IAI, TCG CREST

Avisek Gupta (avisek003@gmail.com)

Machine Learning

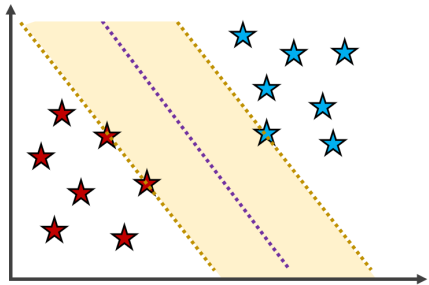
19 – Support Vector Machines (2)

November 22, 2022

Recap: Support Vector Machines

For a binary classification problem, we have samples $(\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(n)}, y^{(n)}), \mathbf{x}^{(i)} \in \mathbb{R}^d, y^{(i)} \in \{1, -1\}$.

We wish to find a hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ that has maximum margin, and correctly classifies the data.



Recap: Support Vector Machines

The objective function for Support Vector Machines:

$$\min_{\mathbf{w},b} \ \mathbf{w}^T\mathbf{w}$$
$$\text{s.t., } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 \ \forall i.$$

Quadratic Optimization Problem (QOP): An optimization problem with a quadratic objective and with linear equality or linear inequality constraints. Quadratic solvers can solve QOPs (but they are not very efficient).

Overview: Unconstrained Optimization

- Objective: $\min_x f(x)$, or $\max_x -f(x)$
- Unconstrained optimization methods aim to obtain an optimal value p^* by producing a sequence of points $x^{(k)}$, k=0,1,..., so that,

$$f(x^{(k)}) \to p^*$$

- Unconstrained optimization methods can also be viewed as iterative methods that solve for the optimality condition

$$\nabla f(x^*) = 0$$

- Methods:
  - Gradient Descent $(x^{(k+1)} = x^{(k)} - \eta \nabla f(x^{(k)})$
  - Newton's Descent $(x^{(k+1)} = x^{(k)} - \eta \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$
  - Nelder-Mead method
  - ...

Overview: Constrained Optimization

Optimization problem written in terms of an objective function f, inequality constraints involving g_i, and equality constraints involving h_j.

$$\begin{aligned} &\min_x f(x) \\ \text{s.t., } &g_i(x) \leq 0 \ \forall i \\ &h_j(x) = 0 \forall j \end{aligned}$$

Whether an optimal solution can be found depends on the nature of the functions f, g_i, h_j.

Linear Constrained Optimization

$$\begin{aligned} &\min_x f(x) \\ \text{s.t., } &g_i(x) \leq 0 \ \forall i \\ &h_j(x) = 0 \ \forall j \end{aligned}$$

Linear Programming - If f is linear, and all g_i, h_j are linear, then an optimal solution can be found.

Methods:

- Simplex
- ...
- Ellipsoid Algorithm
- Karmarkar Algorithm

Convex Constrained Optimization

$$\begin{aligned} &\min_x f(x) \\ \text{s.t., } &g_i(x) \leq 0 \ \forall i \\ &h_j(x) = 0 \ \forall j \end{aligned}$$

Convex Programming - f, g_i are convex, h_j are affine.

If f is quadratic, then quadratic solvers (inefficient) exist to obtain a solution.

Can we get an optimal solution? - Check the *KKT conditions*.

Can we get a better optimization method? - Depending on the problem, we may be able to (e.g.: PCA).

