Machine Learning Course - CS-433

Kernel Ridge Regression and the Kernel Trick

Nov 8, 2018

changes by Martin Jaggi 2016

minor changes by Martin Jaggi 2017

changes by Ruediger Urbanke 2018

Last updated: November 7, 2018

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Motivation

In our last lecture we have formulated the optimization problem corresponding to SVMs. We then derived an alternative formulation using duality. We have seen that in this dual formulation the data only appears in the form of a "kernel" $\mathbf{K} = \mathbf{X}\mathbf{X}^\top$.
The aim of today is the following. First, we will discuss a second problem that admits a "dual" formulation, namely ridge regression. Second, we will see that also for this problem, the data in the dual formulation only enters in form of the kernel $\mathbf{K} = \mathbf{X}\mathbf{X}^\top$. We say that such a problem is "kernelized." Third, we will see that for any kernelized problem we can apply the *kernel trick*. This trick will allow us to use a significantly augmented feature vector without incuring extra costs.

Alternative formulation of ridge regression

Recall the ridge regression problem

$$\min_{\mathbf{w}} \quad \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \tfrac{\lambda}{2}\|\mathbf{w}\|^2,$$

and its solution

$$\mathbf{w}^\star = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_D)^{-1}\mathbf{X}^\top\mathbf{y}.$$

We claim that this solution can be written in the alternative form

$$\mathbf{w}^\star = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}_N)^{-1}\mathbf{y}.$$

This second formulation can be proved using the following identity: let $\mathbf{P}$ be an $N \times M$ matrix and $\mathbf{Q}$ be an $M \times N$ matrix. Then, trivially,

$$\mathbf{P}(\mathbf{QP} + \mathbf{I}_M) = \mathbf{PQP} + \mathbf{P} = (\mathbf{PQ} + \mathbf{I}_N)\mathbf{P}.$$

If we now assume that $(\mathbf{QP} + \mathbf{I}_M)$ and $(\mathbf{PQ} + \mathbf{I}_N)$ are invertible we have the identity

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

To derive from this general statement our alternative representation, let $\mathbf{P} = \mathbf{X}^\top$ and $\mathbf{Q} = \frac{1}{\lambda}\mathbf{X}$.
Why is this alternative representation useful?

1. Define $\boldsymbol{\alpha}^\star := (\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}_N)^{-1}\mathbf{y}$. Then we can write

   $$\mathbf{w}^\star := \mathbf{X}^\top\boldsymbol{\alpha}^\star.$$

   From this representation we see that $\mathbf{w}^\star$ lies in the column space of $\mathbf{X}^\top$, i.e., the space spanned by the feature vectors. Previously, we had already seen that $\hat{\mathbf{y}} = \mathbf{X}^\top\mathbf{w}^\star$, i.e., that the vector of predictions in the column space of $\mathbf{X}^\top$, but this was for the case without regularizer.

2. The original formulation involves computation of order $O(D^3 + ND^2)$, while the second can be computed in time $O(N^3 + DN^2)$. Hence it depends on the size of D versus N, which of the two is more efficient.

The representer theorem

The representer theorem generalizes this result: for a $\mathbf{w}^\star$ minimizing the following function for any $\mathcal{L}_n$,

$$\min_{\mathbf{w}} \sum_{n=1}^{N} \mathcal{L}_n(\mathbf{x}_n^\top \mathbf{w}, y_n) + \tfrac{\lambda}{2}\|\mathbf{w}\|^2$$

there exists $\boldsymbol{\alpha}^\star$ such that $\mathbf{w}^\star = \mathbf{X}^\top \boldsymbol{\alpha}^\star$.

Such a general statement was originally proved by *Schölkopf, 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:

$$\mathbf{w}^\star = \arg\min_{\mathbf{w}} \quad \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \tfrac{\lambda}{2}\|\mathbf{w}\|^2$$

$$\boldsymbol{\alpha}^\star = \arg\max_{\boldsymbol{\alpha}} \; -\frac{1}{2}\boldsymbol{\alpha}^\top(\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I}_N)\boldsymbol{\alpha} + \lambda \boldsymbol{\alpha}^\top \mathbf{y}.$$

They are equivalent in the sense that $\mathbf{w}^* = \mathbf{X}^\top \boldsymbol{\alpha}^*$.
As we discussed previously, depending on the D, the dimension of the feature space, and N, the number of samples, one or the other of the two formulations might be more efficeint. Further, and perhaps most importantly, the second problem is expressed in terms of the kernel matrix $\mathbf{K} = \mathbf{X}\mathbf{X}^\top$.

