Machine Learning
Exercise 6

Marc Toussaint
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

May 11, 2016

1 Kernel Ridge regression

In exercise 2 we implemented Ridge regression. Modify the code to implement Kernel ridge regression. Try to program it in a way that you only need to change one line to have a different kernel. Note that this computes optimal "parameters" $\alpha = (K + \lambda I)^{-1} y$ such that $f(x) = \kappa(x)^\top \alpha$.

a) Using a linear kernel, does this reproduce the linear regression we looked at in exercise 2? Test this on the data. If not, how can you make it equivalent?

b) Is using the squared exponential kernel $k(x, x') = \exp(-\gamma \,|\, x - x' \,|\,^2)$ exactly equivalent to using the radial basis function features we introduced?

2 Positive Definite Kernels (optional/bonus)

For a non-empty set X, a kernel is a symmetric function $k : X \times X \to \mathbb{R}$. Note that the set X can be arbitrarily structured (real vector space, graphs, images, strings and so on). A very important class of useful kernels for machine learning are positive definite kernels. A kernel is called *positive definite*, if for all arbitrary finite subsets $\{x_i\}_{i=1}^n \subseteq X$ the corresponding *kernel matrix* K with elements $K_{ij} = k(x_i, x_j)$ is positive *semi*-definite,

$$\alpha \in \mathbb{R}^n \;\Rightarrow\; \alpha^\top K \alpha \geq 0. \tag{1}$$

For features $\phi : \; X \to H$ (with H a suitable space), prove that

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_H \tag{2}$$

is a positive definite kernel.

3 Kernel Construction (optional/bonus)

Often, one wants to construct more complicated kernels out of existing ones. Let $k_1, k_2 : X \times X \to \mathbb{R}$ be two positive definite kernels. Proof that

1. $k(x, x') = k_1(x, x') + k_2(x, x')$
2. $k(x, x') = c \cdot k_1(x, x') \quad \text{for} \quad c \geq 0$
3. $k(x, x') = k_1(x, x') \cdot k_2(x, x')$
4. $k(x, x') = k_1(f(x), f(x')) \quad \text{for} \quad f : X \to X$

are positive definite kernels.

4 Kernel logistic regression (no implementation to do)

The "kernel trick" is generally applicable whenever the "solution" (which may be the predictive function $f^{\text{ridge}}(x)$, or the discriminative function, or principal components...) can be written in a form that only uses the kernel function $k(x, x')$, but never features $\phi(x)$ or parameters β explicitly.

Derive a kernelization of Logistic Regression. That is, think about how you could perform the Newton iterations based only on the kernel function $k(x, x')$.

Tips: Reformulate the Newton iterations

$$\beta \leftarrow \beta - (\boldsymbol{X}^\top W \boldsymbol{X} + 2\lambda I)^{-1} \; [\boldsymbol{X}^\top(\boldsymbol{p} - \boldsymbol{y}) + 2\lambda I \beta] \tag{3}$$

using the two Woodbury identities

$$(X^\top W X + A)^{-1} X^\top W = A^{-1} X^\top (X A^{-1} X^\top + W^{-1})^{-1} \tag{4}$$

$$(X^\top W X + A)^{-1} = A^{-1} - A^{-1} X^\top (X A^{-1} X^\top + W^{-1})^{-1} X A^{-1} \tag{5}$$

Note that you'll need to handle the $\boldsymbol{X}^\top(\boldsymbol{p} - \boldsymbol{y})$ and $2\lambda I \beta$ differently.

Then think about what is actually been iterated in the kernalized case: surely we cannot iteratively update the optimal parameters, because we want to rewrite equations to never touch β or $\phi(x)$ explicitly.