Kernel Methods

As before we assume a space $\mathcal{X}$ of objects and a feature map $\Phi : \mathcal{X} \rightarrow R^D$. We also assume training data $\langle x_1, y_1 \rangle, \ldots \langle x_N, y_N \rangle$ and we define the data matrix Φ by defining $\Phi_{t,i}$ to be $\Phi_i(x_t)$. In this section we assume L_2 regularization and consider only training algorithms of the following form.

$$w^* = \operatorname*{argmin}_w \sum_{t=1}^{N} L_t(w) + \frac{1}{2}\lambda ||w||^2 \quad (1)$$

$$L_t(w) = L(y_t, w \cdot \Phi(x_t)) \quad (2)$$

We have written $L_t = L(y_t, w \cdot \Phi)$ rather then $L(m_t(w))$ because we now want to allow the case of regression where we have $y \in R$ as well as classification where we have $y \in \{-1, 1\}$.

Here we are interested in methods for solving (1) for $D >> N$. An example of $D >> N$ would be a database of one thousand emails where for each email x_t we have that $\Phi(x_t)$ has a feature for each word of English so that D is roughly 100 thousand. We are interested in the case where inner products of the form $K(x, y) = \Phi(x) \cdot \Phi(y)$ can be computed efficiently independently of the size of D. This is indeed the case for email messages with feature vectors with a feature for each word of English.

1 The Kernel Method

We can reduce $||w||$ while holding $L(y_t, w \cdot \Phi(x_t))$ constant (for all t) by removing any the component of w orthoganal to all vectors $\Phi(x_t)$. Without loss of generality we can therefore assume that w^* is in the span of the vectors $\Phi(x_t)$. More specifically, we can assume that there exists a weight vector α^* such that we have the following.

$$w^* = \sum_{t=1}^{T} \alpha_t^* \Phi(x_t) \quad (3)$$

$$w^* = \Phi^T \alpha^* \quad (4)$$

Equation (3) is called the representor theorem. Any time series α can be viewed as a representation of a feature vector $w = \sum_t \alpha_t \Phi(x_t) = \Phi^T \alpha$. By abuse of notation we will write $f_\alpha(x)$ as an abbreviation for $f_{\Phi^T \alpha}(x)$ which can be computed as follows.

$$
\begin{aligned}
f_\alpha(x) &= (\Phi^T \alpha) \cdot \Phi(x) \\
&= \sum_{t=1}^{N} \alpha_t \Phi(x_t) \cdot \Phi(x) \\
&= \sum_{t=1}^{N} \alpha_t K(x, x_t) \qquad (5)
\end{aligned}
$$

It is important to note that (5) implies that $f_\alpha(x)$ can be computed without explicitly computing any feature vectors. We also note the following.

$$
\begin{aligned}
||\Phi^T \alpha||^2 &= (\Phi^T \alpha) \cdot (\Phi^T \alpha) \\
&= \alpha^T \Phi \Phi^T \alpha \\
&= \alpha K \alpha \qquad (6) \\
K &= \Phi \Phi^T \\
K_{s,t} &= K(x_s, x_t) \qquad (7)
\end{aligned}
$$

It is important to note that the matrix K can be computed without explicitly computing any feature vectors. Hence $||w||^2 = \alpha K \alpha$ can also be computed without explicitly computing feature vectors. We now have that the training algrithm (1) can be rewritten as follows.

$$
\begin{aligned}
\alpha^* &= \operatorname*{argmin}_{\alpha} \sum_{t=1}^{N} L_t(\alpha) + \frac{1}{2} \lambda \alpha K \alpha \qquad (8) \\
L_t(\alpha) &= L(y_t, f_\alpha(x_t)) \\
&= L(y_t, (K\alpha)_t) \qquad (9)
\end{aligned}
$$

The algorithm (8) is now defined as an optimization problem over α. This problem is convex in α if $L(y, z)$ is convex in z. So for any convex version of (1) we get a convex “kernelized” training algorithm defined by (8).

