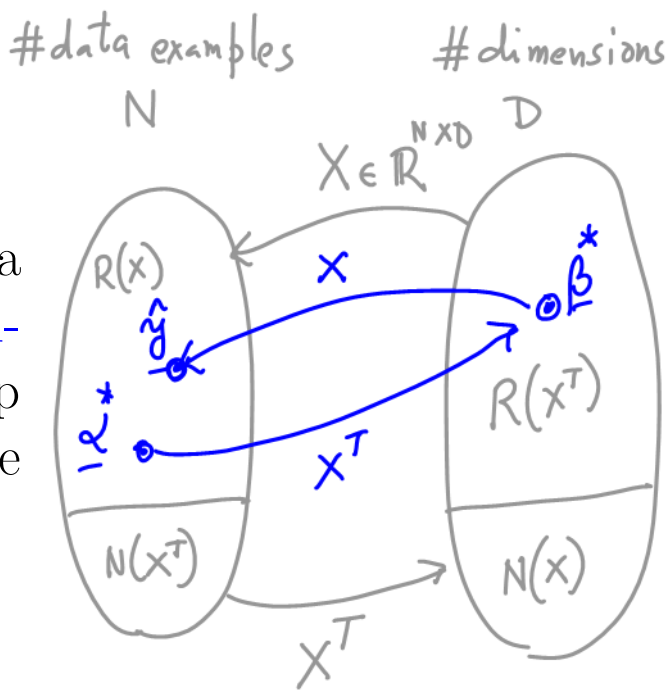
Kernel Ridge Regression

Mohammad Emtiyaz Khan
EPFL

Oct 27, 2015



Motivation

The ridge solution $\boldsymbol{\beta}^* \in \mathbb{R}^D$ has a counterpart $\boldsymbol{\alpha}^* \in \mathbb{R}^N$. Using duality, we will establish a relationship between $\boldsymbol{\beta}^*$ and $\boldsymbol{\alpha}^*$ which leads the way to kernels.

Ridge regression

Throughout, we assume that there is no intercept term β_0 to make the math easier.

The following is true for ridge reg:

$$\boldsymbol{\beta}^* = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_D)^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I}_N)^{-1}\mathbf{y} := \mathbf{X}^T\boldsymbol{\alpha}^*,$$

where $\boldsymbol{\alpha}^* := (\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I}_N)^{-1}\mathbf{y}$.

(A) (B) $\boldsymbol{\alpha}^* \in \mathbb{R}^N$

This can be proved using the following identity: let $\mathbf{P}$ be an $N \times M$ matrix while $\mathbf{Q}$ be an $M \times N$ matrix,

$$(\mathbf{P}\mathbf{Q} + \mathbf{I}_N)^{-1}\mathbf{P} = \mathbf{P}(\mathbf{Q}\mathbf{P} + \mathbf{I}_M)^{-1}.$$

$(X^TX + I_D)^{-1} X^T \quad X^T(XX^T + I_N)^{-1}$

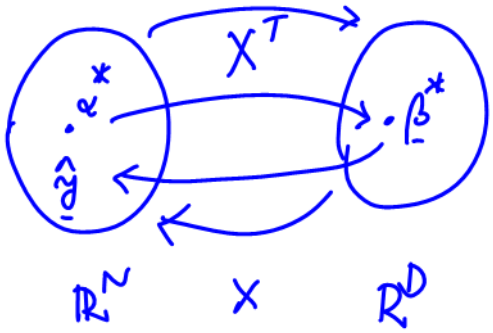
Multiply by $(I_N + PQ)^{-1}$ & $(I_M + QP)^{-1}$

$[P + PQP]^{-1} = [(I_N + PQ)P]^{-1} = [P(I_M + QP)]^{-1}$

What are the computational complexities for these two ways of computing $\boldsymbol{\beta}^*$?

For (A): $O(D^2N + D^3)$, For (B): $O(N^2D + N^3)$

With this, we know that $\boldsymbol{\beta}^* = \mathbf{X}^T\boldsymbol{\alpha}^*$ lies in the row space of $\mathbf{X}$. Previously, we have seen that $\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}^*$ lies in the column space of $\mathbf{X}$. In other words,



$$\boldsymbol{\beta}^* = \sum_{n=1}^{N} \alpha_n^* \mathbf{x}_n, \quad \hat{\mathbf{y}} = \sum_{d=1}^{D} \beta_d^* \bar{\mathbf{x}}_d$$

$$\text{where } \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}$$
$$= \begin{bmatrix} \bar{\mathbf{x}}_1 & \bar{\mathbf{x}}_2 & \dots & \bar{\mathbf{x}}_D \end{bmatrix}$$

The representer theorem

The representer theorem generalizes this result: for a $\boldsymbol{\beta}^*$ minimizing the following function for any $\mathcal{L}$,

$$\min_{\beta} \sum_{n=1}^{N} \mathcal{L}(y_n, \mathbf{x}_n^T\boldsymbol{\beta}) + \sum_{j=1}^{D} \lambda\beta_j^2$$

there exists $\boldsymbol{\alpha}^*$ such that $\boldsymbol{\beta}^* = \mathbf{X}^T\boldsymbol{\alpha}^*$.