Kernel functions

Recall that the kernel is defined as

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

For reasons that will become clear shortly, we call this the *linear* kernel.

Assume that we had first augmented the feature space to $\boldsymbol{\phi}(\mathbf{x})$. The associated kernel with basis functions $\boldsymbol{\phi}(\mathbf{x})$ would then be $\mathbf{K} := \boldsymbol{\Phi}^\top \boldsymbol{\Phi}$, where $\mathbf{K}$ is given as

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

We have already discussed that sometimes it is useful to augment the feature space. This will lead to a more powerful model. Here is a link to a video explaining this point in more detail: https://www.youtube.com/watch?v=3liCbRZPrZA

The kernel trick

The big advantage of using kernels is that rather than first augmenting the feature space and then computing the kernel, we can do both steps together, and we can do it more efficiently. Let us discuss how this works.

Let us define a "kernel function" $\kappa(\mathbf{x}, \mathbf{x}')$ and let us compute the (i, j)-th entry of $\mathbf{K}$ as $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. By the miracle of math it turns out that (for the right choice of kernel map κ) this is equivalent to first augmenting the features to $\boldsymbol{\phi}(\mathbf{x})$ and then computing the standard inner product

$$\kappa(\mathbf{x}, \mathbf{x}') := \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}') \ .$$

This is probably best seen by looking at examples:

1. To start trivially, if we pick the linear kernel $\kappa(\mathbf{x}, \mathbf{x}') := \mathbf{x}^\top \mathbf{x}'$, then the feature map corresponds just to the original features, $\boldsymbol{\phi}(\mathbf{x}') = \mathbf{x}'$.

2. Assume that $\mathbf{x} \in \mathbb{R}$, i.e., $\mathbf{x}$ is a scalar. The kernel $\kappa(x, x') := (xx')^2$ corresponds to $\phi(x) = x^2$.

3. Assume that $\mathbf{x} \in \mathbb{R}^3$, i.e., $\mathbf{x}$ is a vector of dimension 3. The kernel $\kappa(\mathbf{x}, \mathbf{x}') := (x_1 x_1' + x_2 x_2' + x_3 x_3')^2$ corresponds to

   $$\boldsymbol{\phi}(\mathbf{x})^\top = \left[\ x_1^2, \ x_2^2, \ x_3^2, \ \sqrt{2} x_1 x_2, \ \sqrt{2} x_1 x_3, \ \sqrt{2} x_2 x_3 \ \right] .$$

   This is an example of what is called a *polynomial kernel*.

4. The kernel

   $$\kappa(\mathbf{x}, \mathbf{x}') = \exp\left[-\frac{1}{2} (\mathbf{x} - \mathbf{x}')^\top (\mathbf{x} - \mathbf{x}') \right]$$

   corresponds to an infinite feature map! It is called the the *Radial Basis Function* (RBF) kernel. In order to look at this more in detail, consider the simple case

where the $\mathbf{x}$ and $\mathbf{d}'$ are scalars. In this case we have the expansion

$$\mathbf{K}(x, x') = e^{-(x)^2} e^{-(x')^2} \sum_{k=0}^{\infty} \frac{2^k (x)^k (x')^k}{k!}.$$

We see that we can think of this as the inner product of infinite-dimensional vectors whose k-th component, $k = 0, 1, \cdots$ is equal to

$$e^{-(x)^2} \sqrt{\frac{2^k}{k!}} (x)^k \text{ and } e^{-(x')^2} \sqrt{\frac{2^k}{k!}} (x')^k,$$

respectively. And although this is not obvious, let us state that this kernel cannot be represented as an inner product in a finite-dimensional space.

See more examples in Section 14.2 of Murphy's book.

Properties of kernels: Mercer's Condition

A natural question is the following: how can we ensure that there exists a $\boldsymbol{\phi}$ corresponding to a given kernel $\mathbf{K}$?
A kernel function must be an inner-product in some feature space. Mercer's condition statest that this is true if and only if the following two conditions are fulfilled:

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

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