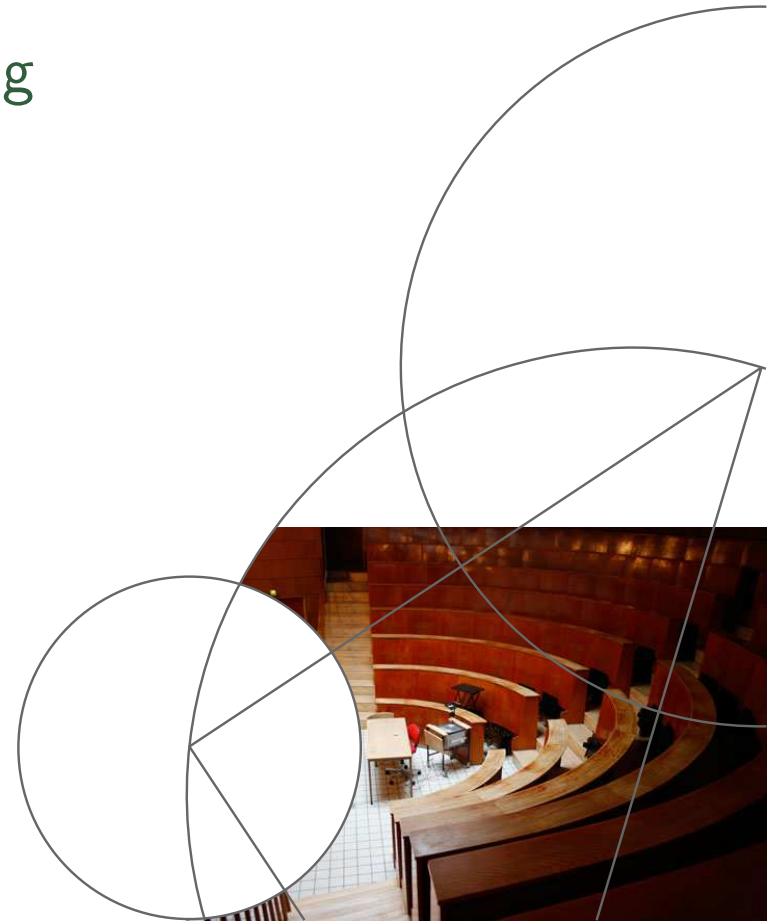
Basic Kernel Methods

Statistical Methods for Machine Learning

Christian Igel
Department of Computer Science



Outline

Outline

Perceptron learning algorithm

Algorithm 1: Kernel perceptron

Input: data $\{(x_1, y_1), \dots\} \subseteq (\mathcal{X} \times \{-1, 1\})^\ell$, kernel k

Output: hypothesis $h(x) = \operatorname{sgn}\left(\sum_{i=1}^{\ell} \alpha_i y_i k(x_i, x)\right)$

$\boldsymbol{\alpha} \leftarrow 0$
repeat
  for $i = 1, \dots, \ell$ **do**
    if $y_i \sum_{j=1}^{\ell} \alpha_j y_j k(x_j, x_i) \leq 0$ **then**
      $\alpha_i \leftarrow \alpha_i + 1$
until *no mistake made within* **for** *loop*

Outline

κ-nearest neighbor (κ-NN)

Algorithm 2: κ-nearest neighbor

Input: kernel k, $\kappa \in \mathbb{N}^+$, data $\{(x_1, y_1), \dots\} \subseteq (\mathcal{X} \times \{-1, 1\})^\ell$, new input x to be classified

Output: predicted label y of x

$S = \{(x_1, y_1), \dots\}$

$S_\kappa = \emptyset$

while $|S_\kappa| < \kappa$ **do**

  $S' \leftarrow \left\{ \operatorname{argmin}_{(x_i, y_i) \in S} \sqrt{k(x, x) - 2k(x, x_j) + k(x_j, x_j)} \right\}$

  $S_\kappa \leftarrow S_\kappa \cup S'$

  $S \leftarrow S \setminus S'$

Result: $y = \operatorname{sgn}\left(\frac{1}{|S_\kappa|} \sum_{(x_i, y_i) \in S_\kappa} y_i \right)$

Outline

Representer theorem

Let $\Omega : [0, \infty[\to \mathbb{R}$ be a strictly monotonic increasing function, $\mathcal{H}$ a RKHS with kernel k on $\mathcal{X}$ and L a loss function. Given $S = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\} \subset (\mathcal{X} \times \mathbb{R})^\ell$, each minimizer $f \in \mathcal{H}^b$ of the regularized empirical risk

$$\sum_{i=1}^{\ell} L(y_i, f(x_i)) + \Omega(\|f\|_k^2)$$

admits a representation of the form

$$f(x) = \sum_{i=1}^{\ell} \beta_i k(x_i, x) + b$$

with $\alpha_1, \ldots, \alpha_\ell, b \in \mathbb{R}$.