$\underline{\beta}^* = X^T \underline{\alpha}$ implies for Ridge regression that

$\underline{x}_*^T \underline{\beta}^* = \sum_{n=1}^{N} \alpha_n (\underline{x}_*^T \underline{x}_n)$

optimal predictor; test input; train i/p

In general,

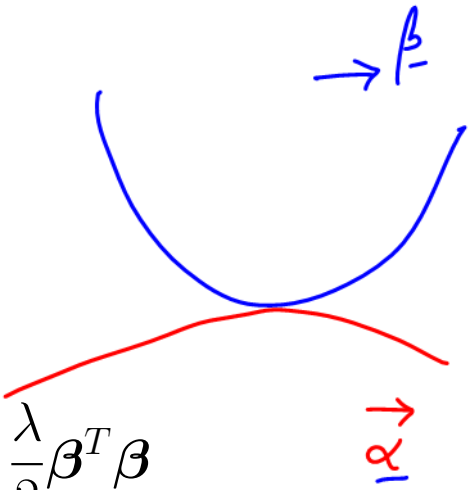
$f^*(\underline{x}_*) = \sum_{n=1}^{N} \alpha_n k(\underline{x}_*, \underline{x}_n)$

optimal predictor; test input; kernel; train inputs

Therefore, no need to work with X.

See even more general statement in Wikipedia, originally proved in Scholkopf, Herbrich and Smola (2001).

Kernelized ridge regression



The representer theorem allows us to write an equivalent optimization problem in terms of $\boldsymbol{\alpha}$. For example, for ridge regression, the following two problems are equivalent:

$$\boldsymbol{\beta}^* = \arg\min_{\beta} \quad \tfrac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}$$

$$\boldsymbol{\alpha}^* = \arg\max_{\alpha} \quad -\tfrac{1}{2}\boldsymbol{\alpha}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I}_N)^T\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\mathbf{y}$$

i.e. they both return the same optimal value and there is a one-to-one mapping between $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$. Note that the optimization over $\boldsymbol{\alpha}$ is a *maximization* problem.

Most importantly, the second problem is expressed in terms of the matrix $\mathbf{X}\mathbf{X}^T$. This is our first example of a kernel matrix.

Note: We don't give a detailed derivation of the second problem, but to show the equivalence, you can show that we obtain equal optimal values for the two problems. We will do a derivation later using the duality principle. You can see a derivation here http://www.ics.uci.edu/~welling/classnotes/papers_class/Kernel-Ridge.pdf.

Advantages of kernelized ridge regression

First, it might be computationally efficient in some cases when solving the system of equations.

$O(ND^2 + D^3)$, $N \gg D$

$O(N^2D + N^3)$, $N \ll D$

Second, by defining $\mathbf{K} = \mathbf{X}\mathbf{X}^T$, we can work directly with $\mathbf{K}$ and never have to worry about $\mathbf{X}$. This is the kernel trick.

(2)

→ explained in the following sections.

Third, working with $\boldsymbol{\alpha}$ is sometimes advantageous (e.g. in SVMs many entries of $\boldsymbol{\alpha}$ will be zero).

(3)

→ We will see this in SVM

Kernel functions

The linear kernel is defined below:

$$\mathbf{K} = \mathbf{X}\mathbf{X}^T = \begin{bmatrix} \mathbf{x}_1^T\mathbf{x}_1 & \mathbf{x}_1^T\mathbf{x}_2 & \dots & \mathbf{x}_1^T\mathbf{x}_N \\ \mathbf{x}_2^T\mathbf{x}_1 & \mathbf{x}_2^T\mathbf{x}_2 & \dots & \mathbf{x}_2^T\mathbf{x}_N \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_N^T\mathbf{x}_1 & \mathbf{x}_N^T\mathbf{x}_2 & \dots & \mathbf{x}_N^T\mathbf{x}_N \end{bmatrix}. \quad X = \begin{bmatrix} \underline{x}_1^T \\ \underline{x}_2^T \\ \vdots \\ \underline{x}_N^T \end{bmatrix}$$