Optimizing a generalized Lagrangian

$$\begin{aligned} &\min_{w} f(w) \\ \text{s.t., } &g_i(w) \leq 0 \ , \ i = 1, ..., n \\ &h_j(w) = 0 \ , \ j = 1, ..., m \end{aligned}$$

The **generalized Lagrangian** can be formed as,

$$L(w, \alpha, \beta) = f(w) + \sum_{i=1}^{n} \alpha_i g_i(w) + \sum_{j=1}^{m} \beta_j h_j(w)$$

Optimizing from the Lagrangian

If the optimization problem has a convex objective with only equality constraints,

$$\min_{w} f(w)$$
$$\text{s.t., } h_j(w) = 0 \ , \ j = 1, ..., m$$

The Lagrangian can be formed as,

$$L(w, \beta) = f(w) + \sum_{i=1}^{n} \sum_{j=1}^{m} \beta_j h_j(w)$$

The optimal solution can be found by solving the derivatives of the Lagrangian set equal to zero.

$$\nabla_w L(w, \beta) = 0, \ \nabla_\beta L(w, \beta) = 0.$$

Optimizing the generalized Lagrangian

$$\begin{aligned} &\min_{w} f(w) \\ \text{s.t.,}\ &g_i(w) \leq 0\ ,\ i = 1, ..., n \\ &h_j(w) = 0\ ,\ j = 1, ..., m \end{aligned}$$

The **generalized Lagrangian** can be formed as,

$$L(w, \alpha, \beta) = f(w) + \sum_{i=1}^{n} \alpha_i g_i(w) + \sum_{j=1}^{m} \beta_j h_j(w)$$

Consider the following two problems:

The Primal Problem

$$\begin{aligned} &\min_{w} f(w) \\ \text{s.t., } &g_i(w) \leq 0 \ , \ i = 1, ..., n \\ &h_j(w) = 0 \ , \ j = 1, ..., m \end{aligned}$$

The **generalized Lagrangian** can be formed as,

$$L(w, \alpha, \beta) = f(w) + \sum_{i=1}^{n} \alpha_i g_i(w) + \sum_{j=1}^{m} \beta_j h_j(w)$$

1. We define the quantity:

$$\theta_P(w) = \max_{\alpha, \beta, \alpha_i \geq 0} L(w, \alpha, \beta)$$

Note that

$$\theta_P(w) = \begin{cases} f(w) & \text{if } w \text{ satisfies constraints} \\ \infty & \text{otherwise} \end{cases}$$

The Primal Problem

$$\begin{aligned} &\min_{w} f(w) \\ \text{s.t., } &g_i(w) \leq 0 \ , \ i = 1, ..., n \\ &h_j(w) = 0 \ , \ j = 1, ..., m \end{aligned}$$

The **generalized Lagrangian** can be formed as,

$$L(w, \alpha, \beta) = f(w) + \sum_{i=1}^{n} \alpha_i g_i(w) + \sum_{j=1}^{m} \beta_j h_j(w)$$

1. We define the quantity:

$$\theta_P(w) = \max_{\alpha, \beta, \alpha_i \geq 0} L(w, \alpha, \beta)$$

We construct the following primal problem:

$$\min_{w} \theta_P(w) = \min_{w} \max_{\alpha, \beta, \alpha_i \geq 0} L(w, \alpha, \beta)$$

The Primal and the Dual

The **generalized Lagrangian** can be formed as,

$$L(w, \alpha, \beta) = f(w) + \sum_{i=1}^{n} \alpha_i g_i(w) + \sum_{j=1}^{m} \beta_j h_j(w)$$

1. We construct the following primal problem:

$$\min_w \theta_P(w) = \min_w \max_{\alpha,\beta,\alpha_i \geq 0} L(w, \alpha, \beta)$$
$$p^* = \min_w \theta_P(w)$$

2. We also construct the following dual problem:

$$\max_{\alpha,\beta,\alpha_i \geq 0} \theta_D(w) = \max_{\alpha,\beta,\alpha_i \geq 0} \min_w L(w, \alpha, \beta)$$
$$d^* = \max_{\alpha,\beta,\alpha_i \geq 0} \theta_D(w)$$

The Primal and the Dual

1. We construct the following primal problem:

$$\min_{w} \theta_P(w) = \min_{w} \max_{\alpha,\beta,\alpha_i \geq 0} L(w, \alpha, \beta)$$
$$p^* = \min_{w} \theta_P(w)$$

