MATH 567: Mathematical Techniques in Data Science
Support vector machines and kernels

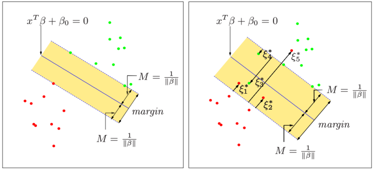
Dominique Guillot

Departments of Mathematical Sciences
University of Delaware

March 20, 2017

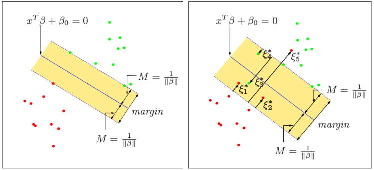
Separating sets: mapping the features

We saw in the previous lecture how support vector machines provide a robust way of finding a separating hyperplane:

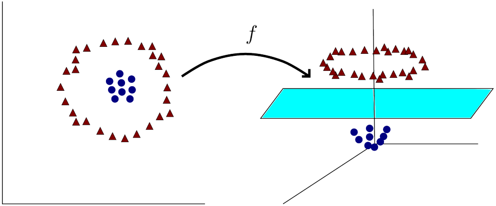


Separating sets: mapping the features

We saw in the previous lecture how support vector machines provide a robust way of finding a separating hyperplane:



What if the data is not separable? Can map into a high-dimensional space.



$$f : \mathbb{R}^2 \to \mathbb{R}^3$$

A brief intro to duality in optimization

Consider the problem:

$$\begin{aligned} \min_{x \in \mathcal{D} \subset \mathbb{R}^n} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \qquad i = 1, \ldots, m \\ & h_i(x) = 0, \qquad i = 1, \ldots, p. \end{aligned}$$

Denote by $p^\star$ the optimal value of the problem.

Consider the problem:

$$\begin{aligned} \min_{x \in \mathcal{D} \subset \mathbb{R}^n} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \qquad i = 1, \ldots, m \\ & h_i(x) = 0, \qquad i = 1, \ldots, p. \end{aligned}$$

Denote by $p^\star$ the optimal value of the problem.

Lagrangian: $L : \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$

$$L(x, \lambda, \nu) := f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x).$$

A brief intro to duality in optimization

Consider the problem:

$$\begin{aligned} \min_{x \in \mathcal{D} \subset \mathbb{R}^n} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \qquad i = 1, \ldots, m \\ & h_i(x) = 0, \qquad i = 1, \ldots, p. \end{aligned}$$

Denote by $p^\star$ the optimal value of the problem.

Lagrangian: $L : \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$

$$L(x, \lambda, \nu) := f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x).$$

Lagrange dual function: $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$

$$g(\lambda, \nu) := \inf_{x \in \mathcal{D}} L(x, \lambda, \nu).$$

A brief intro to duality in optimization

Consider the problem:

$$\begin{aligned} \min_{x \in \mathcal{D} \subset \mathbb{R}^n} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \le 0, \qquad i = 1, \ldots, m \\ & h_i(x) = 0, \qquad i = 1, \ldots, p. \end{aligned}$$

Denote by $p^\star$ the optimal value of the problem.

Lagrangian: $L : \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$

$$L(x, \lambda, \nu) := f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x).$$

Lagrange dual function: $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$

$$g(\lambda, \nu) := \inf_{x \in \mathcal{D}} L(x, \lambda, \nu).$$

Claim: for every $\lambda \ge 0$,

$$g(\lambda, \nu) \le p^\star.$$

Dual problem:

$$\max_{\lambda \in \mathbb{R}^m,\ \nu \in \mathbb{R}^p} g(\lambda, \nu)$$
$$\text{subject to } \lambda \geq 0.$$

Dual problem:

$$\max_{\lambda \in \mathbb{R}^m,\ \nu \in \mathbb{R}^p} g(\lambda, \nu)$$
$$\text{subject to } \lambda \geq 0.$$

Denote by $d^\star$ the optimal value of the dual problem. Clearly

$$d^\star \leq p^\star \qquad \text{(weak duality)}.$$

A brief intro to duality in optimization

Dual problem:

$$\max_{\lambda \in \mathbb{R}^m,\ \nu \in \mathbb{R}^p} g(\lambda, \nu)$$
$$\text{subject to } \lambda \geq 0.$$

Denote by $d^\star$ the optimal value of the dual problem. Clearly

$$d^\star \leq p^\star \qquad \text{(weak duality)}.$$

Strong duality: $d^\star = p^\star$.

- Does not hold in general.
- Usually holds for convex problems.
- (See e.g. Slater's constraint qualification).

The kernel trick

Recall that SVM solves:

$$\min_{\beta_0,\beta,\xi} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i$$

$$\xi_i \geq 0.$$

The kernel trick

Recall that SVM solves:

$$\min_{\beta_0,\beta,\xi} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i$$
$$\xi_i \geq 0.$$

The associated Lagrangian is

$$L_P = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(x_i^T\beta + \beta_0) - (1-\xi_i)] - \sum_{i=1}^{n}\mu_i\xi_i,$$

which we minimize w.r.t. β, β_0, ξ.

