Relevance Vector Machines

Dmitriy Shutin

dshutin@inw.tugraz.at.

Signal Processing and Speech Communication Laboratory

spsc.inw.tugraz.at

Institute of Communications and Wave Propagation

Graz University of Technology

TUG

Outline

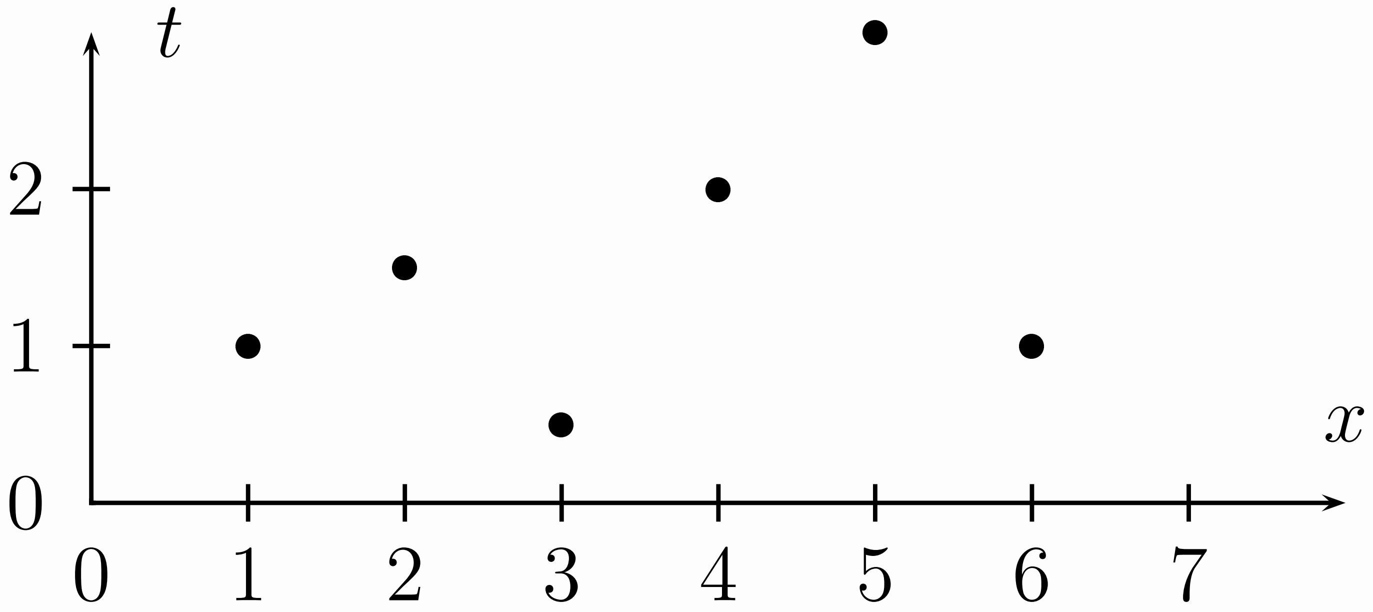
- Machine Learning as function approximation.
  - Kernel methods.
  - Sparsity and SVM.
- Relevance Vector Machines.
  - Problem formulation (regression).
  - Priors.
  - Inference.
- RVM example.
- Conclusions.