2. We also construct the following dual problem:

$$\max_{\alpha,\beta,\alpha_i \geq 0} \theta_D(w) = \max_{\alpha,\beta,\alpha_i \geq 0} \min_{w} L(w, \alpha, \beta)$$
$$d^* = \max_{\alpha,\beta,\alpha_i \geq 0} \theta_D(w)$$

How are the two problems related?

$$d^* = \max_{\alpha,\beta,\alpha_i \geq 0} \min_{w} L(w, \alpha, \beta) \leq \min_{w} \max_{\alpha,\beta,\alpha_i \geq 0} L(w, \alpha, \beta) = p^*$$

KKT Conditions

How are the two problems related?

$$d^* = \max_{\alpha,\beta,\alpha_i \geq 0} \min_{w} L(w, \alpha, \beta) \leq \min_{w} \max_{\alpha,\beta,\alpha_i \geq 0} L(w, \alpha, \beta) = p^*$$

KKT Conditions: Let f and g_i be convex, and h_j are affine. Also, let there exist some w so that $g_i(w) < 0 \ \forall i$. Then there must exist w^*, α^*, β^*, so that w^* is the solution to the primal problem, α^*, β^* are solutions to the dual problem, and $p^* = d^* = L(w^*, \alpha^*, \beta^*)$. The following KKT conditions must be satisfied by w^*, α^*, β^*:

$$\begin{aligned}
\frac{\partial}{\partial w_i} L(w^*, \alpha^*, \beta^*) &= 0, \ \ i = 1, ..., d \\
\frac{\partial}{\partial \beta_i} L(w^*, \alpha^*, \beta^*) &= 0, \ \ i = 1, ..., m \\
\alpha_i^* g_i(w^*) &= 0, \ \ i = 1, ..., n \\
g_i(w^*) &\leq 0, \ \ i = 1, ..., n \\
\alpha_i^* &\geq 0, \ \ i = 1, ..., n
\end{aligned}$$

KKT Conditions

$$d^* = \max_{\alpha,\beta,\alpha_i \geq 0} \min_{w} L(w, \alpha, \beta) \leq \min_{w} \max_{\alpha,\beta,\alpha_i \geq 0} L(w, \alpha, \beta) = p^*$$

KKT Conditions:

$$\begin{aligned}
\frac{\partial}{\partial w_i} L(w^*, \alpha^*, \beta^*) &= 0, \ i = 1, ..., d \\
\frac{\partial}{\partial \beta_i} L(w^*, \alpha^*, \beta^*) &= 0, \ i = 1, ..., m \\
\alpha_i^* g_i(w^*) &= 0, \ i = 1, ..., n \\
g_i(w^*) &\leq 0, \ i = 1, ..., n \\
\alpha_i^* &\geq 0, \ i = 1, ..., n
\end{aligned}$$

The constraint $\alpha_i^* g_i(w^*) = 0$ implies that if $\alpha_i^* > 0$ then $g_i(w^*) = 0$.

SVM-Primal

Support Vector Machines Objective:

$$\min_{\mathbf{w},b} \ \frac{1}{2}\mathbf{w}^T\mathbf{w}$$
$$\text{s.t., } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 \ \forall i.$$

The constraints of the primal can be written as, $-y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) + 1 \leq 0$.

An SVM primal objective is formed,

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i[y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) - 1]$$

The primal problem can be converted to a dual problem, which provides a different way to solve the problem.

SVM-Primal

The primal:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i [y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) - 1]$$

Equating the derivatives of the primal wrt $\mathbf{w}, b$ to zero,

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}^{(i)} = 0 \implies \mathbf{w} = \sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}^{(i)}$$

$$\frac{\partial}{\partial b} L(\mathbf{w}, b) = 0 - \sum_{i=1}^{n} \alpha_i y^{(i)} \implies \sum_{i=1}^{n} \alpha_i y^{(i)} = 0.$$

SVM-Primal

The primal: $L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i[y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) - 1]$

From the derivatives of the primal: $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y^{(i)}\mathbf{x}^{(i)}, \sum_{i=1}^{n} \alpha_i y^{(i)} = 0.$