Recall that SVM solves:

$$\min_{\beta_0,\beta,\xi} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^n \xi_i$$
$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i$$
$$\xi_i \geq 0.$$

The associated Lagrangian is

$$L_P = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i[y_i(x_i^T\beta + \beta_0) - (1-\xi_i)] - \sum_{i=1}^n \mu_i\xi_i,$$

which we minimize w.r.t. β, β_0, ξ. Setting the respective derivatives to 0, we obtain:

$$\beta = \sum_{i=1}^n \alpha_i y_i x_i, \quad 0 = \sum_{i=1}^n \alpha_i y_i, \quad \alpha_i = C - \mu_i \quad (i = 1, \ldots, n).$$

The kernel trick (cont.)

Substituting into L_P, we obtain the Lagrange (dual) objective function:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j.$$

The kernel trick (cont.)

Substituting into L_P, we obtain the Lagrange (dual) objective function:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j.$$

The function L_D provides a lower bound on the original objective function at any feasible point (weak duality).

The kernel trick (cont.)

Substituting into L_P, we obtain the Lagrange (dual) objective function:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j.$$

The function L_D provides a lower bound on the original objective function at any feasible point (weak duality).

The solution of the original SVM problem can be obtained by maximizing L_D under the previous constraints (strong duality).

The kernel trick (cont.)

Substituting into L_P, we obtain the Lagrange (dual) objective function:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j.$$

The function L_D provides a lower bound on the original objective function at any feasible point (weak duality).

The solution of the original SVM problem can be obtained by maximizing L_D under the previous constraints (strong duality).

Now suppose $h : \mathbb{R}^p \to \mathbb{R}^m$, transforming our features to

$$h(x_i) = (h_1(x_i), \ldots, h_m(x_i)) \in \mathbb{R}^m.$$

The kernel trick (cont.)

Substituting into L_P, we obtain the Lagrange (dual) objective function:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j.$$

The function L_D provides a lower bound on the original objective function at any feasible point (weak duality).

The solution of the original SVM problem can be obtained by maximizing L_D under the previous constraints (strong duality).

Now suppose $h : \mathbb{R}^p \to \mathbb{R}^m$, transforming our features to

$$h(x_i) = (h_1(x_i), \ldots, h_m(x_i)) \in \mathbb{R}^m.$$

The Lagrange dual function becomes:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{h}(\mathbf{x_i})^{\mathbf{T}} \mathbf{h}(\mathbf{x_j}).$$

Substituting into L_P, we obtain the Lagrange (dual) objective function:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j.$$

The function L_D provides a lower bound on the original objective function at any feasible point (weak duality).

The solution of the original SVM problem can be obtained by maximizing L_D under the previous constraints (strong duality).

Now suppose $h : \mathbb{R}^p \to \mathbb{R}^m$, transforming our features to

$$h(x_i) = (h_1(x_i), \ldots, h_m(x_i)) \in \mathbb{R}^m.$$

The Lagrange dual function becomes:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{h(x_i)^T h(x_j)}.$$

Important observation: L_D only depends on $\langle h(x_i), h(x_j) \rangle$.

Positive definite kernels

Important observation: L_D only depends on $\langle h(x_i), h(x_j) \rangle$.

Positive definite kernels

Important observation: L_D only depends on $\langle h(x_i), h(x_j)\rangle$.

In fact, we don't even need to specify h, we only need:

$$K(x, x') = \langle h(x), h(x')\rangle.$$

Positive definite kernels

Important observation: L_D only depends on $\langle h(x_i), h(x_j)\rangle$.

In fact, we don't even need to specify h, we only need:

$$K(x, x') = \langle h(x), h(x')\rangle.$$

Question: Given $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, when can we guarantee that

$$K(x, x') = \langle h(x), h(x')\rangle$$

for some function h?

Positive definite kernels

Important observation: L_D only depends on $\langle h(x_i), h(x_j)\rangle$.

In fact, we don't even need to specify h, we only need:

$$K(x, x') = \langle h(x), h(x')\rangle.$$

Question: Given $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, when can we guarantee that

$$K(x, x') = \langle h(x), h(x')\rangle$$

for some function h?

The previous question can be answered using the notion of *positive definite function* in functional analysis.

Positive definite kernels

Important observation: L_D only depends on $\langle h(x_i), h(x_j)\rangle$.

In fact, we don't even need to specify h, we only need:

$$K(x, x') = \langle h(x), h(x')\rangle.$$

Question: Given $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, when can we guarantee that

$$K(x, x') = \langle h(x), h(x')\rangle$$

for some function h?

The previous question can be answered using the notion of *positive definite function* in functional analysis.

Observation: Suppose K has the desired form. Then, for $x_1, \dots, x_N \in \mathbb{R}^p$, and $v_i := h(x_i)$,

$$\begin{aligned}(K(x_i, x_j)) &= (\langle h(x_i), h(x_j)\rangle) \\ &= (\langle v_i, v_j\rangle) \\ &= V^T V, \qquad \text{where } V = (v_1^T, \dots, v_N^T).\end{aligned}$$

Positive definite kernels

Important observation: L_D only depends on $\langle h(x_i), h(x_j)\rangle$.

