9.520: Class 14

Support Vector Machines for Classification

Alessandro Verri

About this class

Theme Support Vector Machines for classification are introduced and described in depth. The connections with Regularization Theory and Statistical Learning Theory are explained.

Math required Lagrange multipliers technique

Web The slides and all what concerns this class can be found on the web.

Plan

- Perceptrons and optimal separating hyperplanes
- Linear SVMs
- Nonlinear SVMs
- Conclusions
- About next class

A new loss function

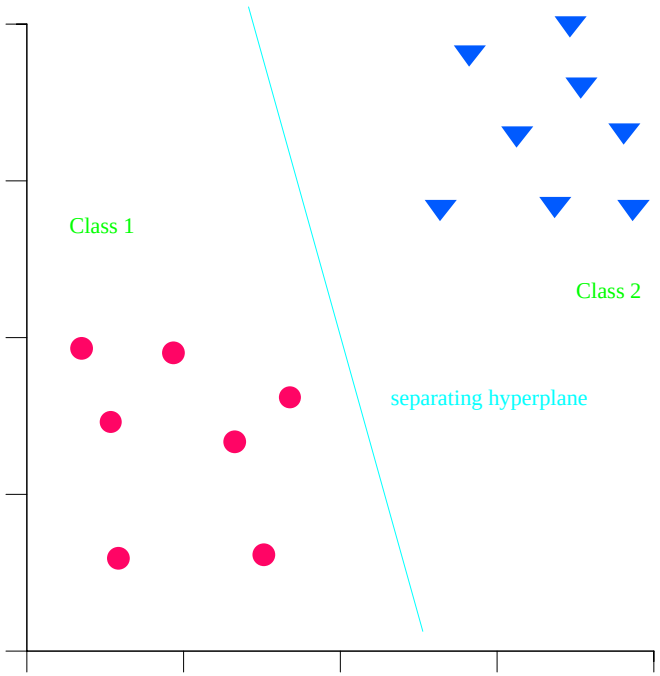
Support Vector Machines are characterized by the loss function

$$V = (1 - yf(\mathbf{x}))_+$$

But let us first explore the connection between SVMs and perceptrons...

Linearly separable classes

A perceptron algorithm separates linearly separable training data with no error.



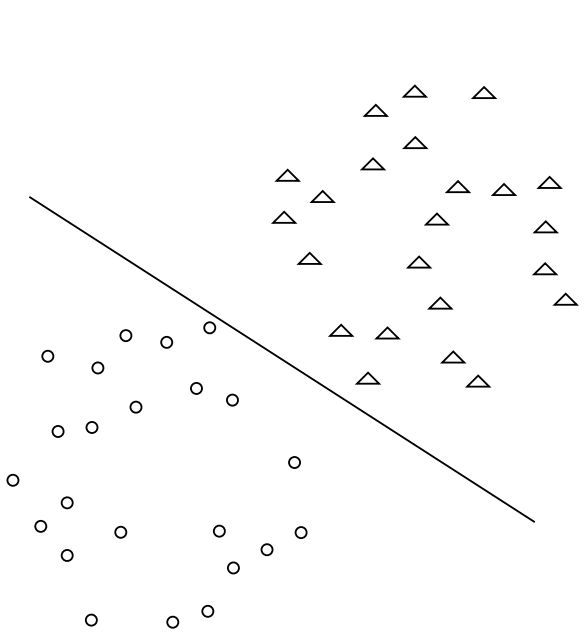
Perceptrons

Why should a perceptron generalize well?