Substituting the expression for $\mathbf{w}$ in the primal, we get,

$$\begin{aligned}
L(\alpha) =& \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i y^{(i)}\mathbf{w}^T x^{(i)} - \sum_{i=1}^{n} \alpha_i y^{(i)} b + \sum_{i=1}^{n} \alpha_i \\
=& \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y^{(i)}y^{(j)}\mathbf{x}^{(i)T}\mathbf{x}^{(j)} - \sum_{i,j=1}^{n} \alpha_i\alpha_j y^{(i)}y^{(j)}\mathbf{x}^{(i)T}\mathbf{x}^{(j)} \\
& - \sum_{i,j=1}^{n} \alpha_i y_{(i)} b + \sum_{i,j=1}^{n} \alpha_i \\
=& \sum_{i,j=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y^{(i)}y^{(j)}\mathbf{x}^{(i)T}\mathbf{x}^{(j)}
\end{aligned}$$

SVM-Dual

$$\max_{\alpha} \sum_{i,j=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} \mathbf{x}^{(i)T} \mathbf{x}^{(j)}$$

$$\text{s.t., } \alpha_i \geq 0 \ \forall i \quad , \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0.$$

Recovering $\mathbf{w}$ from optimal α^*:

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha^* y^{(i)} \mathbf{x}^{(i)}$$

Recovering b from optimal α^*:

$$b_A = \max_{i, y_i = -1} \mathbf{w}^{*T} \mathbf{x}^{(i)}$$

$$b_B = \min_{i, y_i = 1} \mathbf{w}^{*T} \mathbf{x}^{(i)}$$

Then, $b^* = \frac{b_A + b_B}{2}$.

KKT Conditions

$$d^* = \max_{\alpha,\beta,\alpha_i \geq 0} \min_{w} L(w, \alpha, \beta) \leq \min_{w} \max_{\alpha,\beta,\alpha_i \geq 0} L(w, \alpha, \beta) = p^*$$

KKT Conditions:

$$1.\ \frac{\partial}{\partial w_i} L(w^*, \alpha^*, \beta^*) = 0,\ i = 1, ..., d$$

$$2.\ \frac{\partial}{\partial \beta_i} L(w^*, \alpha^*, \beta^*) = 0,\ i = 1, ..., m$$

$$3.\ \alpha_i^* g_i(w^*) = 0,\ i = 1, ..., n$$

$$4.\ g_i(w^*) \leq 0,\ i = 1, ..., n$$

$$5.\ \alpha_i^* \geq 0,\ i = 1, ..., n$$

The constraint $\alpha_i^* g_i(w^*) = 0\ \forall i$ implies that if $\alpha_i^* > 0$ then $g_i(w^*) = 0$.

SVM: KKT Conditions

The primal:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i[y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) - 1]$$

KKT Conditions:

$$1.\ \frac{\partial}{\partial w_i} L(w^*, b^*, \alpha^*) = 0,\ i = 1, ..., d$$

$$\frac{\partial}{\partial b} L(w^*, b^*, \alpha^*) = 0$$

$$3.\ \alpha_i^*[y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) - 1] = 0,\ i = 1, ..., n$$

$$4.\ y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) - 1 \leq 0,\ i = 1, ..., n$$

$$5.\ \alpha_i^* \geq 0,\ i = 1, ..., n$$

The constraint $\alpha_i^*[y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) - 1] = 0\ \forall i$ implies that if $\alpha_i^* > 0$ then $y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) = 1$.

SVM: Support Vectors

The constraint $\alpha_i^*[y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) - 1] = 0 \; \forall i$ implies that if $\alpha_i^* > 0$ then $y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) = 1$.

Support Vectors: For those data instances $\mathbf{x}^{(i)}$ for which $y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) = 1$, $\alpha_i^* > 0$.

For all other data instances for which $y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) > 1$, $\alpha_i^* = 0$.

