annotated version

Machine Learning Course - CS-433

Kernel Ridge Regression

Nov 3, 2016

changes by Martin Jaggi 2016

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Motivation

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

$X^T \in \mathbb{R}^{N \times D}$

$X \in \mathbb{R}^{D \times N}$

Ridge regression

Recall the ridge regression problem

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

$N << D$

$N >> D$

For its solution, we have that

$$\mathbf{w}^\star = (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I}_D)^{-1}\mathbf{X}\mathbf{y} \quad \text{(A)}$$
$$= \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda \mathbf{I}_N)^{-1}\mathbf{y} =: \mathbf{X}\boldsymbol{\alpha}^\star, \quad \text{(B)}$$

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

Lemma

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

D

$(X^T X + I_N)^{-1} X^T = X^T (X X^T + I_D)^{-1}$

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

Proof

$\Leftrightarrow P(QP + I_M) = (PQ + I_N)\, P$

$P I_M = I_N P$ ∎

ASSUMPTION:

$PQ + 1$
$QP + 1$ are invertible

$X : D \times N$

XX^T

What are the computational complexities for the above two ways of computing $\mathbf{w}^\star$?

cost:

(A) $O(D^3 + ND^2)$

(B) $O(N^3 + DN^2)$

With this, we know that $\overset{\mathbb{R}^D}{\overset{\in}{\mathbf{w}^\star}} = \mathbf{X}\overset{\mathbb{R}^N}{\overset{\in}{\underline{\boldsymbol{\alpha}}^\star}}$ lies in the column space of $\mathbf{X}$,

$$\text{where } \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{D1} & x_{D2} & \dots & x_{DN} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \end{bmatrix}$$

The representer theorem

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

$\lambda > 0$

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

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

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:

$w^* = X\alpha^*$

$$\mathbb{R}^D \ni \mathbf{w}^\star = \arg\min_{\mathbf{w}} \; \frac{1}{2}\|\mathbf{y} - \mathbf{X}^\top \mathbf{w}\|^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

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

i.e. they both have the same optimal value. Also, we can always have the correspondence mapping $\mathbf{w} = \mathbf{X}\boldsymbol{\alpha}$.

$X^T X \in \mathbb{R}^{N \times N}$

Most importantly, the second problem is expressed in terms of the matrix $\mathbf{X}^\top\mathbf{X}$. 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. You can find a derivation of this duality 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.

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

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

Kernel functions

The linear kernel is defined below:

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

Kernel with basis functions $\boldsymbol{\phi}(\mathbf{x})$ with $\mathbf{K} := \boldsymbol{\Phi}^\top \boldsymbol{\Phi}$ is shown below:

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

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 $\kappa(\mathbf{x}, \mathbf{x}')$ and compute the (i, j)-th entry of $\mathbf{K}$ as $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. A kernel function κ is usually associated with a feature map $\boldsymbol{\phi}$, such that

x_i training data

$\underline{K} = (\kappa(x_i, x_j))_{ij} \in \mathbb{R}^{N \times N}$

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

N^2 evaluations of $\kappa(\cdot,\cdot)$

For example, for the linear kernel $\kappa(\mathbf{x}, \mathbf{x}') := \mathbf{x}^\top \mathbf{x}',$, the feature map is just the original features, $\boldsymbol{\phi}(\mathbf{x}') = \mathbf{x}'$.

Another example: The kernel $\kappa(x, x') := x^2 (x')^2$ corresponds to $\phi(x) = x^2$, and $\kappa(\mathbf{x}, \mathbf{x}') := (x_1 x_1' + x_2 x_2' + x_3 x_3')^2$ corresponds to

$= \phi(x)^\top \phi(x')$

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

The good news is that the evaluation of a kernel is often faster when using κ instead of $\boldsymbol{\phi}$.

Visualization

Why would we want such general feature maps?

See video explaining linear separation in the kernel space (where $\phi(\mathbf{x})$ maps to) corresponding to non-linear separation in the original $\mathbf{x}$-space: https://www.youtube.com/watch?v=3liCbRZPrZA

Examples of kernels

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

Poly: $K(x,x') = (x^T x')^5 = \phi(x)^T \phi(x')$

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

RBF: $K(x,x') = r(\|x-x'\|^2)$

See more examples in Section 14.2 of Murphy's 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 the kernel satisfies 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. $\kappa(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x}', \mathbf{x})$.
2. For any arbitrary input set $\{\mathbf{x}_n\}$ and all N, $\mathbf{K}$ should be positive semi-definite.

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

$x \in \mathbb{R}^D$

$\Phi(x) \in \mathbb{R}^\infty$

ToDo

1. Understand the relationship $\mathbf{w}^\star = \mathbf{X}\boldsymbol{\alpha}^\star$. Understand the statement of the representer theorem.
2. Show that the primal and dual formulations of ridge regression are equivalent. Hint: show that the optimization problems corresponding to $\mathbf{w}$ 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 Murphy's book for examples of kernels.
4. Revise and understand the difference between positive-definite and positive-semi-definite matrices.
5. If you are interested in more details about kernels, read about Mercer and Matern kernels from Kevin Murphy's Section 14.2. There is also a small note by Matthias Seeger on the git repository under lectures/07, in particular for the case of infinite dimensional $\boldsymbol{\phi}$.