_i = \min_i \left\{ y_i(w \cdot x_i + b) / \|w\| \right\}$$

The **support vectors** are the collection of $\{x_j\}$ such that $M = y_j(w \cdot x_j + b)$.

---

[a] Sometimes, the margin refers to $2M$, the distance between the two hyperplanes $w \cdot x + b = \pm M / \|w\|$.

- M depends on w, b and the dataset $\{x_i, y_i\}$.
- If there exists a linear function $f = w \cdot x + b$ such that $M > 0 \iff$ the dataset is linearly separable, i.e.

$$\mathsf{sign} f(x_i) = y_i, \forall i.$$

- The margin M is a function of w and b; we consider its maximal value over the choice of w and b:

$$M^* := \max_{w,b} M(w, b) = \max_{w,b} \left(\min_i \{y_i(w \cdot x_i + b)/\|w\|\} \right) \quad (1)$$

- If M^* is positive, then the corresponding optimal w^* and b^* : the dataset is linearly separabble
- If M^* is negative, then for any w and b, $M < 0$, i.e., there always exists some data misclassified by f. The dataset is not linearly separable.

Maximal Margin Classifier

Write the max-min problem (1) as follows:

Definition (maximal margin classifier)

The maximal margin classifier solves the problem

$$\begin{aligned} &\max_{w\in\mathbb{R}^d, b\in\mathbb{R}} \quad M \\ &\text{subject to} \quad y_i(w\cdot x_i + b)/\|w\| \geq M, \quad i = 1, 2, \ldots, n \end{aligned} \tag{2}$$

- The equivalent form of maximal margin classifier is

$$\begin{aligned} &\max_{w\in\mathbb{R}^d, b\in\mathbb{R}} \quad M \\ &\text{subject to} \quad \|w\| = 1 \\ &\qquad\qquad\quad y_i(w\cdot x_i + b) \geq M, \forall i \end{aligned} \tag{3}$$

- The constraint $\|w\| = 1$ is only for the uniqueness of w and b; without this constraint, the solution is a family of the linear discriminant functions $\{\lambda f^*(x) : \lambda > 0\}$, which all share the **same** classifier sign f^*.

Exercise (XOR)

Suppose the dataset has $n = 4$ *examples in* $\mathbb{R}^2$ *plane as follows:*

$$\begin{array}{ll} x_1=(1,-1) & y_1=-1 \\ x_2=(1,1) & y_2=1 \\ x_3=(-1,1) & y_3=-1 \\ x_4=(-1,-1) & y_4=1 \end{array}$$

. Find the maximal margin classifier

$f(x) = w_1 x_{(1)} + w_2 x_{(2)} + b$ *where* $x = (x_{(1)}, x_{(2)}) \in \mathbb{R}^2$

$$\max_{w \in \mathbb{R}^d, b \in \mathbb{R}} \quad M$$

$$\begin{aligned} \text{subject to} \quad & w_1^2 + w_2^2 = 1 \\ & w_1 + w_2 + b \geq M \\ & -w_1 - w_2 + b \geq M \\ & w_1 - w_2 - b \geq M \\ & -w_1 + w_2 - b \geq M \end{aligned}$$

The constraints are equivalent to $|w_1 + w_2| \leq -M + b$ and $|w_1 - w_2| \leq -M - b$. Then $|w_1| \leq -M$. So any admissible M is negative. It is easy to show that $M \pm b \leq 0$. So the possible max of M is $M = b$ or $M = -b$. If $M = b$, then $w_1 = -w_2 = \pm b$ and $f(x) = (-x_1 + x_2 \pm 1)/\sqrt{2}$. If $M = -b$, then $w_1 = w_2 = \pm b$ and the solution is $f(x) = (-x_1 - x_2 \pm 1)/\sqrt{2}$

Exercise

Ex. 4.7. [ESL] Consider the averaged margin of M_i, not the minimal one: $\bar{M} = \frac{1}{N}\sum_{i=1}^{N} M_i = \frac{1}{N}\sum_{i=1}^{N} y_i(w \cdot x_i + b)/\left\|w\right\|$*. Solve the problem*

$$\max_{w,b} \bar{M}.$$

Cover theorem: high input dimensionality improves linear separability

Cover's theorem:

> *"pattern-classification problem cast in a high dimensional space non-linearly is more likely to be linearly separable than in a low-dimensional space"*

Kernel trick:

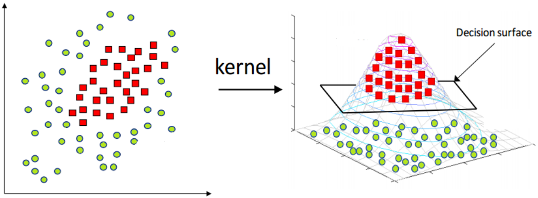


image source

Linearly Separable case

Recall in the maximal margin classifier (2), $y_i(w \cdot x_i + b) \geq \|w\| \, M$. Since we can scale w, b by a **positive** factor arbitrarily, we can assume $M > 0$ and use the normalization $M \|w\| = 1$ *if the dataset is linearly separable*, instead of using the normalization $\|w\| = 1$. Then (2) becomes

$$\begin{aligned} &\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2} \|w\|^2 \\ &\text{subject to } \boxed{y_i(w \cdot x_i + b) \geq 1, \forall i} \end{aligned} \tag{4}$$

- Now there is NO admissible solution if the dataset is not linearly separable.
- The margin M is recover as $\frac{1}{\|w\|}$.
- The problem (4) is the standard quadratic programming problem ☺.
- The support vectors are those on the two hyperplanes

  $$\boxed{w \cdot x + b = \pm 1},$$

  i.e., the inequality constraints at these support vectors actually are equalities.

Linear separation assumption is too strong and hard to verify in practice.

The non-separable case means there are some examples (x_i, y_i) such that $y_i(w \cdot x_i + b) < 0$. Then by adding n slack variables $\xi = (\xi_1, \ldots, \xi_n)$, we have the support vector classifier

Definition (support vector classifier)

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}, \xi \in \mathbb{R}^n} \frac{1}{2} \|w\|^2 \tag{5}$$

$$\text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \forall i \tag{6}$$

$$\xi_i \geq 0, \forall i \tag{7}$$

$$\sum_{i=1}^{n} \xi_i \leq s \tag{8}$$

where the constant $s > 0$ is a tuning parameter.

Totally, $d + 1 + n$ unknowns.

Remarks on relaxation budget

- hyperparameter s controls the budget of relaxation:
  - $s = 0 \iff \xi_i \equiv 0$, maximal margin classifier becomes (4)(for linearly separable case) and does not allow violation of the margin.
  - If $s = +\infty$, any w and b are admissible, and then the optimal $w^* = 0$, b arbitrary: huge bias, low variance
- The budget $\sum_i \xi_i < s$ can be rewritten as the penalty form with a tuning cost parameter $C > 0$

Definition (SVC)

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}, \xi \in \mathbb{R}^n} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \tag{9}$$

$$\text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \forall i \tag{10}$$

$$\xi_i \geq 0, \forall i \tag{11}$$

Interpretation of slack variables at optimality

There are two constraints:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \text{and } \xi_i \geq 0$$

At the optimal parameters, some constraints may be active, some others may be inactive. With the abuse of language, we define "margin" as the set between two hyperplanes $\mathcal{M} := \{x : |w \cdot x + b| \leq 1\}$

- $\xi_i = 0$, then x_i is correctly classified and furthermore it is not inside of the margin $\mathcal{M}$ (not violating the margin).
- if $\xi_i = 0$ and $y_i(w \cdot x_i + b) = 1$, then x_i is a support vector, sitting on the boundary of margin.
- $\xi_i > 0$, then $y_i(w \cdot x_i + b) = 1 - \xi_i < 1$, x_i violates the margin and enters $\mathcal{M}$;
  - $0 < \xi < 1$: then $1 - \xi_i > 0$ so x_i is still correctly classified but too close to the decision boundary.
  - $\xi_i > 1$: $y_i(w \cdot x_i + b)$ is negative and x_i is misclassified.

[1]

---

[1] The theory of constrained optimization such as linear programming theory is helpful in understanding this part.

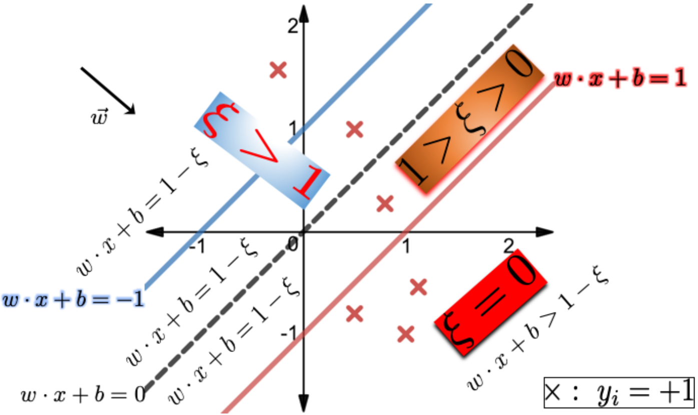


Figure: The role of slack variables. Only the data points in class $+1$ shown with markers("$\times$"). $x \in \mathbb{R}^2$.