Machine learning

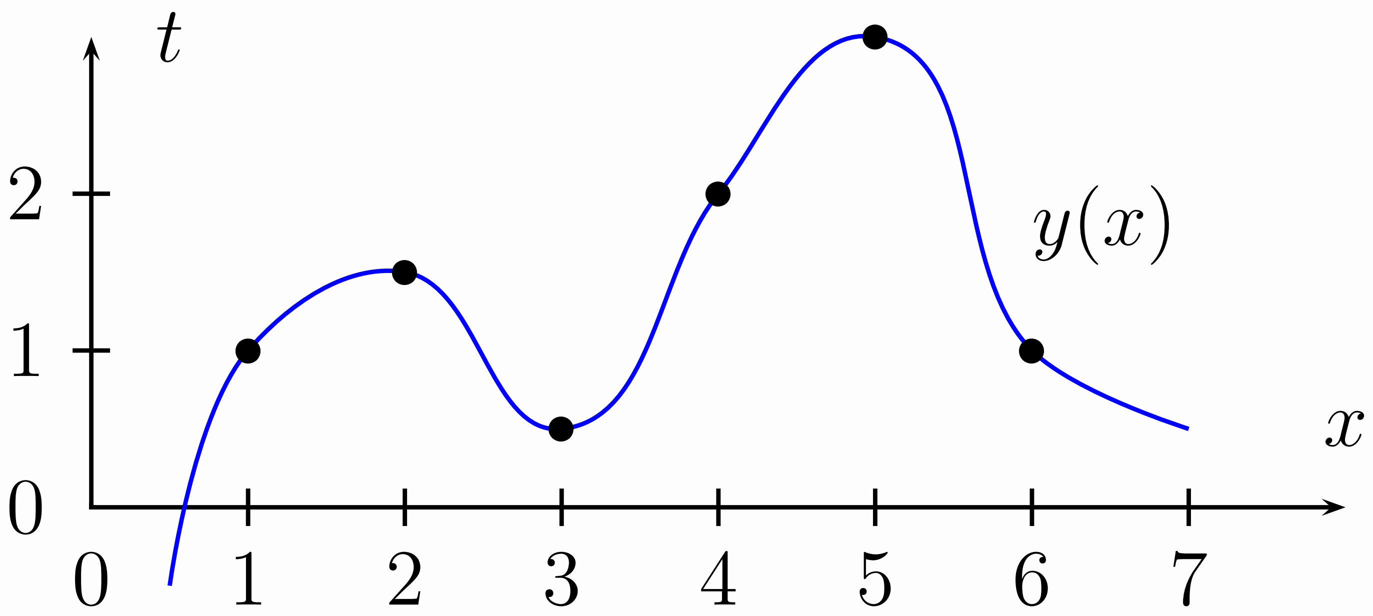
The goal of **supervised machine learning** is to use a training set S_x to “learn” a function $y(\cdot)$ that correctly “explains” observations/targets $\boldsymbol{t}$ given input data $\boldsymbol{x}$, i.e.,

$$\boldsymbol{t} = y(\boldsymbol{x}),\ \boldsymbol{x} \in S_x$$

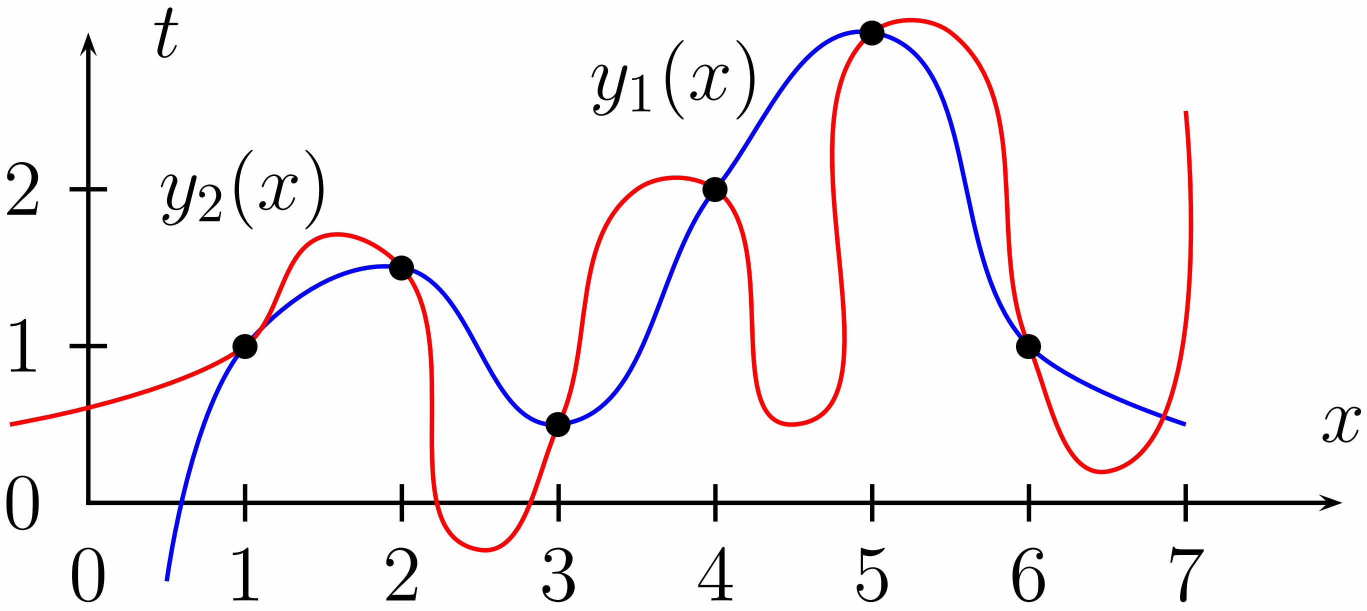
Consider the following example:
given a set of points (x_i, t_i) find the function $y(x)$, such that $t_i = y(x_i)$



