8. Kernel methods

- motivation
- kernel formulations
- kernel functions
Linear-in-parameters model

Linear-in-parameters model (in the notation of 133A, lecture 9)

\[ \theta^T F(x) = \theta_1 f_1(x) + \theta_2 f_2(x) + \cdots + \theta_p f_p(x) \]

- \( x \) is an independent variable, not necessarily a vector
- \( F(x) \) is a feature map: maps \( x \) to a \( p \)-vector of features (possibly redundant)

\[ F(x) = (f_1(x), f_2(x), \ldots, f_p(x)) \]

Training set: \( N \) data points \( x^{(1)}, \ldots, x^{(N)} \) define an \( N \times p \) data matrix

\[ A = \begin{bmatrix} F(x^{(1)})^T \\ F(x^{(2)})^T \\ \vdots \\ F(x^{(N)})^T \end{bmatrix} \]
Kernel methods

Kernel matrix

\[ Q = AA^T \]

\( Q \) is \( N \times N \) and symmetric positive semidefinite with entries

\[ Q_{ij} = F(x^{(i)})^T F(x^{(j)}), \quad i, j = 1, \ldots, N \]

Kernel function

\[ \kappa(x, y) = F(x)^T F(y) \]

in this notation, the entries of the kernel matrix are

\[ Q_{ij} = \kappa(x^{(i)}, x^{(j)}), \quad i, j = 1, \ldots, N \]

Kernel methods

- algorithms that use \( \kappa(x, y) \) and \( Q \), avoid \( F(x), A, A^T A \)
- of interest if \( N \ll p \) (including extensions to infinite-dimensional feature maps)
Polynomial kernel

$\theta^T F(x)$ is a polynomial of degree $d$ or less in $n$ variables

• here we assume $x$ is an $n$-vector

• dimension of $F(x)$ is extremely large unless $n$ or $d$ is small:

$$p = \binom{n+d}{n} = \frac{(n+d)!}{n! \cdot d!}$$

• with appropriately scaled monomials as features in $F(x)$,

$$\kappa(x, y) = F(x)^T F(y) = (1 + x^T y)^d$$

(see 133A, lecture 12)
Model fitting by regularized least squares

an example of a kernel method was discussed in 133A, lecture 12

\[
\text{minimize } \|A\theta - y^d\|^2 + \lambda \|\theta\|^2
\]

• we fit a model \( \hat{f}(x) = \theta^T F(x) \) to data points \( x^{(1)}, \ldots, x^{(N)}, y^{(1)}, \ldots, y^{(N)} \)

• \( y^d \) is the \( N \)-vector with entries \( y^{(1)}, \ldots, y^{(N)} \)

• second objective \( \lambda \|\theta\|^2 \) is added to avoid over-fitting

• optimal solution is \( \hat{f}(x) = \hat{\theta}^T F(x) \) where

\[
\hat{\theta} = (A^T A + \lambda I)^{-1} A^T y^d
\]
Kernel method for regularized least squares fitting

via the “push-through” identity the solution can be written as

$$\hat{\theta} = (A^T A + \lambda I)^{-1} A^T y_d = A^T (AA^T + \lambda I)^{-1} y_d$$

- can be computed as $\hat{\theta} = A^T \hat{w}$ where

$$\hat{w} = (Q + \lambda I)^{-1} y_d$$

- fitted model can be evaluated without referring to $A$ or $F(x)$:

$$\hat{f}(x) = \hat{\theta}^T F(x) = \hat{w}^T AF(x) = \hat{w}^T \begin{bmatrix} \kappa(x^{(1)}, x) \\ \vdots \\ \kappa(x^{(N)}, x) \end{bmatrix} = \sum_{i=1}^{N} \hat{w}_i \kappa(x^{(i)}, x)$$

this method only requires $Q$ and $\kappa$, not $A$, $F$, or $A^T A$
Principal components

another example is principal component analysis of the data matrix $A$

- compute the leading right singular vectors $v_1, \ldots, v_k$ of the data matrix $A$
- projections of $F(x)$ on principal components are $v_1^T F(x), \ldots, v_k^T F(x)$

Recursive formulation

- first right singular vector is solution $\theta$ of

\[
\begin{align*}
\text{maximize} & \quad \|A\theta\| \\
\text{subject to} & \quad \|	heta\| \leq 1
\end{align*}
\]