The larger value of ξ_i, the further the point x_i away from the correct domain. This justifies the penalty of $\sum_i \xi_i$.

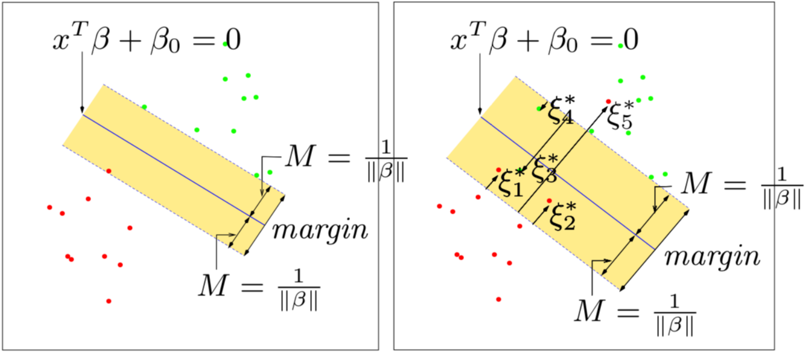


FIGURE 12.1. *Support vector classifiers. The left*

From [ESL]: here $\xi_i^* := M\xi_i$

Exercise: discuss the range of $\xi_i^* = M\xi_i$ in the right figure.

Support Vectors

The support vectors are those x_i such that $y_i(w \cdot x_i + b) = 1 - \xi_i$. This equation of w, b can easily solved if these support vectors as well as ξ_i are known. In fact the solution of SVC only depends on these support vectors:

$$w^* = \sum_{i \in S} \hat{\alpha} y_i x_i$$

where $S = \{i : y_i(w \cdot x_i + b) = 1 - \xi_i\}$. Refer to Section 12.2.1 in [ESL].

- Complexity of training support vector classifier is characterized by the number of support vectors rather than the dimensionality. So it works well on small as well as high dimensional data spaces.
- They are not suitable for larger datasets because the training time with SVMs can be high and much more computationally intensive.
- The solution is insensitive to the outliers (the data points significantly far away from the decision boundary).

SVM with kernel tricks: nonlinear decision boundaries

The key idea of extending linear SVC, and many other linear procedures, to nonlinear is to:

- Enlarge the predictor space using basis expansion functions $h_1(x), \ldots, h_M(x)$
- Construct a linear separating hyperplane $f(x) = w \cdot h(x) + b$ in the enlarged space for better training performance
- The linear separating hyperplane in the enlarged space can be translated into a nonlinear separating hyperplane in the original space.

The procedure is to replace x in SVC by $h(x)$ in SVM. We do not discuss the details further: this is a general principle.

NEXT:
Rewrite the Constraint Optimization form of SVC into the classic form
Loss function + Penalty

Note that two constraints in SVC $y_i f(x_i) \geq 1 - \xi_i$ and $\xi_i \geq 0$ together are equivalent to

$$\xi_i \geq \max\{0, 1 - y_i f(x_i)\} =: (1 - y_i f(x_i))_+.$$

SVM= hinge Loss + L_2-Regularization

Then the SVC in (9) is equivalent to

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}, \xi \in \mathbb{R}^n} \frac{1}{2} \|w\|^2 + C \sum_i \xi_i$$

$$\text{subject to} \quad \xi_i \geq (1 - y_i(w \cdot x_i + b))_+, \forall i$$

which is equivalent to

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \sum_i (1 - y_i(w \cdot x_i + b))_+ + \frac{1}{2C} \|w\|^2 \tag{12}$$

This is the form of (hinge) loss + (L_2) regularization

$$\boxed{\ell_{hinge}(y, f) = (1 - yf)_+, \quad y \in \{-1, 1\}, f \in \mathbb{R}}$$

The population risk is

$$\mathcal{E}(f) = \mathbb{E}\,\ell_{hinge}(Y, f(X))$$

The penalty for the roughness of f : $R(f) = \|f\|$.
When $f(x) = w \cdot x + b$,

$$\min_{w,b} \sum_{i=1}^{n} \ell_{hinge}(y_i, f(x_i)) + \frac{\lambda}{2} \|w\|^2 \tag{13}$$