2 A General Expression for α^*

In the case where the loss function $L(y,z)$ is differentiable we can get an expression for α_t by setting the gradiant the right hand side of (1) to zero. This gives the following.

$$\begin{aligned}
0 &= \sum_{t=1}^{N} L_t'(w^*)\Phi(x_t) + \lambda w^* \\
L_t'(w) &= \left.\frac{\partial L(y,z)}{\partial z}\right|_{y=y_t,\ z=w\cdot\Phi(x_t)} \\
w^** &= \frac{1}{\lambda}\sum_{t=1}^{D} -L_t'(w^*)\Phi(x_t) \qquad (10)\\
\alpha_t^* &= -\frac{1}{\lambda}L_t'(\alpha^*) \qquad (11)
\end{aligned}$$

Equation (11) gives N equations in N unknowns and in principle can be used to solve for α^*. Equation (11) also gives insight into the weights α_t. For square loss we get that α_t is propotional to the residual at point x_t (see the next section). For sigmoidal loss we get that $\alpha_t \approx 0$ for $|m_t(\alpha^*)| >> 1$ and $\alpha_t \approx \frac{y_t}{\lambda}$ for $m_t(\alpha^*) \approx 0$. For log loss we get that $\alpha_t \approx 0$ for $m_t(\alpha^*) >> 1$ and $\alpha_t \approx \frac{y_t}{\lambda}$ for $m_t(\alpha^*) << -1$.

3 Kernel Regression

In the case of square loss equation (11) can be solved in closed form.

$$\begin{aligned}
L(y,z) &= \frac{1}{2}(z-y)^2 \\
L'(y,z) &= (z-y) \\
\alpha_t^* &= -\frac{1}{\lambda}((K\alpha^*)_t - y_t) \qquad (12)\\
\lambda\alpha^* &= -(K\alpha^* - y) \\
\alpha^* &= (K+\lambda I)^{-1}y \qquad (13)
\end{aligned}$$

Note that (12) states that α_t is proportional to the residual $y_t - f_{\alpha^*}(x_t)$.

4 Kernel SVMs

In the case of hinge loss (the SVM case) the kernel optimization problem (8) becomes a convex quadratic program for which good optimization methods exist. To see this we first rewrite (8) explicitly in terms of hinge loss.

$$\alpha^* = \underset{\alpha}{\operatorname{argmin}} \sum_{t=1}^{N} \max(0, 1 - y_t(K\alpha)_t) + \frac{1}{2}\lambda\alpha K\alpha \tag{14}$$

$$\tag{15}$$

This optimization problem can be refurmulatd equivalently as follows.

$$\begin{aligned} \text{minimize} \quad & \sum_{t=1}^{N} \eta_t + \frac{1}{2}\lambda\alpha K\alpha \\ \text{subject to} \quad & \eta_t \geq 0 \\ & \eta_t \geq y_t(K\alpha)_t \end{aligned} \tag{16}$$

This is minimization of a convex quadratic objective function subject to linear constraints — a convex quadratic program.

5 Kernels without Features

We now onsider the case of $D = \infty$. The space ℓ_2 is defined to be the set of sequences w_1, w_2, w_3, ... which have finite norm, i.e., where we have the following.

$$||w||^2 = \sum_{i=1}^{\infty} w_i^2 < \infty \tag{17}$$

We are now interested in regression and classsification with infinite dimensional feature vectors and weight parameters. In other words we have $\Phi(x) \in \ell_2$ and $w \in \ell_2$. In practice there is little difference between the infinite dimensional case and the finite dimensional case with $D >> N$.

> **Definition**: A function K on $\mathcal{X} \times \mathcal{X}$ is called a *kernel function* if there exists a function Φ mapping $\mathcal{X}$ into ℓ_2 such that for any x_1, $x_2 \in \mathcal{X}$ we have that $K(x_1, x_2) = \Phi(x_1) \cdot \Phi(x_2)$.

We will show below that for $x_1, x_2 \in R^q$ the functions $(x_1 \cdot x_2 + 1)^p$ and $\exp(-\frac{1}{2}(x_1 - x_2)^T \Sigma^{-1}(x_1 - x_2))$ are both kernels. The first is called a polynomial kernel and the second is called a Gaussian kernel. The Gaussian kernel is particularly widely used. For the Gaussian kernel we have that $K(x_1, x_2) \leq 1$ where the equality is achieved when $x_1 = x_2$. In this case $K(x_1, x_2)$ expresses a nearness of x_1 to x_2. When K is a Gaussian kernel we get that $f_\alpha(x)$ as computed by (eqn:falpha) can be viewed as a classifying x using a weighted nearest neighbor rule where $K(x, x_t)$ is interpreted as giving the "nearness" of x to x_t. Empirically (8) works better for setting the weights α_t than other weight setting heuristics for weighted nearest neighbor rules.

