Kernel-Based Learning & Multivariate Modeling

MIRI Master - DMKM Master

Lluís A. Belanche

belanche@cs.upc.edu

Soft Computing Research Group

Universitat Politècnica de Catalunya

2016-2017

Kernel-Based Learning & Multivariate Modeling

Contents by lecture

Sep 14 Introduction to Kernel-Based Learning
Sep 21 The SVM for classification, regression & novelty detection (I)
Sep 28 The SVM for classification, regression & novelty detection (II)
Oct 05 Kernel design (I): theoretical issues
Oct 19 Kernel design (II): practical issues
Oct 26 Kernelizing ML & stats algorithms
Nov 02 **Advanced topics**

Small project

Information

60 % exercises (4 grades) + 40 % small project (groups of **two** people):

(T) 20 % - theoretical work (technical correctness)

(E) 20 % - experimental work (methodological correctness)

(V) 20 % - volume of work (*e.g.*, own implementation?)

(O) 20 % - originality of work (replication? ideas?)

(C) 20 % - conclusions (insight, scope)

Small project

Possibilities

A) Apply an **standard kernel method** (SVM for CRND) to a **specific problem** of your interest; comparison to other approaches. The focus is on the **application**

B) Choose and apply an algorithm/technique that has already been **kernelized** (except the SVM), study it [program it], and apply it to one or more [benchmark] problem(s); comparison to the standard version. The focus is on the **technique**

C) Study a **non-standard kernel** (not for $\mathbb{R}^d$), and apply it to a **specific problem** of your interest, with one or more kernel method(s). The focus is on the **kernel function**

Advanced topics

Outline

1. An example of a Hilbert space

2. Relevance Vector Machines

3. The Representer theorem

4. Guidelines for the small project

Advanced topics

An example of a Hilbert space

Definition of a Hilbert space:

- A vector space endowed with an **inner product** whose associated norm defines a complete metric
- Distances, lenghts and angles are well-defined for the elements of the space
- Completeness means that all Cauchy sequences defined in $\mathcal{H}$ converge to an element of $\mathcal{H}$

Generalizes the notion of Euclidean space: an infinite dimensional space with the structure of $\mathbb{R}^N$

Advanced topics

An example of a Hilbert space

Example: the ℓ_2 space of square-summable sequences

$$\ell_2 := \left\{ (a_n)_{n=1}^{\infty}, a_n \in \mathbb{R}, \sum_{n=1}^{\infty} a_n^2 < \infty \right\}$$

- This is a vector space with inner product $\langle a, b \rangle := \sum_{n=1}^{\infty} a_n b_n$
- Completeness comes from the fact that $\mathbb{R}$ is complete

Relevance Vector Machines

Extended linear models

Extended linear models are a commonly used form of models for continuous regression scenarios:

$$t_n = f(x_n) + \epsilon_n, \qquad \epsilon_n \sim \mathcal{N}(0, \sigma^2)$$

where $D = \{(\boldsymbol{x}_1, t_1), \ldots, (\boldsymbol{x}_N, t_N)\}$ is an i.i.d. random sample for the probabilistic mechanism $p(t, \boldsymbol{x})$ of length N.

Relevance Vector Machines

These models take the form:

$$f(\boldsymbol{x}) = \sum_{j=0}^{M} w_j \phi_j(\boldsymbol{x}) = \boldsymbol{w}^{\mathsf{T}} \phi(\boldsymbol{x})$$

- there is a set of M **basis functions** (arbitrary real-valued functions) with $\phi_0(\cdot) = 1$
- $\boldsymbol{w}$ is the vector of regression **coefficients**

Relevance Vector Machines

Sparsity

A linear model is **sparse** if a significant number of its coefficients is very small or effectively zero

- These coefficients are the weights associated to the most relevant (training) data examples

- Intuitively, sparsity indicates the complexity of a model

Relevance Vector Machines

The **Relevance Vector Machine** (RVM) is a sparse Bayesian method for training linear models extended with BFs in which:

1. A prior on the weights is introduced (usually Gaussian)

2. The likelihood of the marginal probability is maximized

3. Sparse solutions are explicitly enforced

Relevance Vector Machines

Assuming an independent zero-mean homoscedastic Gaussian noise model of variance σ^2, the **likelihood** $\mathcal{L}$ of the parameters $(\boldsymbol{w}, \sigma^2)$ is:

$$\mathcal{L}(\boldsymbol{w}, \sigma^2) := p(\boldsymbol{t}|\boldsymbol{w}, \{\boldsymbol{x}_n\}, \sigma^2) = \prod_{n=1}^{N} p(t_n|\boldsymbol{w}, \boldsymbol{x}_n, \sigma^2)$$

$$= (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{t} - \Phi\boldsymbol{w}\|^2\right)$$

where $\Phi_{N\times(M+1)}$ is the design matrix of the inputs
(the n-th row of Φ represents the vector $\phi(\boldsymbol{x}_n)$)

Relevance Vector Machines

In a classical statistical approach,

- One minimizes "minus the logarithm" of the likelihood $-\ln \mathcal{L}$ to obtain an estimation $(\hat{\boldsymbol{w}}, \widehat{\sigma}^2)$ of $(\boldsymbol{w}, \sigma^2)$

- Which often leads to overfitting and thus regularization becomes necessary

This is because we are attempting to obtain *point estimates* about the optimal value of the model weight parameters

Relevance Vector Machines

- In the **Bayesian framework**, regularization terms to the likelihood are added, on the basis of some prior knowledge over weight distribution

- In the RVM, a zero-mean Gaussian **prior distribution** with independent variances (acting as **hyperparameters**) is defined over each weight:

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{j=1}^{M} \sqrt{\frac{\alpha_j}{2\pi}} \exp\left(-\frac{1}{2}\alpha_j {w_j}^2\right)$$

where $\alpha_j := 1/\sigma_{w_j}^2$

Relevance Vector Machines

- We could go on, defining **hyperprior distributions** on $\boldsymbol{\alpha}$ and σ^2 (initially proposed to be Gamma istributions)

- With the previous prior and likelihood distributions, the **posterior distribution** over all unknown parameters is:

$$p(\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2 | \boldsymbol{t}) = p(\boldsymbol{w} | \boldsymbol{t}, \boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{\alpha}, \sigma^2 | \boldsymbol{t})$$

which we wish to maximize w.r.t. the unknown parameters $(\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2)$

Relevance Vector Machines

The posterior over the weights $p(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{\alpha}, \sigma^2)$ is also a Gaussian that can be obtained using the **Bayes formula**:

$$p(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{\alpha}, \sigma^2) = \frac{p(\boldsymbol{t}|\boldsymbol{w}, \sigma^2)p(\boldsymbol{w}|\boldsymbol{\alpha})}{p(\boldsymbol{t}|\boldsymbol{w}, \sigma^2)} \sim \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \Sigma)$$

$$\boldsymbol{\mu} := \sigma^{-2}\Sigma\Phi^{\mathsf{T}}\boldsymbol{t}$$
$$\Sigma := (\sigma^{-2}\Phi^{\mathsf{T}}\Phi + A)^{-1}$$

being $A = diag(\alpha_1, \ldots, \alpha_M)$

Relevance Vector Machines

Moreover, we must find the hyperparameters $\boldsymbol{\alpha}, \sigma^2$ that maximize:

$$p(\boldsymbol{\alpha}, \sigma^2 | \boldsymbol{t}) = \frac{p(\boldsymbol{t}|\boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{\alpha}, \sigma^2)}{p(\boldsymbol{t})}$$

$p(\boldsymbol{\alpha}, \sigma^2) = p(\boldsymbol{\alpha}) p(\sigma^2)$ (they vanish because we assume them uniform)
$p(\boldsymbol{t})$ does not depend on any parameter (or hyperparameter)

$\rightarrow$ the maximization of $p(\boldsymbol{t}|\boldsymbol{\alpha}, \sigma^2)$ suffices