- other right singular vectors $v_i$ (for $i \leq \text{rank}(A)$) are solutions $\theta$ of

\[
\begin{align*}
\text{maximize} & \quad \|A\theta\| \\
\text{subject to} & \quad \|	heta\| \leq 1 \\
& \quad v_k^T \theta = 0, \quad k = 1, \ldots, i - 1
\end{align*}
\]

(equality can also be written as $u_k^T A\theta = 0$ where $u_k$ is left singular vector)
Kernel PCA

• find leading singular values, left singular vectors of $A$ via eigendecomposition

$$AA^T = Q = \sum_{i=1}^{\text{rank}(A)} \sigma_i^2 u_i u_i^T$$

• right singular vectors $v_i$ follow from

$$A^T u_i = \sigma_i v_i, \quad i = 1, \ldots, \text{rank}(A)$$

• projection of $F(x)$ on principal components can be computed as

$$v_i^T F(x) = \frac{1}{\sigma_i} u_i^T A F(x) = \frac{1}{\sigma_i} u_i^T \begin{bmatrix} \kappa(x^{(1)},x) \\ \vdots \\ \kappa(x^{(N)},x) \end{bmatrix}$$

this method only requires $Q$ and $\kappa$, not $A$, $F$, or $A^T A$
Outline

• motivation

• kernel formulations

• kernel functions
A general class of model fitting problems

we consider optimization problems in which the variable $\theta$ enters in only two ways

1. terms in objective and constraints that depend on model predictions on data set

$$A\theta = \begin{bmatrix} F(x^{(1)})^T \theta \\ \vdots \\ F(x^{(N)})^T \theta \end{bmatrix}$$

2. terms in objective that penalize $||\theta||$, or upper bounds on $||\theta||$ in the constraints

the two properties imply that we can restrict $\theta$ to the row space of $A$

- $A\theta$ only depends on component of $\theta$ in the row space of $A$
- adding a nonzero component from the nullspace of $A$ would only increase $||\theta||$

this idea can be implemented in several ways; we will discuss one approach
Factorization of kernel matrix

suppose the data matrix $A$ has rank $r$

- the data matrix $A$ can be factorized as

$$A = BC$$

where $B$ is $N \times r$ and $C$ is $r \times p$, and $C$ has orthonormal rows

- $B$ has linearly independent columns and can be computed from a factorization

$$Q = BB^T$$

- the rows of $C = B^\dagger A$ are an orthonormal basis for

$$\text{range}(A^T) = \text{span}(F(x^{(1)}), \ldots, F(x^{(N)}))$$
Reformulation of model fitting problem

every $\theta$ can be decomposed in components in the row space and nullspace of $A$:

$$\theta = C^T w + v, \quad Cv = 0$$

- the vector $A\theta$ of model predictions only depends on $w$, and not on $v$:

$$A\theta = (BC)(C^T w + v) = Bw$$

- for given $w$, the Euclidean norm of $\theta$ is minimized by setting $v = 0$:

$$\|\theta\|^2 = \|C^T w\|^2 + \|v\|^2 = \|w\|^2 + \|v\|^2$$

therefore we can set $\theta = C^T w$ in any problem of the type described on page 8.9
Change of variables

we make the substitution

\[ \theta = C^T w = (B^\dagger A)^T w \]

- \( A\theta \) is replaced by \( Bw \)
- \( \|\theta\| \) is replaced by \( \|w\| \)
- the \( r \)-vector \( w \) replaces the \( p \)-vector variable \( \theta \) (a huge reduction if \( N \ll p \))
- the model function is linearly parametrized by the optimal solution \( \hat{w} \):

\[
\hat{f}(x) = \hat{\theta}^T F(x) = \hat{w}^T B^\dagger A F(x) = \hat{w}^T B^\dagger \\
\begin{bmatrix}
\kappa(x^{(1)}, x) \\
\kappa(x^{(2)}, x) \\
\vdots \\
\kappa(x^{(N)}, x)
\end{bmatrix}
\]

this formulation only requires \( B \) (computed from \( Q \)) and \( \kappa \), not \( A \), \( A^T A \), \( F \), or \( C \)
Example: regularized least squares

\[
\text{minimize} \quad \|A\theta - y^d\|^2 + \lambda \|\theta\|^2
\]

- variable \(\theta\) is a \(p\)-vector
- solution \(\hat{\theta}\) parametrizes the fitted model \(\hat{f}(x) = \hat{\theta}^T F(x)\)

**Kernel method:** solve reformulated problem

\[
\text{minimize} \quad \|Bw - y^d\|^2 + \lambda \|w\|^2
\]