- $\lambda = 1/C$: the budget for relaxation.
- $\lambda > 0$: the penalty on the margin-relevant variable $\|w\|$.
- The soft margin performs like regularization; so the SVM is usually good at generalization.

logistic regression : binomial deviance loss without Regularization

Recall the logistic regression solves

$$\min_f \mathbb{E}\, \ell_{bd}(Y, f(X)) \approx \frac{1}{n}\sum_{i=1}^{n} \ell_{bd}(y_i, f(x_i)) \tag{14}$$

where the binomial deviance loss

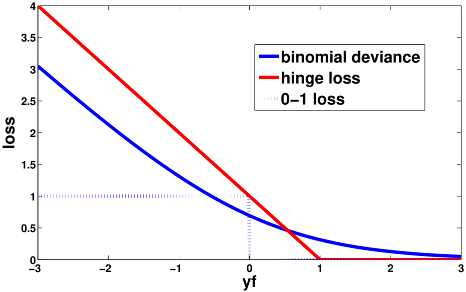
$$\ell_{bd}(y, f) = \mathsf{softplus}(-yf) = \log(1 + e^{-yf}), \quad y \in \{-1, 1\}\,.$$

We know the optimal $f^*_{bd}(x) = \mathsf{logit}(h) = \log \frac{h}{1-h}$ where $h(x) = \mathbb{P}(Y = +1 | X = x)$. The classifier is

$$x \to \mathsf{sign}(f^*_{bd}(x))$$

This is equivalent to the Bayes classifier with the 0-1 loss $\mathbb{E}\, \ell_{01}(Y, G(X))$: $G^*(x) = \mathsf{sign}(h(x) - 0.5) = \mathsf{sign}\, f^*_{bd}(x)$

Then we have three loss functions ℓ_{bd}, ℓ_{hinge} and ℓ_{01}. They are functions in term of the product $yf(x)$. More choice of loss functions are in Table 12.1 [ESL].



<u>discussion</u>: What differences ? Fisher Consistence ? Computational issues ? Which data examples feel the "gradient" force? Read more on surrogate loss function

Exercise

Consider the risk minimization problem for the hinge loss

$$\inf_f \mathbb{E}\, \ell_{hinge}(Y, f(X))$$

in the $\{\pm 1\}$*-encoded binary classification problem. Show that the optimal* f^*_{hinge} *is*

$$\textit{sign}(h(x) - 0.5), \ \textit{where} \ \ h(x) = \mathbb{P}(Y = +1|X = x).$$

Recall a similar exercise for the binomial deviance loss.

$$
\begin{aligned}
&\mathbb{E}\,\ell_{hinge}(Y, f(X)) \\
=&\pi_+\,\mathbb{E}_{X|Y=+1}\,\ell_{hinge}(+1, f(X)) + \pi_-\,\mathbb{E}_{X|Y=-1}\,\ell_{hinge}(+1, f(X)) \\
=&\pi_+ \int_{\mathcal{X}} I(1 - f(x) > 0)\,(1 - f(x))\,\rho_+(x)\mathrm{d}x \\
&\quad + \pi_- \int_{\mathcal{X}} I(1 + f(x) > 0)\,(1 + f(x))\,\rho_-(x)\mathrm{d}x
\end{aligned}
$$

Consider the domain $\Omega_+ \subset \mathcal{X}$ where $f(x) > 1$, then the integration over Ω_+ is $\pi_- \int_{\Omega_+} (1 + f(x))\rho_-(x)\mathrm{d}x$: by decreasing the value of f on this domain Ω_+ to the minimal possible value 1, one has a smaller loss. So, for f^* to be optimal, Ω_+ must be empty. For the same reason for Ω_- case, we deduce that $f^*(x) \in [-1, 1]$ almost everywhere. Then we only consider to minimize within this bounded function class $\{f : |f(x)| \le 1\}$ $\pi_+ \int_{\mathcal{X}} (1 - f(x))\,\rho_+(x)\mathrm{d}x + \pi_- \int_{\mathcal{X}} (1 + f(x))\,\rho_-(x)\mathrm{d}x =$ $\int_{\mathcal{X}} f(x)\,(\pi_-\rho_-(x) - \pi_+\rho_+(x))\,\mathrm{d}x + 1$. So if $\pi_-\rho_-(x) < \pi_+\rho_+(x)$, the minimizer is $f^*(x) = 1$; otherwise $f^*(x) = -1$. Equivalently, $f^*(x) = \mathrm{sign}(\pi_+\rho_+(x) - \pi_+\rho_+(x)) = \mathrm{sign}(h(x) - 0.5)$ since $h(x) = \frac{\pi_+\rho_+(x)}{\pi_+\rho_+(x)+\pi_+\rho_+(x)}$