6 Some Closure Properties on Kernels

Note that any kernel function K must be symmmetric, i.e., $K(x_1, x_2) = K(x_2, x_1)$. It must also be positive semidefinite, i.e., $K(x, x) \geq 0$.

If K is a kernel and $\alpha > 0$ then αK is also a kernel. To see this let Φ be a feature map for K. Define Φ_2 so that $\Phi_2(x) = \sqrt{\alpha}\Phi_1(x)$. We then have that $\Phi_2(x_1) \cdot \Phi_2(x_2) = \alpha K(x_1, x_2)$. Note that for $\alpha < 0$ we have that αK is not positive semidefinite and hence cannot be a kernel.

If K_1 and K_2 are kernels then $K_1 + K_2$ is a kernel. To see this let Φ_1 be a feature map for K_1 and let Φ_2 be a feature map for K_2. Let Φ_3 be the feature map defined as follows.

$$
\begin{aligned}
\Phi_3(x) &= f_1(x), g_1(x), f_2(x), g_2(x), f_3(x), g_3(x), \ldots \\
\Phi_1(x) &= f_1(x), f_2(x), f_3(x), \ldots \\
\Phi_2(x) &= g_1(x), g_2(x), g_3(x), \ldots
\end{aligned}
$$

We then have that $\Phi_3(x_1) \cdot \Phi_3(x_2)$ equals $\Phi_1(x_1) \cdot \Phi_1(x_2) + \Phi_2(x_1) \cdot \Phi_2(x_2)$ and hence Φ_3 is the desired feature map for $K_1 + K_2$.

If K_1 and K_2 are kernels then so is the product $K_1 K_2$. To see this let Φ_1 be a feature map for K_1 and let Φ_2 be the feature map for K_2. Let $f_i(x)$ be the ith feature value under feature map Φ_i and let $g_i(x)$ be the ith feature value under the feature map Φ_2. We now have the following.

$$
\begin{aligned}
K_1(x_1, x_2) K_2(x_1, x_2) &= (\Phi_1(x_1) \cdot \Phi_1(x_2))(\Phi_2(x_1) \cdot \Phi_2(x_2)) \\
&= \left(\sum_{i=1}^{\infty} f_i(x_1) f_i(x_2)\right) \left(\sum_{j=1}^{\infty} g_j(x_1) g_j(x_2)\right) \\
&= \sum_{i,j} f_i(x_1) f_i(x_2) g_j(x_1) g_j(x_2) \\
&= \sum_{i,j} (f_i(x_1) g_j(x_1)) \quad (f_i(x_2) g_j(x_2))
\end{aligned}
$$

We can now define a feature map Φ_3 with a feature $h_{i,j}(x)$ or each pair $\langle i, j \rangle$ defined as follows.

$$h_{i,j}(x) = f_i(x)g_j(x)$$

. We then have that $K_1(x_1, x_2)K_2(x_1, x_2)$ is $\Phi_3(x_1) \cdot \Phi_3(x_2)$ where the inner product sums over all pairs $\langle i, j \rangle$. Since the number of such pairs is countable, we can enumerate the pairs in a linear sequence to get $\Phi_3(x) \in \ell_2$.

It follows from these closure properties that if p is a polynomial with positive coefficients, and K is a kernel, then $p(K(x_1, x_2))$ is also a kernel. This proves that polynomial kernels are kernels. One can also give a direct proof that if K is a kernel and p is a convergent infinite power series with positive coeffficients (an convergent infinite polynomial) then $p(K(x_1, x_2))$ is a kernel. The proof is similar to the proof that a product of kernels is a kernel but uses a countable set of higher order moments as features. The result for infinite power series can then be used to prove that a Gaussian kernel is a kernel. These proofs are homework problems for these notes. Unlike most proofs in the literature, we do not require compactness of the set X on which the Gaussian kernel is defined.