- \(N \times r\) matrix \(B\) is full-rank factor of kernel matrix \(Q = BB^T\)
- variable \(w\) is an \(r\)-vector, where \(r = \text{rank}(Q) \leq N\)
- from solution \(\hat{w}\), we obtain fitted model

\[
\hat{f}(x) = \hat{w}^T B^\dagger \begin{bmatrix}
\kappa(x^{(1)}, x) \\
\vdots \\
\kappa(x^{(N)}, x)
\end{bmatrix}
\]
Approximation problems

model fitting with non-Euclidean norms or non-quadratic penalty functions

\[
\text{minimize} \quad h(A\theta - y_d) + \lambda \|\theta\|^2
\]

Examples

\[h(u) = \|u\|_1\] or a smooth approximation

\[h(u) = \sum_{i=1}^{N} \phi(u_i)\]

Kernel method

• solve problem in \(r\)-vector variable \(w\) (for example, using Newton’s method)

\[
\text{minimize} \quad h(Bw - y_d) + \lambda \|w\|^2
\]

• note that no assumptions are imposed on \(h\)
Nonlinear least squares example

another example from 133A (lecture 13)

\[
\text{minimize } \sum_{i=1}^{N} \left( \phi(F(x^{(i)}^T \theta) - y^{(i)})^2 + \lambda \|\theta\|^2 \right)
\]

- \(y^{(i)} \in \{-1, 1\}\) are labels for two classes in a Boolean classification problem
- \(\phi(u)\) is sigmoidal function (a smooth approximation of \(\text{sign}(u)\))

\[
\phi(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}
\]

**Kernel method:** solve the nonlinear least squares problem in \(r\)-vector variable \(w\)

\[
\text{minimize } \sum_{i=1}^{N} (\phi((Bw)_i) - y^{(i)})^2 + \lambda \|w\|^2
\]
Boolean classification

the goal is to find a nonlinear decision function $\theta^T F(x)$ for a Boolean classifier:

$$\hat{f}(x) = 1 \quad \text{if} \quad \theta^T F(x) > 0, \quad \hat{f}(x) = -1 \quad \text{if} \quad \theta^T F(x) < 0$$

Maximum-margin classifier

- given $N$ examples $x^{(i)}$ with labels $y^{(i)} \in \{-1, 1\}$, find $\theta$ by solving

  $$\begin{align*}
    \text{minimize} & \quad \|\theta\|^2 \\
    \text{subject to} & \quad \theta^T F(x^{(i)}) \geq 1 \quad \text{if} \quad y^{(i)} = 1 \\
    & \quad \theta^T F(x^{(i)}) \leq -1 \quad \text{if} \quad y^{(i)} = -1
  \end{align*}$$

- in matrix–vector form, if $y^d = (y^{(1)}, \ldots, y^{(N)})$ and $A$ has rows $F(x^{(i)})^T$, 

  $$\begin{align*}
    \text{minimize} & \quad \|\theta\|^2 \\
    \text{subject to} & \quad \text{diag}(y^d) A \theta \geq 1
  \end{align*}$$

  this is a quadratic program
Kernel formulation of maximum-margin classifier

solve a quadratic program in $r$-vector variable $w$:

$$
\begin{align*}
\text{minimize} & \quad \|w\|^2 \\
\text{subject to} & \quad \text{diag}(y^d)Bw \geq 1
\end{align*}
$$

- $B$ is computed from a kernel matrix factorization $Q = AA^T = BB^T$
- optimal solution $\hat{w}$ determines the nonlinear decision function $\tilde{f}(x) = \hat{\theta}^T F(x)$:

$$
\tilde{f}(x) = \hat{w}^T B^+ \begin{bmatrix}
\kappa(x^{(1)}, x) \\
\vdots \\
\kappa(x^{(N)}, x)
\end{bmatrix}
$$

- Boolean classifier returns

$$
\hat{f}(x) = 1 \quad \tilde{f}(x) > 0, \quad \hat{f}(x) = -1 \quad \tilde{f}(x) < 0
$$
Support vector machine classifier

a variation on the maximum-margin classifier: compute $\theta$ from

$$\text{minimize} \sum_{i=1}^{N} \max \left\{ 0, 1 - y^{(i)}(\theta^T F(x^{(i)})) \right\} + \lambda \|\theta\|^2$$

instead of imposing hard constraints

$$\theta^T F(x^{(i)}) \geq 1 \text{ if } y^{(i)} = 1, \quad \theta^T F(x^{(i)}) \leq -1 \text{ if } y^{(i)} = -1$$

we impose a penalty on misclassified points:

$$\max \{0, 1 - u_i\}, \quad \max \{0, 1 + u_i\}$$
Kernel formulation of support vector machine classifier

first term in support vector machine objective is a function of $A\theta$:

$$\minimize \sum_{i=1}^{N} \max\{0, 1 - y^{(i)}(A\theta)_i\} + \lambda \|\theta\|^2$$