Thus the SVM max-margin hyperplane is defined only in terms of the support vectors. Recall:

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y^{(i)} \mathbf{x}^{(i)}$$

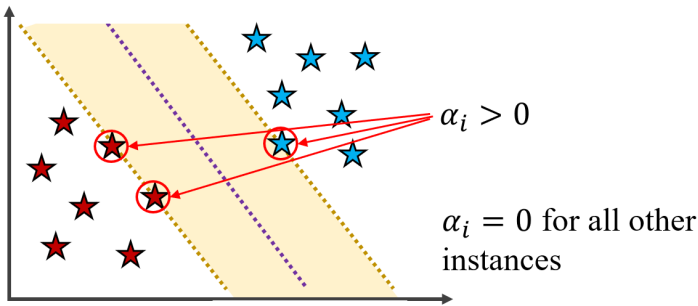
$$b^* = \frac{\max_{i,y_i=-1} \mathbf{w}^{*T}\mathbf{x}^{(i)} + \min_{i,y_i=1} \mathbf{w}^{*T}\mathbf{x}^{(i)}}{2}$$

SVM: Support Vectors

The constraint $\alpha_i^*[y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) - 1] = 0 \; \forall i$ implies that if $\alpha_i^* > 0$ then $y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) = 1$.

Support Vectors: For those data instances $\mathbf{x}^{(i)}$ for which $y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) = 1$, $\alpha_i^* > 0$.

For all other data instances for which $y^{(i)}(\mathbf{w}^{*T}\mathbf{x}^{(i)} + b) > 1$, $\alpha_i^* = 0$.



Testing SVM

Any new test data $\mathbf{x}'$ can be assigned to a class by fitting it on the hyperplane:

$$\begin{aligned}\mathbf{w}^{*T}\mathbf{x}' + b &= \left(\sum_{i=1}^{n} \alpha_i^* y^{(i)} \mathbf{x}^{(i)}\right)^T \mathbf{x}' + b \\ &= \sum_{i=1}^{n} \alpha_i^* y^{(i)} < \mathbf{x}^{(i)}, \mathbf{x}' > + b\end{aligned}$$

Since $\alpha_i^* = 0$ for all non-support vectors, testing is done only using the support vectors.

SVM for non-linearly separable data

The SVM dual objective in terms of the dot product between $\mathbf{x}^{(i)}, \mathbf{x}^{(j)}, \ \forall i, j$:

$$\max_{\alpha} \sum_{i,j=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} < \mathbf{x}^{(i)}, \mathbf{x}^{(j)} >$$

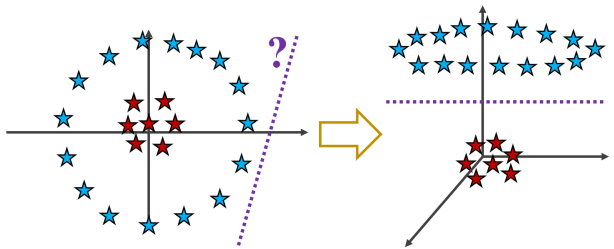
$$\text{s.t., } \alpha_i \geq 0 \ \forall i \ , \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0.$$

Efficient algorithms exist to identify the support vectors that lead to the solution of the objective (SMO algorithm).

How can the objective be extended for data that is not linearly separable?

Kernels

Even if the data is not linearly separable, it may be possible to transform the data by projecting it to a higher dimensional space where the data becomes linearly separable.



For a data set $[\mathbf{x}_1, ..., \mathbf{x}_n], \mathbf{x}_i \in \mathbb{R}^d$, let $\phi(\mathbf{x}) \in \mathbb{R}^h, h > d$ be a mapping of the data instances to a higher dimensional space.
For a specific mapping ϕ, the inner product in the higher dimensional space can be defined in terms of a Kernel function,

$$K(\mathbf{x}_i, \mathbf{x}_j) = < \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) >$$

Kernels functions

For a specific mapping ϕ, the inner product in the higher dimensional space can be defined in terms of a kernel function,

$$K(\mathbf{x}_i, \mathbf{x}_j) = < \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) >$$

Instead of explicitly mapping each $\mathbf{x}_i$ to $\phi(\mathbf{x}_i)$ and then calculating the inner product, instead the kernel function can be directly computed on every pair $\mathbf{x}_i, \mathbf{x}_j \ \forall i, j$.