Relevance Vector Machines

The likelihood distribution can be calculated by integrating out the weights to obtain the **evidence** (marginal likelihood) for the hyperparameters:

$$p(\boldsymbol{t}|\boldsymbol{\alpha}, \sigma^2) = \int p(\boldsymbol{t}|\boldsymbol{w}, \sigma^2) p(\boldsymbol{w}|\boldsymbol{\alpha}) d\,\boldsymbol{w} =$$

$$(2\pi\sigma^2)^{-\frac{N}{2}} \exp(-E(\boldsymbol{t}))\sqrt{|\Sigma|} \prod_{j=1}^{M} \sqrt{\alpha_j} \sim \mathcal{N}(\boldsymbol{t}; \mathbf{0}, C),\ C = \sigma^2 I + \Phi A^{-1} \Phi^{\mathsf{T}}$$

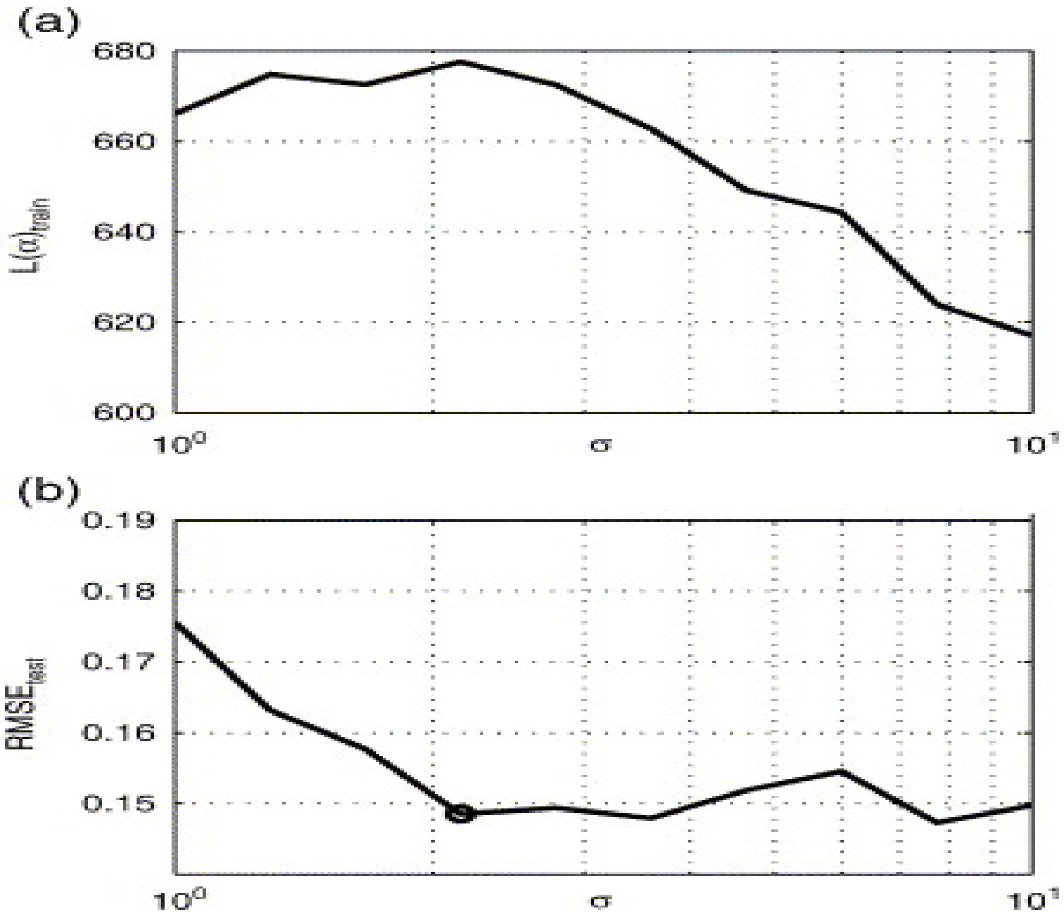
where $E(\boldsymbol{t}) := \frac{1}{2}(\sigma^{-2}||\boldsymbol{t}||^2 - \boldsymbol{\mu}^{\mathsf{T}} \Sigma^{-1} \boldsymbol{\mu})$

Relevance Vector Machines

For computational efficiency, the logarithm of the evidence is maximized:

$$\ln p(\boldsymbol{t}|\boldsymbol{\alpha}, \sigma^2) = \frac{1}{2}\left(-N\ln(\sigma) - 2E(\boldsymbol{t}) - \ln|\Sigma| - N\ln(2\pi) + \sum_{j=1}^{M}\ln(\alpha_j)\right)$$

Relevance Vector Machines



Example of maximizing the evidence, as a function of kernel parameter $\sigma^2_{\mathrm{R}BF}$; bottom plot is RMSE in the test set.