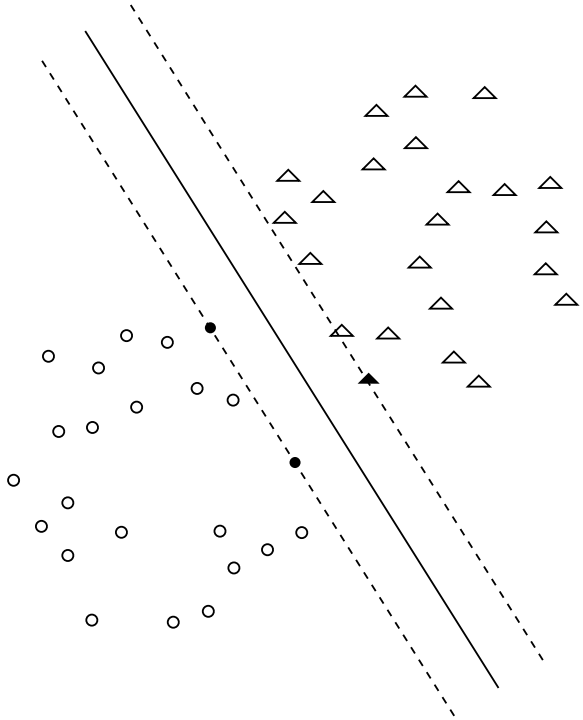
There are many hyperplanes which can separate the data, but there is no criterion to choose.

A criterion for choosing the optimal hyperplane can be embedded in an appropriate cost function ...

Optimal separating hyperplane



(a)



(b)

Linearly separable case

Let S be a set of ℓ points $\boldsymbol{x}_i \in \mathbb{R}^d$. Each $\boldsymbol{x}_i$ is given a label $y_i \in \{-1, 1\}$.

Def 1. S is **linearly separable** if for some $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1, \quad \text{for } i = 1, 2, \ldots, \ell.$$

The hyperplane $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ is a **separating hyperplane**. The signed distance d_i of $\boldsymbol{x}_i$ from the separating hyperplane is

$$d_i = \frac{\boldsymbol{w} \cdot \boldsymbol{x}_i + b}{w}.$$

Optimal separating hyperplane

Def 2. The **optimal separating hyperplane OSH** is the separating hyperplane solution of

Problem **P1**

$$\begin{array}{ll} \text{Minimize} & \frac{1}{2} w \cdot w \\ \text{subject to} & y_i (w \cdot x_i + b) \geq 1, \\ & i = 1, 2, \ldots, \ell. \end{array}$$

The quantity $2/w$ is the **margin**

Saddle point

The solution of Problem **P1** is equivalent to determining the **saddle point** of

$$L = \frac{1}{2} w \cdot w - \sum_{i=1}^{\ell} \alpha_i \left\{ y_i (w \cdot x_i + b) - 1 \right\}.$$

L has a minimum for $w = \bar{w}$ and $b = \bar{b}$ and a maximum for $\alpha = \bar{\alpha}$, and thus

$$\begin{aligned} \frac{\partial L}{\partial b} &= \sum_{i=1}^{\ell} y_i \alpha_i = 0, \\ \frac{\partial L}{\partial w} &= w - \sum_{i=1}^{\ell} \alpha_i y_i x_i = 0 \end{aligned}$$

Dual problem

Problem **P1** reduces to the **dual**

Problem **P2**	
Maximize	$-\frac{1}{2}\boldsymbol{\alpha}^\top D\boldsymbol{\alpha} + \sum \alpha_i$
subject to	$\sum y_i\alpha_i = 0$
	$\boldsymbol{\alpha} \geq 0,$

where D is an $\ell \times \ell$ matrix such that

$$D_{ij} = y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j.$$

Support vectors

As for $(\bar{w}, \bar{b})$, we have

$$\bar{w} = \sum_{i=1}^{\ell} \bar{\alpha}_i y_i x_i,$$

while $\bar{b}$ can be determined from $\bar{\alpha}$, solution of the dual problem, and from the l Kuhn-Tucker conditions

$$\bar{\alpha}_i \Big(y_i(\bar{w} \cdot x_i + \bar{b}) - 1 \Big) = 0.$$

Thus $\bar{w}$ is a linear combination of the x_i for which $\bar{\alpha}_i \neq 0$. These x_i are termed **support vectors**.