Kernel with basis functions $\boldsymbol{\phi}(\mathbf{x})$: $\mathbb{R}^D \to \mathbb{R}^M$ with $\mathbf{K} := \boldsymbol{\Phi}\boldsymbol{\Phi}^T$ is shown below:

$$\begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}_1)^T\boldsymbol{\phi}(\mathbf{x}_1) & \boldsymbol{\phi}(\mathbf{x}_1)^T\boldsymbol{\phi}(\mathbf{x}_2) & \dots & \boldsymbol{\phi}(\mathbf{x}_1)^T\boldsymbol{\phi}(\mathbf{x}_N) \\ \boldsymbol{\phi}(\mathbf{x}_2)^T\boldsymbol{\phi}(\mathbf{x}_1) & \boldsymbol{\phi}(\mathbf{x}_2)^T\boldsymbol{\phi}(\mathbf{x}_2) & \dots & \boldsymbol{\phi}(\mathbf{x}_2)^T\boldsymbol{\phi}(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\phi}(\mathbf{x}_N)^T\boldsymbol{\phi}(\mathbf{x}_1) & \boldsymbol{\phi}(\mathbf{x}_N)^T\boldsymbol{\phi}(\mathbf{x}_2) & \dots & \boldsymbol{\phi}(\mathbf{x}_N)^T\boldsymbol{\phi}(\mathbf{x}_N) \end{bmatrix}.$$

The kernel trick

A big advantage of using kernels is that we do not need to specify $\boldsymbol{\phi}(\mathbf{x})$ explicitly, since we can work directly with $\mathbf{K}$.

We will use a kernel function $k(\mathbf{x}, \mathbf{x}')$ and compute the (i, j)th entry of $\mathbf{K}$ as follows: $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. For example, for linear kernel and basis function expansion, the kernel function is the following:

$$k(\mathbf{x}, \mathbf{x}') := \mathbf{x}^T\mathbf{x}', \qquad k(\mathbf{x}, \mathbf{x}') := \boldsymbol{\phi}(\mathbf{x})^T\boldsymbol{\phi}(\mathbf{x}')$$

However, a kernel function k is usually associated with a $\boldsymbol{\phi}$, e.g. $k(x, x') = x^2(x')^2$ corresponds to $\phi(x) = x^2$ and $k(\mathbf{x}, \mathbf{x}') = (x_1x_1' + x_2x_2' + x_3x_3')^2$ corresponds to

$$\boldsymbol{\phi}(\mathbf{x})^T = [\, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3 \,]$$

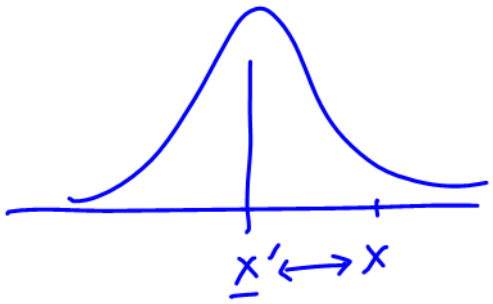
The good news is that the evaluation of a kernel is usually faster with k than with $\boldsymbol{\phi}$.

Ex: Show that this ϕ leads to this k.

Check it in the above example.

Examples of kernels

The above kernel is an example of the polynomial kernel. Another example is the Radial Basis Function (RBF) kernel.



$$k(\mathbf{x}, \mathbf{x}') = \exp\left[-\tfrac{1}{2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')\right]$$

what is ϕ? well, we don't care!

↳ See more examples in Section 14.2 of the KPM book.

A natural question is the following: how can we ensure that there exists a $\boldsymbol{\phi}$ corresponding to a given kernel $\mathbf{K}$? The answer is: as long as a kernel satisfy certain properties.

Properties of a kernel

A kernel function must be an inner-product in some feature space. Here are a few properties that ensure it is the case.

1. $\mathbf{K}$ should be symmetric, i.e. $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$.

2. For any arbitrary input set $\{\mathbf{x}_n\}$ and all N, $\mathbf{K}$ should be positive semidefinite.

Use the definition of p.s.d. $\underline{t}^T K \underline{t} \geq 0$, to show $\sum_{ij} t_i x_i^2 x_j^2 t_j > 0$.

An important subclass is the positive-definite kernel functions, giving rise to infinite-dimensional feature spaces.

Read about Mercer and Matern kernels from Kevin Murphy's Section 14.2. There is a small note about Reproducing kernel Hilbert Space in the website (written by Matthias Seeger), please read that as well.

To do

1. Clearly understand the relationship $\boldsymbol{\beta}^* = \mathbf{X}^T \boldsymbol{\alpha}^*$. Understand the statement of the representer theorem.
2. Show that ridge regression and kernel ridge regression are equivalent. Hint: show that the optimization problems corresponding to $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ have the same optimal value.
3. Get familiar with various examples of kernels. See Section 6.2 of Bishop on examples of kernel construction. Read Section 14.2 of KPM book for examples of kernels.
4. Revise and understand the difference between positive-definite and positive-semidefinite matrices.
5. If curious about infinite $\boldsymbol{\phi}$, see Matthias Seeger's notes (uploaded on the website).