Relevance Vector Machines

$$\frac{\partial}{\partial \alpha_j} \ln p(\boldsymbol{t}|\boldsymbol{\alpha}, \sigma^2) = 0 \qquad \Rightarrow \qquad \alpha_j = \frac{\gamma_j}{\mu_j^2}$$

$$\frac{\partial}{\partial \sigma^2} \ln p(\boldsymbol{t}|\boldsymbol{\alpha}, \sigma^2) = 0 \qquad \Rightarrow \qquad \sigma^2 = \frac{||\boldsymbol{t} - \Phi \boldsymbol{\mu}||^2}{N - \sum_j \gamma_j}$$

where $\gamma_j = 1 - \alpha_j \Sigma_{jj}$.

Relevance Vector Machines

The procedure is then:

1. Estimate the hyperparameters $\boldsymbol{\alpha}$ and σ^2 by maximizing the log-marginal likelihood $\ln p(\boldsymbol{t}|\boldsymbol{\alpha}, \sigma^2)$

2. Estimate the weight distribution as $p(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{\alpha}, \sigma^2) \sim \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \Sigma)$ (that is, estimate $\boldsymbol{\mu}, \Sigma$)

→ these two steps are iterated (using an EM-like procedure)

—see `http://www.miketipping.com/sparsebayes.htm`

Relevance Vector Machines

- Once the process has converged, the optimum weights are given by their maximum a posteriori (MAP) estimate, which is the mean of their (posterior) distribution

- Sparsity is achieved because in practice many of the hyperparameters α_j tend to infinity, yielding a posterior distribution of the weight w_j that is sharply peaked around zero

- These weights can then be deleted from the model, as well as their associated basis functions $\phi(\boldsymbol{x}_j)$; the remaining data points are called the **relevance vectors**, resembling the SVs in the SVMR framework

Relevance Vector Machines

Φ is the Gram matrix (or kernel matrix) of the inputs: we can select a suitable kernel function and use it to create Φ

1. **Choose** starting values for $\boldsymbol{\alpha}, \sigma^2$

2. **Update** $\boldsymbol{\mu}, \Sigma$

3. **Update** $\boldsymbol{\alpha}, \sigma^2$

4. **Iterate** steps 2 and 3 until **convergence** (typically $\boldsymbol{\alpha}$ stabilizes)

Relevance Vector Machines

Once we have the marginal likelihood, we can build a **predictive distribution** over t to predict a new data point $\boldsymbol{x}^*$:

$$p(t|\boldsymbol{x}^*, \boldsymbol{\alpha}, \sigma^2) = \int p(t|\boldsymbol{w}, \sigma^2) p(\boldsymbol{w}|\boldsymbol{\alpha}, \sigma^2) d\,\boldsymbol{w} \sim \mathcal{N}(f(\boldsymbol{x}^*), \sigma^2(\boldsymbol{x}^*))$$

In words, the prediction is $f(\boldsymbol{x}^*) = \boldsymbol{\mu}^\mathsf{T} \phi(\boldsymbol{x}^*)$ with uncertainty $\sigma^2(\boldsymbol{x}^*) = \sigma^2 + \phi(\boldsymbol{x}^*)^\mathsf{T} \Sigma \phi(\boldsymbol{x}^*)$.

Notice that $\phi(\boldsymbol{x}^*) = (\phi_n(\boldsymbol{x}^*))_{n \in RV}$, where $\phi_n(\boldsymbol{x}^*) = k(\boldsymbol{x}_n, \boldsymbol{x}^*)$.