Kernel formulation

$$\minimize \sum_{i=1}^{N} \max\{0, 1 - y^{(i)}(Bw)_i\} + \lambda \|w\|^2$$

- $B$ is a full-rank factor of the kernel matrix $Q = BB^T$
- variable $w$ is an $r$-vector
- from optimal $\hat{w}$ we directly find the decision function

$$\hat{\theta}^T F(x) = \hat{w}^T B^\dagger \begin{bmatrix} \kappa(x^{(1)}, x) \\ \vdots \\ \kappa(x^{(N)}, x) \end{bmatrix}$$
Outline

• motivation

• kernel formulations

• kernel functions
Kernel property

A kernel function must have the property that the matrix $Q$ with entries

$$Q_{ij} = \kappa(x^{(i)}, x^{(j)}) \quad i, j = 1, \ldots, N$$

is symmetric positive semidefinite for every finite set of points $x^{(1)}, \ldots, x^{(N)}$

Examples

- Gaussian kernel
  $$\kappa(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

- Products and nonnegative sums of kernel functions (see homework 2)

- A polynomial $q$ with nonnegative coefficients applied to the inner product $x^T y$:
  $$\kappa(x, y) = q(x^T y)$$
From kernel to feature map

suppose $\kappa$ is a function with the kernel property on page 8.20

• it can be shown that there exists a feature map $F$ such that

$$\kappa(x, y) = \langle F(x), F(y) \rangle$$

• however, in general the feature map $F(x)$ has infinite dimension

Finite-dimensional feature map

• for any given data set $x^{(1)}, \ldots, x^{(N)}$ we can construct a feature map $F$ such that

$$\kappa(x^{(i)}, x) = F(x^{(i)})^T F(x) \quad \text{for all } x \text{ and for } i = 1, \ldots, N$$

• $F(x)$ can be chosen to have dimension $r = \operatorname{rank}(Q)$
Constructing a finite-dimensional feature map

we are given a kernel function $\kappa$ and $N$ points $x^{(1)}, \ldots, x^{(N)}$

• construct the $N \times N$ kernel matrix $Q$

$$Q_{ij} = \kappa(x^{(i)}, x^{(j)}), \quad i, j = 1, \ldots, N$$

• factor $Q$ as $Q = BB^T$ with $B$ an $N \times r$ matrix and $r = \text{rank}(Q)$

• define the feature map

$$F(x) = B^\dagger \begin{bmatrix} \kappa(x^{(1)}, x) \\ \kappa(x^{(2)}, x) \\ \vdots \\ \kappa(x^{(N)}, x) \end{bmatrix}$$

on the next page we show that $F(x^{(i)})^T F(x) = \kappa(x^{(i)}, x)$ for all $x$ and $i = 1, \ldots, N$
Proof

• the vectors $F(x^{(1)}), \ldots, F(x^{(N)})$ are the transposes of the rows of $B$:

$$F(x^{(i)}) = B^\dagger \begin{bmatrix} \kappa(x^{(1)}, x^{(i)}) \\ \vdots \\ \kappa(x^{(N)}, x^{(i)}) \end{bmatrix} = B^\dagger Q e_i = B^\dagger (BB^T) e_i = B^T e_i$$

• consider any $x$ and define $d = \begin{bmatrix} \kappa(x^{(1)}, x) \\ \vdots \\ \kappa(x^{(N)}, x) \end{bmatrix}$

• by the kernel property the following matrix is positive semidefinite

$$\begin{bmatrix} Q & d \\ d^T & \kappa(x, x) \end{bmatrix} = \begin{bmatrix} BB^T & d \\ d^T & \kappa(x, x) \end{bmatrix}$$

• this implies that $d \in \text{range}(B)$, i.e., $BB^\dagger d = d$, and therefore

$$F(x^{(i)})^T F(x) = e_i^T BB^\dagger d = e_i^T d = \kappa(x^{(i)}, x)$$