An important observation

The fact that that the optimal separating hyperplane is determined only by the support vectors is most remarkable.

Usually, the number of support vectors is very small.

The whole data set could be replaced by these few points and the same hyperplane would be obtained.

All the information contained in the data set is summarized by the support vectors.

Kuhn-Tucker condition

Given a support vector x_j, the parameter $\bar{b}$ can be obtained from the corresponding Kuhn-Tucker condition as

$$\bar{b} = y_j - \bar{\boldsymbol{w}} \cdot \boldsymbol{x}_j.$$

The problem of classifying a new data point x is now simply solved by looking at the sign of

$$\bar{\boldsymbol{w}} \cdot \boldsymbol{x} + \bar{b}.$$

Linearly nonseparable case

If the set S is not linearly separable or one simply ignores whether or not the set S is linearly separable, the previous analysis can be generalized by looking for the solution to

Problem **P3**

Minimize	$\frac{1}{2} w \cdot w + C \sum \xi_i$
subject to	$y_i(w \cdot x_i + b) \geq 1 - \xi_i$
	$i = 1, 2, \ldots, \ell$
	$\xi \geq 0.$

Nonseparable case (cont)

Similarly to the linearly separable case, we have

$$\bar{\boldsymbol{w}} = \sum_{i=1}^{\ell} \bar{\alpha}_i y_i \boldsymbol{x}_i,$$

while $\bar{b}$ can again be determined from $\bar{\boldsymbol{\alpha}}$, solution of the dual problem of **P3**, and from the new Kuhn-Tucker conditions

$$\begin{aligned} \bar{\alpha}_i \left(y_i(\bar{\boldsymbol{w}} \cdot \boldsymbol{x}_i + \bar{b}) - 1 + \bar{\xi}_i \right) &= 0 \\ (C - \bar{\alpha}_i)\bar{\xi}_i &= 0 \end{aligned}$$

New dual problem

Problem **P3** reduces to the **dual**

Problem **P4**

Maximize $-\frac{1}{2}\boldsymbol{\alpha}^\top D\boldsymbol{\alpha} + \sum \alpha_i$