Relevance Vector Machines

Compared to the **SVM for regression** (SVMR):

1. Both have the same functional form
2. RVMs typically leads to sparser models
3. RVMs are less sensitive to hyperparameter setting
4. RVMs are able to yield posterior distributions (instead of point estimates)

Relevance Vector Machines

In addition, compared to **neural networks** (ANNs):

1. In ANNs training often results in a complex, time-consuming task
2. ANNs are easy to overfit, leading to poor generalization
3. ANNs are in trouble dealing with low-sized data sets

Kernel design (I): theoretical issues

The Representer theorem

Regularization provides a way of interpreting SVMs in the context of other machine learning algorithms: choosing a **fitting function** (model) that finds a balance between:

1. low training error

2. is not “too complex”

For example, regularized least squares is a special case of regularization using the squared error loss

SVM are another special case of regularization, with the hinge loss

Kernel design (I): theoretical issues

The Representer theorem

Consider $L : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ a convex loss function, a data set $D = \{(\boldsymbol{x}_1, t_1), \ldots, (\boldsymbol{x}_N, t_N)\}$, with $\boldsymbol{x}_n \in \mathcal{X}, t_n \in \mathbb{R}$, and $\mathcal{H}$ a RKHS of functions $y : \mathcal{X} \to \mathbb{R}$ with reproducing kernel k. Then, for all $\lambda > 0$,

1. There exists a unique solution $y_{D,\lambda}$ to the problem:

$$y_{D,\lambda} := \operatorname*{arg\,min}_{y \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} L(t_n, y(\boldsymbol{x}_n)) + \lambda \|y\|_{\mathcal{H}}^2$$

2. There exist $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$ such that

$$y_{D,\lambda}(\boldsymbol{x}) = \sum_{n=1}^{N} \alpha_n k(\boldsymbol{x}_n, \boldsymbol{x}), \ \forall \boldsymbol{x} \in \mathcal{X}$$

Kernel design (I): theoretical issues

The Representer theorem

$$
\begin{aligned}
\|y\|_{\mathcal{H}}^2 &= \langle y, y \rangle_{\mathcal{H}} \\
&= \left\langle \sum_{n=1}^{N} \alpha_n k(\boldsymbol{x}_n, \cdot), \sum_{n=1}^{N} \alpha_n k(\boldsymbol{x}_n, \cdot) \right\rangle_{\mathcal{H}} \\
&= \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) \\
&= \boldsymbol{\alpha}^{\mathsf{T}} K \boldsymbol{\alpha} \geq 0
\end{aligned}
$$

where $K = (k_{nm})$, and $k_{nm} = k(\boldsymbol{x}_n, \boldsymbol{x}_m)$.

Kernel design (I): theoretical issues

Application to SVMs

Choose the **hinge loss** (the tightest convex upper bound to 0/1 error)

$$L(t, g(\boldsymbol{x})) = \text{máx}(1 - t\, g(\boldsymbol{x}), 0), \qquad g(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b$$

The optimization problem is:

$$\arg\min_{y \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \text{máx}(1 - t_n\, g(\boldsymbol{x}_n), 0) + \lambda \|y\|_{\mathcal{H}}^2$$

Setting $C = \frac{1}{2N\lambda}$ and rearranging we obtain the **primal SVM** problem:

$$\arg\min_{y \in \mathcal{H}} \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{n=1}^{N} \text{máx}(1 - t_n\, g(\boldsymbol{x}_n), 0)$$

where in this case $\|y\|_{\mathcal{H}}^2 = \boldsymbol{\alpha}^{\mathsf{T}} K \boldsymbol{\alpha} = \|\boldsymbol{w}\|^2$.

Kernel design (I): theoretical issues

Afterthoughts

- Regularization aims to fit data while explicitly controlling the fit (to avoid under/overfitting). To do this we choose a fitting function that has a balance between low error on the training set and low complexity

- Complex functions are functions with high (square) norms in some function space given by a feature map

- SVMs use RKHSs: Hilbert spaces of functions with reproducing kernel k. The kernel k can be used to describe all functions in the space

- The optimal solution is one function in this space and can be expressed as a finite combination of (at most) N kernel evaluations. The coefficients of this combination are data dependent