This is one possible solution



Here is another, and quite a different one. Which solution to choose?



Kernel methods

One possible solution is the following:

- Choose a certain hypothesis space $\mathcal{H}$, $y \in \mathcal{H}$.
- Impose constrains on the function y. In most cases it is the norm of the sought function y.
- Find the regularized solution.

$$\min_{y \in \mathcal{H}} \left\{ \frac{1}{N} \sum_{i=1}^{N} V(\boldsymbol{t}_i, y(\boldsymbol{x}_i)) + \lambda \|y\|^2 \right\}$$

Kernel methods

It can be shown that the solution to this regularization could be written in the following form

$$y(\boldsymbol{x}) = \sum_{i=1}^{N} w_i \cdot K(\boldsymbol{x}, \boldsymbol{x_i})$$

$K(\cdot,\cdot)$ is called the kernel.

Usually the kernel $K(\cdot,\cdot)$ is associated with the corresponding hypothesis space $\mathcal{H}$.

Sparsity

With as many parameters as training examples, we would expect severe over-fitting. By setting some of the weights to zero this can be avoided. Thus, the model becomes *sparse*.

The direct posterior of such an approach leads to Support Vector Machines (SVMs).

In the SVM case, every $\boldsymbol{x}_i$ for which $w_i \neq 0$ becomes a support vector.

Relevance Vector Machines

Relevance Vector Machines

In a nutshell, RVM is a Bayesian approach to estimate the parameters w_i of the model

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{N} w_i \cdot K(\boldsymbol{x}, \boldsymbol{x}_i) + w_0$$

and introduce sparsity.

Relevance Vector Machines

- RVM is **not** a Bayesian interpretation of SVM but rather the method on its own, which adopts the same functional form.
- The kernel functions in RVM are treated simply as a set of basis functions without many restrictions imposed on SVM kernels.
- RVM uses a fully probabilistic framework.
- RVM uses significantly fewer basis functions then SVM.

Problem formulation

$\{\boldsymbol{x}_n, t_n\}_{n=1}^{N}$ is a training data set.

The targets are samples from the model with additive noise

$$t_n = y(\boldsymbol{x}_n; \boldsymbol{w}) + \epsilon_n$$

where

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{N} w_i \cdot K(\boldsymbol{x}, \boldsymbol{x}_i) + w_0$$

Problem formulation

ϵ_n is assumed to be zero-mean Gaussian noise process with variance σ^2.
Thus,

$$p(t_n|\boldsymbol{x}) = \mathcal{N}(t_n|y(\boldsymbol{x}_n), \sigma^2)$$

We rewrite the kernel sum in the following form:

$$\sum_{i=1}^{N} w_i \cdot K(\boldsymbol{x}, \boldsymbol{x}_i) + w_0 = \sum_{i=0}^{N} w_i \cdot \phi_i(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$$

where $\phi_0(\boldsymbol{x}) \equiv 1$

Problem formulation

The likelihood of the complete data set can be written as follows:

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

where
$\boldsymbol{t} = [t_1, t_2, \ldots, t_N]^T$, $N \times 1$ vector;
$\boldsymbol{w} = [w_0, w_2, \ldots, w_N]^T$; $(N+1) \times 1$ vector;
$\boldsymbol{\Phi} = [\boldsymbol{\phi}(\boldsymbol{x}_1), \boldsymbol{\phi}(\boldsymbol{x}_2), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_N)]^T$, $N \times (N+1)$ matrix.