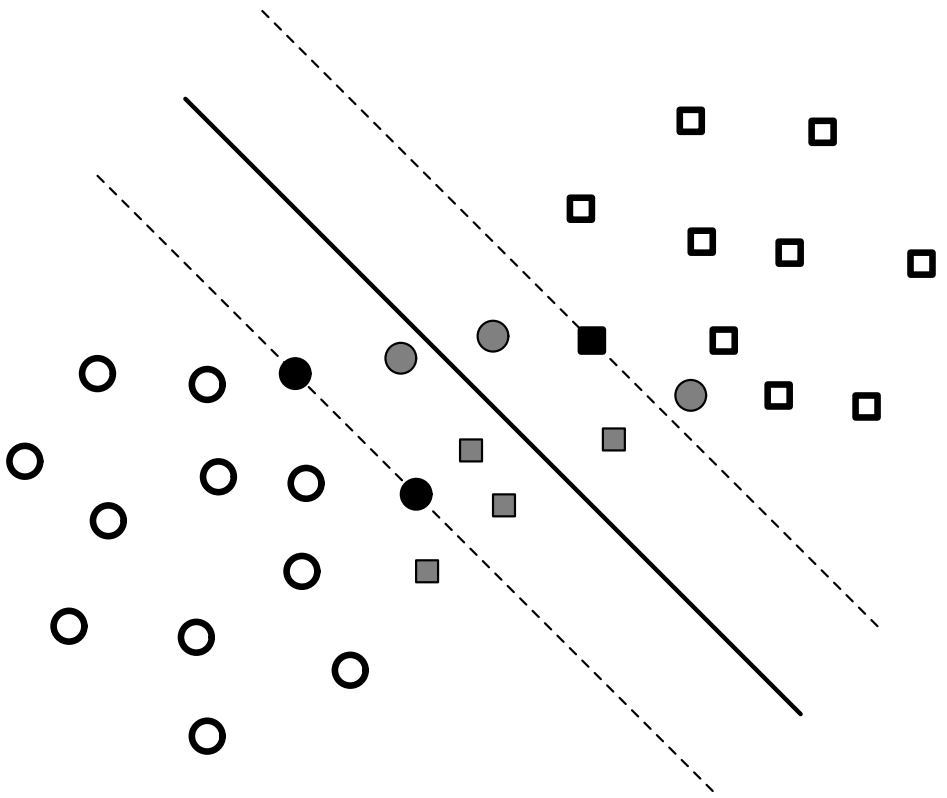
subject to $\sum y_i\alpha_i = 0$

$0 \leq \boldsymbol{\alpha} \leq C,$

Support vectors

The points x_i for which $\bar{\alpha}_i > 0$ are termed **support vectors**. The main difference is that here support vectors do not necessarily lie on the margin.

We call *essential* support vectors the support vectors that appear in all possible expansions of $\mathbf{w}$.



Leave-one-out bound

If E_ℓ is the expected value of the number of misclassifications (the expectation taken over both training and test data, the training set being of size ℓ), we have

$$E_\ell \leq \frac{K_{\ell+1}}{\ell+1}$$

with $K_{\ell+1}$ is the average number of essential support vectors.

Note that this bound is **distribution dependent**.

Who wants a linear classifier?

Linear classifiers in the classical sense are very limited. However, one can consider linear classifiers in *feature space*, rather than in the original space.

Idea: map $\mathbf{x}$ into a vector of features $\phi_i, \ldots, \phi_N$:

$$\mathbf{x} \rightarrow \mathbf{z}(\mathbf{x}) = (a_1\phi_1(\mathbf{x}), a_2\phi_2(\mathbf{x}) \ldots, a_N\phi_N(\mathbf{x}))$$

where ϕ_i are fixed functions, and consider a linear classifier in the variable $\mathbf{z}(\mathbf{x})$ instead of $\mathbf{x}$.

Example

$$\mathbf{x} = (x, y, z)$$
$$\phi_1(\mathbf{x}) = x; \quad \phi_2(\mathbf{x}) = y; \quad \phi_3(\mathbf{x}) = z$$
$$\phi_4(\mathbf{x}) = x^2; \quad \phi_5(\mathbf{x}) = y^2; \quad \phi_6(\mathbf{x}) = z^2$$
$$\phi_7(\mathbf{x}) = xy; \quad \phi_8(\mathbf{x}) = xz; \quad \phi_9(\mathbf{x}) = yz$$

$$\mathbf{z}(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_9(\mathbf{x}))$$

A linear classifier in the feature space ($\mathbf{z}(\mathbf{x})$)

$$f(\mathbf{x}) = \theta(\mathbf{w} \cdot \mathbf{z}(\mathbf{x}) + b)$$

$$\Downarrow$$

a polynomial classifier in the original space ($\mathbf{x}$).

Optimal Separating Hyperplanes in feature space

The following *dual* quadratic programming problem has to be solved:

$$\max_{\boldsymbol{\alpha}} \ (\boldsymbol{\alpha} \cdot 1 - \frac{1}{2}\boldsymbol{\alpha} \cdot D\boldsymbol{\alpha})$$

subject to the constraints:

$$\boldsymbol{\alpha} \cdot \mathbf{y} = 0$$

$$0 \leq \alpha_i \leq C$$

where $D_{ij} = y_i y_j \mathbf{z}(\mathbf{x}_i) \cdot \mathbf{z}(\mathbf{x}_j)$.