All three loss functions (binomial deviance , hinge, 0-1) for

$$\inf_f \mathbb{E}\, \ell(Y, f(X))$$

produces the same classifier $x \to \mathrm{sign}(f^*(x))$, which are the Bayes classifier:

$$x \to \mathrm{sign}\left(\mathbb{P}(Y = +|X = x) - 0.5\right)$$

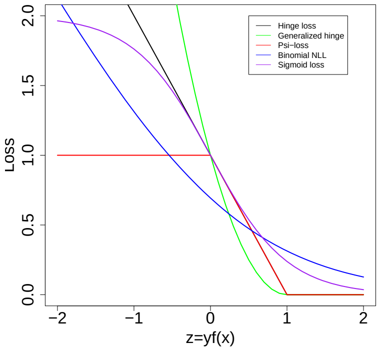
But the minimizers f^* are different. The dynamics of minimizing procedure is also different even f^* are the same.

- Note $\mathrm{sign}(f^*_{hinge}) = f^*_{hinge}$ since f^*_{hinge} only takes value ± 1.
- f^*_{bd} is continuous, but f^*_{hinge} is a step-type function.

Compared with other losses as in Table 12.1,

- Other losses aim to estimate $\mathbb{P}(Y = 1|X = x)$, which contains more information than classification;
- Hinge loss aims to estimate $\mathrm{sign}(\mathbb{P}(Y = 1|X = x) - 1/2)$, which targets on classification directly.

Other loss functions $L(z)$



- Hinge loss (Vapnik, 1995): $L(z) = (1 - z)_+ = \max(1 - z, 0)$
- Generalized hinge loss (Lin, 2002): $L(z) = (1 - z)_+^q$
- ψ-loss (Shen, Tseng, Zhang and Wong, 2003): $L(z) = \psi(z) = \min(1, (1 - z)_+)$
- Binomial deviance (Zhu and Hastie, 2004): $L(z) = \log(1 + e^{-z})$
- Sigmoid loss (Mason, Bartlett and Baxter, 2000): $L(z) = 1 - \tanh(\lambda z)$

P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity , Classification , and Risk Bounds," J. Am. Stat. Assoc., pp. 1–36, 2003

Regularization in SVM smoothes f^*_{hinge}

- The objective function with regularization

$$\inf_{f \in \mathcal{H}} \mathbb{E}\, \ell_{hinge}(Y, f(X)) + \lambda R(f)$$

gives a smooth function approximation f_λ to the step function f^*_{hinge}.
- The sign operation in the final classifier

$$x \to \mathsf{sign}(f_\lambda)$$

places back f_λ to the step function again.
- SVC uses a linear function f to approximate a step function with jumps. The maximal margin $\|w\|$ has the meaning of penalty.

Variants of SVM

Weighted SVM

Different weights (π_+, π_-) are associated with positive and negative cases,

$$\min_f \left(\pi_+ \sum_{y_i=1} L(f(x_i)) + \pi_- \sum_{y_i=-1} L(-f(x_i)) \right)$$

where $L(z) = (1-z)_+$ is the hinge loss and can be other loss functions as well.

Exercise (weighted SVM)

Assume $L(z)$ is hinge loss, and choose $s(Y) = 1 - \pi$ if $Y = +1$, and $s(Y) = \pi$ if $Y = -1$. Prove that

$$\underset{f}{\operatorname{argmin}} \ \mathbb{E}\left(s(Y)L(Yf(X))\right) = \textit{sign}(f_\pi(x))$$

where $f_\pi(x) = \mathbb{P}(Y = +1|x) - \pi$.