Priors

To avoid over-fitting, we “constrain” the parameters by defining an explicit prior over them.

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=0}^{N} \mathcal{N}(w_i|0, \alpha_i^{-1})$$

with α being a vector of $(N+1)$ hyperparameters.

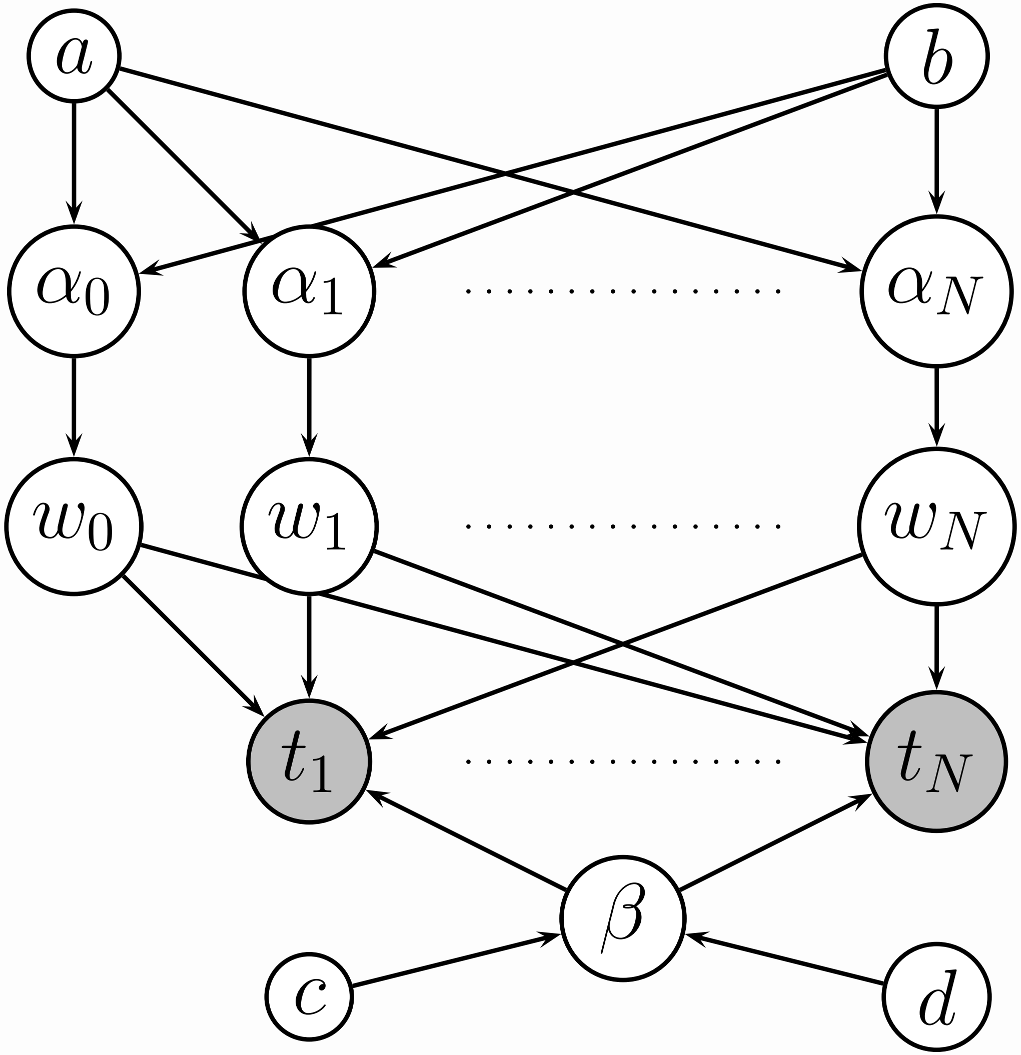
Priors

To complete the specification of priors, we define a hyperprior over α as well as over the noise variance σ^2.

$$p(\boldsymbol{\alpha}) = \prod_{i=0}^{N} \mathrm{Gamma}(\alpha_i|a,b)$$

$$p(\beta) = \mathrm{Gamma}(\beta|c,d), \text{ where } \beta \equiv \sigma^{-2}$$