In fact, we don't even need to specify h, we only need:

$$K(x, x') = \langle h(x), h(x')\rangle.$$

Question: Given $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, when can we guarantee that

$$K(x, x') = \langle h(x), h(x')\rangle$$

for some function h?

The previous question can be answered using the notion of *positive definite function* in functional analysis.

Observation: Suppose K has the desired form. Then, for $x_1, \dots, x_N \in \mathbb{R}^p$, and $v_i := h(x_i)$,

$$\begin{aligned}(K(x_i, x_j)) &= (\langle h(x_i), h(x_j)\rangle) \\ &= (\langle v_i, v_j\rangle) \\ &= V^T V, \qquad \text{where } V = (v_1^T, \dots, v_N^T).\end{aligned}$$

Conclusion: the matrix $(K(x_i, x_j))$ is positive semidefinite.

Positive definite kernels (cont.)

- **Necessary condition to have $K(x, x') = \langle h(x), h(x') \rangle$:**

$$(K(x_i, x_j))_{i,j=1}^N \text{ is psd}$$

for any $x_1, \ldots, x_N$, and any $N \geq 1$.

Positive definite kernels (cont.)

- **Necessary condition to have $K(x, x') = \langle h(x), h(x') \rangle$:**

$$(K(x_i, x_j))_{i,j=1}^N \text{ is psd}$$

for any $x_1, \ldots, x_N$, and any $N \geq 1$.
- Note also that $K(x, x') = K(x', x)$ if $K(x, x') = \langle h(x), h(x') \rangle$.

Positive definite kernels (cont.)

- **Necessary condition to have $K(x, x') = \langle h(x), h(x')\rangle$:**

$$(K(x_i, x_j))_{i,j=1}^N \text{ is psd}$$

for any $x_1, \ldots, x_N$, and any $N \geq 1$.
- Note also that $K(x, x') = K(x', x)$ if $K(x, x') = \langle h(x), h(x')\rangle$.

Definition: Let $\mathcal{X}$ be a set. A symmetric kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be a *positive (semi)definite kernel* if

$$(K(x_i, x_j))_{i,j=1}^N \text{ is positive (semi)definite}$$

for all $x_1, \ldots, x_N \in \mathcal{X}$ and all $N \geq 1$.

Positive definite kernels (cont.)

- **Necessary condition to have $K(x, x') = \langle h(x), h(x') \rangle$:**

$$(K(x_i, x_j))_{i,j=1}^{N} \text{ is psd}$$

for any $x_1, \ldots, x_N$, and any $N \geq 1$.

- Note also that $K(x, x') = K(x', x)$ if $K(x, x') = \langle h(x), h(x') \rangle$.

Definition: Let $\mathcal{X}$ be a set. A symmetric kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be a *positive (semi)definite kernel* if

$$(K(x_i, x_j))_{i,j=1}^{N} \text{ is positive (semi)definite}$$

for all $x_1, \ldots, x_N \in \mathcal{X}$ and all $N \geq 1$.

- One can show that positive definite kernels can be written $K(x, x') = \langle h(x), h(x') \rangle$ for some function h defined on an appropriate space.

Back to SVM

We can replace h by any positive definite kernel in the SVM problem:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{h}(\mathbf{x_i})^{\mathbf{T}} \mathbf{h}(\mathbf{x_j})$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{K}(\mathbf{x_i}, \mathbf{x_j}).$$

Back to SVM

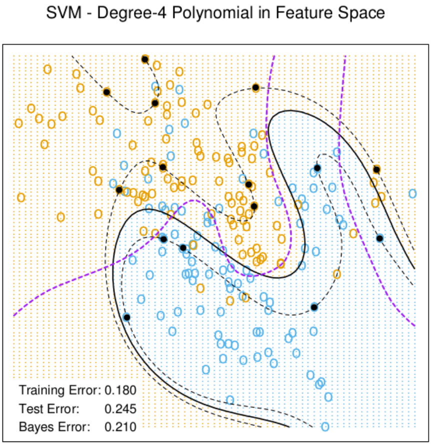
We can replace h by any positive definite kernel in the SVM problem:

$$\begin{aligned} L_D &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{h}(\mathbf{x_i})^{\mathbf{T}} \mathbf{h}(\mathbf{x_j}) \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{K}(\mathbf{x_i}, \mathbf{x_j}). \end{aligned}$$

Three popular choice in the SVM literature:

$$K(x, x') = e^{-\gamma \|x - x'\|_2^2} \qquad \text{(Gaussian kernel)}$$
$$K(x, x') = (1 + \langle x, x' \rangle)^d \qquad (d\text{-th degree polynomial})$$
$$K(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2) \qquad \text{(Neural networks)}.$$

Example: decision function



ESL, Figure 12.3 (solid black line = decision boundary, dotted line = margin).