Examples of kernel functions:

- Polynomial kernel: $K_{a,b}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + a)^b$
- Gaussian kernel: $K_\sigma(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2})^b$

Kernel similarity matrices

A kernel similarity matrix can be defined: $K_{ij} = K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$

Kernel matrices are symmetric: $K_{ij} = K_{ji}\ \forall i, j$

Kernel matrices are also positive semidefinite:

$$
\begin{aligned}
\forall z, z^T K z &= \sum_i \sum_j z_i K_{ij} z_j \\
&= \sum_i \sum_j z_i \phi(\mathbf{x}^{(i)})^T \phi(\mathbf{x}^{(j)}) z_j \\
&= \sum_i \sum_j z_i \sum_k \phi_k(\mathbf{x}^{(i)}) \phi_k(\mathbf{x}^{(j)}) z_j \\
&= \sum_k \sum_i \sum_j z_i \phi_k(\mathbf{x}^{(i)}) \phi_k(\mathbf{x}^{(j)}) z_j \\
&= \sum_k (\sum_i z_i \phi_k(\mathbf{x}^{(i)}))^2 \geq 0.
\end{aligned}
$$

Kernel similarity matrices

A kernel function K is a valid kernel if it corresponds to some feature mapping ϕ.

Mercer's Theorem: Let $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ be given. For K to be a valid kernel, it is necessary and sufficient that for any $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(n)}\}, n < \infty$, the corresponding kernel matrix is symmetric and positive semi-definite.

Kernel similarity matrices

A kernel function K is a valid kernel if it corresponds to some feature mapping ϕ.

Mercer's Theorem: Let $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ be given. For K to be a valid kernel, it is necessary and sufficient that for any $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(n)}\}, n < \infty$, the corresponding kernel matrix is symmetric and positive semi-definite.

In general any learning problem formulated in terms of an inner product $< \mathbf{x}_i, \mathbf{x}_j >$ can also be expressed in terms of the inner product of the data mapped to a higher dimensional space $< \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) >$.

Kernel similarity matrices

A kernel function K is a valid kernel if it corresponds to some feature mapping ϕ.

Mercer's Theorem: Let $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ be given. For K to be a valid kernel, it is necessary and sufficient that for any $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(n)}\}, n < \infty$, the corresponding kernel matrix is symmetric and positive semi-definite.

In general any learning problem formulated in terms of an inner product $< \mathbf{x}_i, \mathbf{x}_j >$ can also be expressed in terms of the inner product of the data mapped to a higher dimensional space $< \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) >$.

To identify non-linearly separable data, the SVM dual objective can then be expressed as:

$$\max_{\alpha} \sum_{i,j=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} < \phi(\mathbf{x}^{(i)}), \phi(\mathbf{x}^{(j)}) >$$

$$\text{s.t., } \alpha_i \geq 0 \ \forall i \ , \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0.$$

Kernel similarity matrices

To identify non-linearly separable data, the SVM dual objective can be expressed as:

$$\max_{\alpha} \sum_{i,j=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} < \phi(\mathbf{x}^{(i)}), \phi(\mathbf{x}^{(j)}) >$$

$$\text{s.t., } \alpha_i \geq 0 \ \forall i \ , \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0.$$

Equivalently, in terms of the kernel function:

$$\max_{\alpha} \sum_{i,j=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

$$\text{s.t., } \alpha_i \geq 0 \ \forall i \ , \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0.$$

An optimization method will require $O(n^2)$ time to compute all $K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$.

References

[1] Andrwe Ng., CS229 *Support Vector Machines* Lecture Notes.

[2] John Shawe-Taylor, Nello Christianini. (2004) Kernel Methods for Pattern Analysis. Cambridge University Press.