7 Hilbert Space

The set ℓ_2 is an infinite dimensional Hilbert space. In fact, all Hilbert spaces with a countable basis are isomorphic to ℓ_2. So ℓ_2 is really the only Hilbert space we need to consider. But different feature maps yield different interpretations of the space ℓ_2 as functions on $\mathcal{X}$. A particularly interesting feature map is the following.

$$\Phi(x) = 1, x, \frac{x^2}{\sqrt{2}}, \frac{x^3}{\sqrt{3!}}, \ldots, \frac{x^n}{\sqrt{n!}}, \ldots$$

Now consider any function f all of whose derivatives exist at 0. Define $w(f)$ to be the following infinite sequence.

$$w(f) = f(0), f'(0), \frac{f''(0)}{\sqrt{2}}, \ldots, \frac{f^k(0)}{\sqrt{k!}}, \ldots$$

For any f with $w(f) \in \ell_2$ (which is many familiar functions) we have the following.

$$f(x) = w(f) \cdot \Phi(x) \tag{18}$$

So under this feature map, the parameter vectors w in ℓ_2 represent essentially all functions whose Taylor series converges. For any given feature map Φ on $\mathcal{X}$ define $\mathcal{H}(\Phi)$ to be the set of functions f from $\mathcal{X}$ to R such that there exists a parameter vector $w(f) \in \ell_2$ satisfying (18). Equation (1) can then be written as follows where $||f||^2$ abbreviates $||w(f)||^2$.

$$f^* = \underset{f \in \mathcal{H}(\Phi)}{\operatorname{argmin}} \left(\sum_{t=1}^{T} L(y_t, f(x_t)) \right) + \lambda ||f||^2$$

This way of writing the equation emphasizes that with a rich feature map selecting w is equivalent to selecting a function from a rich space of functions.

8 Problems

1. Let $P(z)$ be an infinite power series (where z is a single real number) with positive coefficients such that $P(z)$ converges for all $z \in R$.

$$P(z) = \sum_{k=0}^{\infty} a_k z^k, \;\; a_k \geq 0, \;\; P(z) \text{ finite } \forall z$$

Let K be a kernel on a set $\mathcal{X}$. This problem is to show that that $P(K(x,y))$ is a kernel on $\mathcal{X}$.

a. Let $\Phi : \mathcal{X} \rightarrow \ell_2$ be the feature map for K. Let s range over all finite sequences of positive integers where $|s|$ is the length of the sequence s and for $1 \leq j \leq |s|$ we have $s_j \geq 1$ is the jth integer in the sequence s. For $x \in \mathcal{X}$, and s a sequence of indeces, let $\Psi_s(x)$ be defined as follows.

$$Psi_s(x) = \sqrt{a_{|s|}} \prod_{j=1}^{|s|} \Phi_{s_j}(x)$$

We note that the set of all sequences s is countable and hence can be enumerated in a single infintie sequence of sequences. Hence the map Ψ maps x into an infinite series. Show that $\Psi(x) \in \ell_2$, i.e., that $\sum_s \Psi_s^2(x) < \infty$. (Consider $P(K(x,x))$).

b. Show that Ψ is the feature map for $P(K(x,y))$.

2. Here will show that the Gaussian kernel is indeed a kernel. Consider $x, y \in R^d$. The problem is to show that there exists a feature map Ψ, with $\Psi(x), \Psi(y) \in \ell_2$, such that $\exp(-\frac{1}{2}(x-y)^T\Sigma^{-1}(x-y)) = \Psi(x) \cdot \Psi(y)$.

a. Show

$$\exp\left(-\frac{1}{2}(x-y)^T\Sigma^{-1}(x-y)\right) = \exp\left(-\frac{1}{2}x\Sigma^{-1}x\right)\exp\left(-\frac{1}{2}y\Sigma^{-1}y\right)\exp\left(x\Sigma^{-1}y\right)$$

c. Show that $x\Sigma^{-1}y$ is a kernel in x and y.

b. Show that $\exp(-\frac{1}{2}x\Sigma^{-1}x)\exp(-\frac{1}{2}y\Sigma^{-1}y)$ is a kernel in x and y (Hint: you only need a single feature.)

c. Use the result of part 1 and a, b, and c, plus the result in the notes that the product of kernels is a kernel, to show that the Gaussian kernel is a kernel.