Priors - graphical model



Learning RVM

Inference

How it should work: for the given test point x_* we should correctly predict the target t_*

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

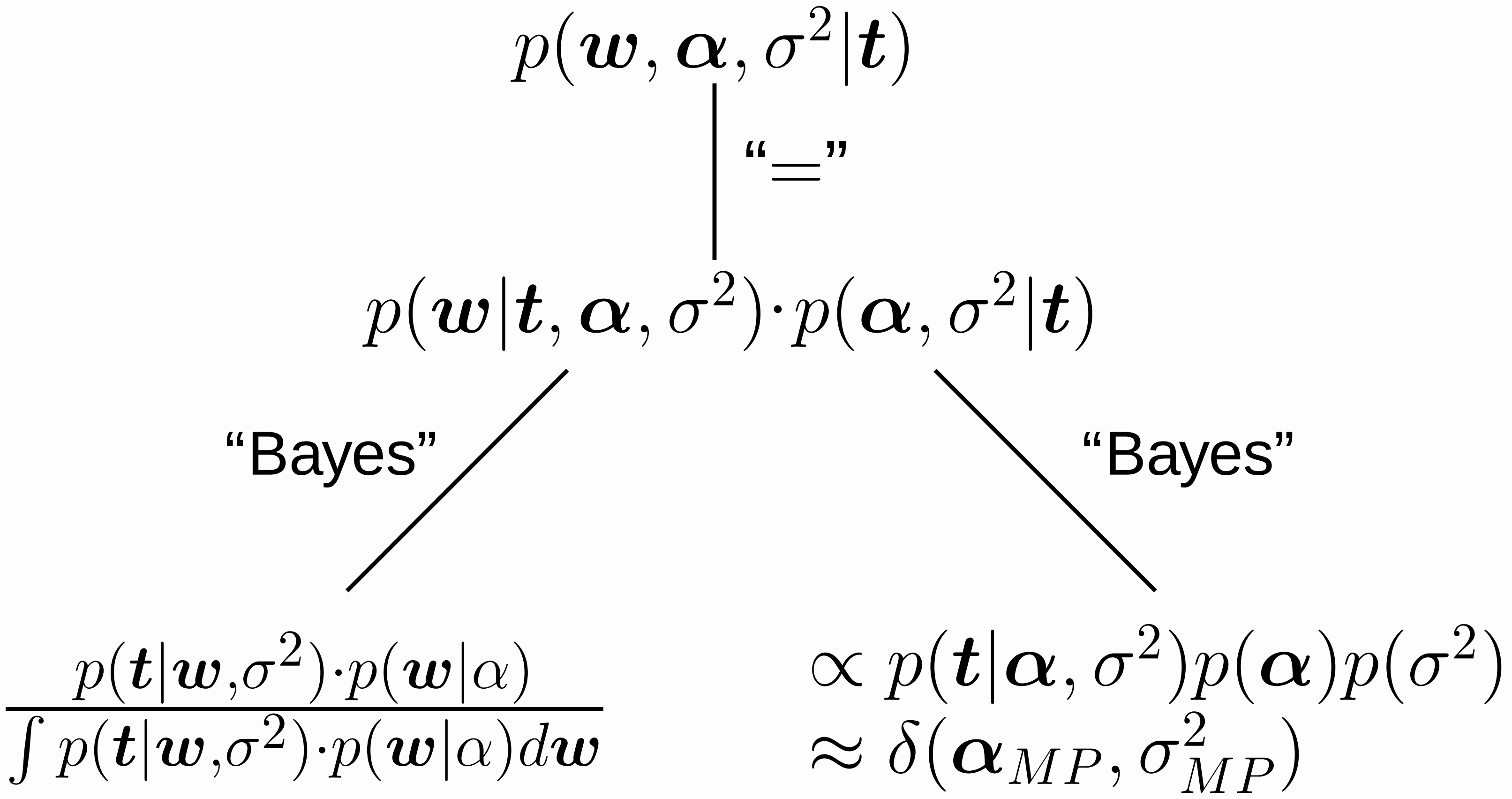
Where $p(\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2|\boldsymbol{t})$ is

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

This form has no analytical solution.