- The weighted SVM approximates sign$(f_\pi(x))$
- $f_\pi(x) = 0 \iff \mathbb{P}(Y = +1|x) = \pi$: one can recover the conditional prob. $\mathbb{P}(Y = 1|X = x)$ by using SVM!
- $f_\pi(x)$ and $f^* = \text{sign}(f_\pi(x))$ are nonincreasing in π: for instance, $\mathbb{P}(Y = 1|X = x)$ is between 0.6 and 0.7 for a given x from this table

π	0	0.1	$\cdots$	0.6	0.7	$\cdots$	1
$f^*(x))$	+1	+1	$\cdots$	+1	-1	$\cdots$	-1

Algorithm

- Initialize $\pi_j = (j-1)/m$; $j = 1, \cdots, m+1$.
- Train weighted support vector classifier with the weight parameter being π_j; $j = 1, \cdots, m+1$.
- Estimate labels of x by $f_j^*(x) = \text{sign}(\hat{f}_{\pi_j}(x))$.
- Compute

$$\begin{aligned} \pi^* &= \max\{\pi_j : \text{sign}(\hat{f}_{\pi_j}(x)) = +1\}, \\ \pi_* &= \min\{\pi_j : \text{sign}(\hat{f}_{\pi_j}(x)) = -1\}. \end{aligned}$$

Then (π^*, π_*) or (π_*, π^*) is the interval containing $p(x)$, and thus the estimated class probability is

$$\mathbb{P}(Y = +1 | X = x) \approx \widehat{p}(x) = \frac{1}{2}(\pi^* + \pi_*).$$

Multiclass SVM

Training sample $(x_i, y_i)_{i=1}^n$ with $x_i \in \mathbb{R}^p$ and $y_i \in \{1, \dots, K\}$. Denote $\boldsymbol{f} = (f_1, \dots, f_K)^T$ with

$$\sum_k f_k(x) = 0.$$

The classifier is $x \to \operatorname{argmax}_k f_k(x)$.

$$\min_{\boldsymbol{f}} \sum_{i=1}^n L(y_i, \boldsymbol{f}(x_i))$$

- combine multiple binary classifiers: One-vs-one ($K(K-1)/2$ classifiers) or one-vs-rest (K classifiers)
- Simultaneous approaches:
  - (Vapnik, 1998): $L(y, \boldsymbol{f}(x)) = \sum_{k \neq y} \Big(1 - (f_y(x) - f_k(x))\Big)_+$
  - (Crammer and Singer, 2001): $L(y, \boldsymbol{f}(x)) = \Big(1 - \min_k (f_y(x) - f_k(x))\Big)_+$;
  - (Lee et al., 2004; JASA) $L(y, \boldsymbol{f}(x)) = \sum_{k \neq y} (1 + f_k(x))_+$

Exercise (naive hinge loss for multi-class)

The minimizer $\boldsymbol{f}^*(x)$ *of*

$$\inf_{\boldsymbol{f}} \mathbb{E}[(1 - f_Y(X))_+ | X = x]$$

subject to $\sum_k f_k(x) = 0$ *is*

$$f_k^*(x) = \begin{cases} 1 - K & \textit{if } k = \operatorname{argmax}_y \mathbb{P}(Y = y | X = x) \\ 1 & \textit{otherwise} \end{cases}$$

This shows that the naive hinge loss is **not Fisher consistent** since the classifier $\operatorname{argmax}_k f_k^*(x)$ is not the Bayes classifier $\operatorname{argmax}_y \mathbb{P}(Y = y | X = x)$

solution

Support vector regression

SVR is formulated as:

$$\begin{aligned} \min_{f \in \mathcal{H}_K} \quad & \sum_{i=1}^{n} \xi_i + \frac{\lambda}{2} \|f\|^2_{\mathcal{H}_K} \\ \text{subject to} \quad & |y_i - f(x_i))| \leq \epsilon + \xi_i;\ \xi_i \geq 0;\ i = 1, \ldots, n. \end{aligned}$$

Some remarks:

- Similar as robust regression
- Require additional effort for the optimization
- ϵ is an additional tolerance parameter that needs to be tuned

One-class SVM

One-class SVM looks for the minimal hypersphere containing all points in feature space, while incorporating outliers in the solution

$$
\begin{aligned}
\min_{\mu,R,\xi} \quad & R^2 + C\sum_{i=1}^{n}\xi_i \\
\text{subject to} \quad & \|x_i - \mu\|^2 \le R^2 + \xi_i;\ \xi_i \ge 0;\ i = 1, \ldots, n.
\end{aligned}
$$

Applications:

- Change point/outlier detection
- Cluster analysis

Important topics not touched here for SVM

- kernel trick, RKHS. Section 12.3.1.
- optimization theory and methods for SVM Section 12.2.1

Other references:

- C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", 1998
- P.S. Sastry, "An Introduction to Support Vector Machine"
- J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines", 1999