Support Vector Machines in input space

The final classifier has the form

$$f(\mathbf{x}) = \theta\left(\sum_{i=1}^{\ell} \alpha_i y_i \mathbf{z}(\mathbf{x}) \cdot \mathbf{z}(\mathbf{x}_i) + b\right)$$

and therefore it is not linear in $\mathbf{x}$ anymore. Notice that:

- We never need to compute $\mathbf{z}(\mathbf{x})$, but only scalar products, like $\mathbf{z}(\mathbf{x}) \cdot \mathbf{z}(\mathbf{x}_i)$ and $\mathbf{z}(\mathbf{x}_i) \cdot \mathbf{z}(\mathbf{x}_j)$

- if the dimensionality of $\mathbf{x}$ is large, the dimensionality of $\mathbf{z}$ could be huge (a 2nd degree polynomial in 265 variables has 35245 terms ...).

A crucial step

Let us define the *kernel function* K as following:

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

Notice that we only need to know $K(\mathbf{x}, \mathbf{y})$ in order to solve the problem.

Admissible kernels

A kernel $K(\mathbf{x}, \mathbf{y})$ is an admissible kernel if it satisfy Mercer's theorem... or

$$\int \int d\mathbf{x} d\mathbf{y} \; K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) > 0 \quad \forall g \in L_2(\mathbb{R}^d)$$

Connection with RT and SLT

SVMs can be seen as the solution to the minimization of the functional

$$\sum_{i=1}^{\ell} V_i + \lambda \|f\|_K^2$$

with

$$V_i = (1 - y_i f(\mathbf{x}_i))_+.$$

- SVMs are an example of Regularization Theory

- SVMs can be thought of as an approximate implementation of the SRM principle discussed in the previous class

Examples of SVMs

When the kernel is $K = K(\mathbf{x}, \mathbf{y})$ the resulting classifier has the form:

$$f(\mathbf{x}) = \theta \left(\sum_{i \in SV} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) \right)$$

where SV is the set of support vectors.

The kernel K could be any of the kernels we met in previous classes...

RBF SVMs

When the kernel is $K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$, with σ given, the resulting classifier has the form:

$$f(\mathbf{x}) = \theta\left(\sum_{i \in SV} y_i \alpha_i e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/2\sigma^2} + b\right)$$

where SV is the set of support vectors. *This is a* **Radial Basis Function** *classifier, where the centers are the support vectors, and the coefficients are the Lagrange multipliers (except for the sign).*

Polynomial SVMs

When the kernel is $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^n$, for a given n, the resulting classifier has the form:

$$f(\mathbf{x}) = \theta \left(\sum_{i \in SV} y_i \alpha_i (\mathbf{x} \cdot \mathbf{x}_i + 1)^n + b \right)$$

This is a (very sparse) **polynomial** *classifier.*

Example: Polynomial of Degree 2 in 2 Variables

$$\mathbf{x} = (x_1, x_2) \ , \quad \mathbf{y} = (y_1, y_2)$$

$$\mathbf{z}(\mathbf{x}) = (1, \sqrt{2}\ x_1, \sqrt{2}\ x_2, x_1^2, x_2^2, \sqrt{2}\ x_1 x_2)$$

$$\mathbf{z}(\mathbf{y}) = (1, \sqrt{2}\ y_1, \sqrt{2}\ y_2, y_1^2, y_2^2, \sqrt{2}\ y_1 y_2)$$

$$\Downarrow$$

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{z}(\mathbf{x}) \cdot \mathbf{z}(\mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^2$$

Training an SVM

Training an SVM requires the choice of few relevant quantities:

- the kernel function, that determines the shape of the decision surface;
- a "smoothing" parameter in the kernel function (variance of the Gaussian, degree of the polynomial, ...);
- the constant C for the cost of the misclassification errors;

Conclusions

- Training an SVM is equivalent to solve a QP problem with linear constraints and as many variables as data points;

- Generalization performances can be bounded after training, counting the number of support vectors;

- SVMs provide, as a side-effect, a compact description of the data;

About next class

Theme Project discussion...