Proof of representer theorem

Projecting candidate solution onto span of training patterns

$$f(x) = f_{\|}(x) + f_{\perp}(x) + b = \sum_{i=1}^{\ell} \alpha_i k(x_i, x) + f_{\perp}(x) + b$$

$$\forall j \in \{1, \ldots, \ell\} : f(x_j) = \langle f(\cdot), k(x_j, \cdot) \rangle + b$$

$$= \sum_{i=1}^{\ell} \alpha_i k(x_i, x_j) + \langle f_{\perp}(\cdot), k(x_j, \cdot) \rangle + b = \sum_{i=1}^{\ell} \alpha_i k(x_i, x_j) + b$$

$$\Omega\left(\left\|\sum_{i=1}^{\ell} \alpha_i k(x_i, .)\right\|_k^2 + \|f_{\perp}\|_k^2\right) \geq \Omega\left(\left\|\sum_{i=1}^{\ell} \alpha_i k(x_i, .)\right\|_k^2\right)$$

Outline

Regularization networks I

The squared loss function gives an empirical risk

$$\frac{1}{\ell}\sum_{i=1}^{\ell}(y_i - f(x_i))^2 \ .$$

Applying Tikhonov regularization leads to regularized riks

$$\frac{1}{\ell}\sum_{i=1}^{\ell}(y_i - f(x_i))^2 + \gamma\|f\|^2$$

for $f \in \mathcal{H}$; we know there is a solution of the form

$$f(x) = \sum_{i=1}^{\ell}\alpha_i k(x_i, x) \ .$$

Regularization networks II

We have $\partial f(x)/\partial \alpha_i = k(x_i, x)$. Setting functional derivative of regularized loss to zero yields for all $i = 1, \ldots, \ell$:

$$\frac{2}{\ell} \sum_{j=1}^{\ell} (y_j - f(x_j)) k(x_i, x_j) - 2\gamma \langle f, k(x_i, \cdot) \rangle = 0$$

$$\sum_{j=1}^{\ell} (y_j - f(x_j)) k(x_i, x_j) - \ell\gamma f(x_i) = 0$$

$$\sum_{j=1}^{\ell} \left[y_j - \sum_{m=1}^{\ell} \alpha_m k(x_m, x_j) \right] k(x_i, x_j) - \ell\gamma \sum_{l=1}^{\ell} \alpha_l k(x_l, x_i) = 0$$

$$\sum_{j=1}^{\ell} \left[y_j - \sum_{m=1}^{\ell} \alpha_m k(x_m, x_j) - \ell\gamma\alpha_j \right] k(x_i, x_j) = 0$$

Regularization networks III

$$\sum_{j=1}^{\ell} \left[y_j - \sum_{m=1}^{\ell} \alpha_m k(x_m, x_j) - \ell\gamma\alpha_j \right] k(x_i, x_j) = 0$$

for all i is fulfilled if for all j

$$y_j - \sum_{m=1}^{\ell} \alpha_m k(x_m, x_j) - \ell\gamma\alpha_j = 0$$

(which is necessary if k is strictly positive definite)

In matrix form we have

$$\boldsymbol{y} - (\ell\gamma\boldsymbol{I} + \boldsymbol{K})\boldsymbol{\alpha} = \boldsymbol{0}$$

$\rightarrow$ Algorithm "almost magical for its simplicity and effectiveness" (Poggio & Smale, 2003)

Regularization networks IV

Algorithm 3: Regularization network

Input: kernel k, regularization parameter $\gamma \in \mathbb{R}^+$, data
$\{(x_1, y_1), \dots\} \subseteq (\mathcal{X} \times \mathbb{R})^\ell$

Output: hypothesis $h(x) = \sum_{i=1}^{\ell} \alpha_i k(x_i, x)$

1 $\boldsymbol{y} = (y_1, \dots, y_\ell)^\mathsf{T}$

2 $\boldsymbol{I} = \mathrm{diag}(1, \dots, 1) \in \mathbb{R}^{\ell \times \ell}$

3 $\boldsymbol{K} \in \mathbb{R}^{\ell \times \ell}, [\boldsymbol{K}]_{ij} = k(x_i, x_j)$

4 $\boldsymbol{\alpha} \leftarrow (\ell\gamma \boldsymbol{I} + \boldsymbol{K})^{-1} \boldsymbol{y}$

Summary

- Kernel trick leads to many simple, but effective algorithms
- Regularization networks algorithm is key learning method
- Minimizer of the regularized loss lies in the span of the kernels centered on the training points

References:

B. Schölkopf and A. J. Smola, Learning with Kernels, MIT Press, 2002.

T. Poggio and S. Smale, The mathematics of learning: Dealing with data. Notices of the American Mathematical Society, 50(5):537–544, 2003