Inference

This is the way around :



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

"$=$"

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

"Bayes"

"Bayes"

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

$$\propto p(\boldsymbol{t}|\boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{\alpha}) p(\sigma^2)$$
$$\approx \delta(\boldsymbol{\alpha}_{MP}, \sigma^2_{MP})$$

Inference - posterior over w

The posterior over the weight is expressed as

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

Here, all the PDFs are Gaussian. Thus, we can obtain the analytical expression for the posterior PDF over the weights.

Inference - posterior over w

The posterior over the weights is expressed as

$$p(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{\alpha}, \sigma^2) = \frac{1}{(2\pi)^{\frac{N+1}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left\{-\frac{(\boldsymbol{w}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{w}-\boldsymbol{\mu})}{2}\right\}$$

where

$$\boldsymbol{\Sigma} = (\sigma^{-2} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \mathbf{A})^{-1}$$

$$\mathbf{A} = diag(\alpha_0, \alpha_1, \ldots, \alpha_N)$$

$$\boldsymbol{\mu} = \sigma^{-2} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \boldsymbol{t}$$

Inference - predictive approximation

In case of $p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{t})$ we have to adopt some approximations.
We exchange $p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{t})$ with a delta function at its mode in a sense that

$$p(t_*|\boldsymbol{t})_{MP} = \int p(t_*|\boldsymbol{\alpha}, \sigma^2)\delta(\boldsymbol{\alpha}_{MP}, \sigma^2_{MP})d\boldsymbol{\alpha}d\sigma^2 \approx$$

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

is a good approximation

Inference - predictive approximation

Relevance vector “learning” thus becomes the search for the hyperparameters that maximize

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

with respect to $\boldsymbol{\alpha}$ and σ^2.

Inference - marginal likelihood

In case of uniform priors, we only have to maximize the term

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

$$= \frac{(2\pi)^{-\frac{N}{2}}}{|\sigma^2 \mathbf{I} + \mathbf{\Phi} \mathbf{A}^{-1} \mathbf{\Phi}^T|^{\frac{1}{2}}} \exp\left\{ -\frac{\boldsymbol{t}^T (\sigma^2 \mathbf{I} + \mathbf{\Phi} \mathbf{A}^{-1} \mathbf{\Phi}^T)^{-1} \boldsymbol{t}}{2} \right\}$$

Its maximization is known as *type-II maximum likelihood* method.

Inference - optimization

In case of non-uniform priors, the maximization is a bit more complex, but finally leads to the iterative re-estimation formulas:

$$\alpha_i^{new} = \frac{(1-\alpha_i\Sigma_{ii})+2a}{\mu_i^2+2b},$$

$$(\sigma^2)^{new} = \frac{\|\boldsymbol{t}-\boldsymbol{\Phi}\boldsymbol{\mu}\|^2+2d}{N-\sum_i(1-\alpha_i\Sigma_{ii})+2c},$$

Σ_{ij} and μ_i are the scalar values taken from the corresponding matrix and vector.

Inference - the learning algorithm

- $$\alpha_i^{new} = \frac{(1 - \alpha_i \Sigma_{ii}) + 2a}{\mu_i^2 + 2b},$$
  $$(\sigma^2)^{new} = \frac{\|\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 + 2d}{N - \sum_i (1 - \alpha_i \Sigma_{ii}) + 2c},$$

- $$\boldsymbol{\Sigma} = (\sigma^{-2} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \mathbf{A})^{-1}$$
  $$\mathbf{A} = diag(\alpha_0, \alpha_1, \ldots, \alpha_N)$$
  $$\boldsymbol{\mu} = \sigma^{-2} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \boldsymbol{t}$$

Inference - introducing sparsity

A large proportion of α_i are driven to large values (in principle they become infinite) during the learning procedure.

Thus, $p(w_i|\boldsymbol{t}, \boldsymbol{\alpha}, \sigma^2)$ becomes highly peaked around zero – i.e. we are *a posteriori* “certain” that these w_i are zero.

The vectors $\boldsymbol{x}_i$ for which w_i are not zero are called relevance vectors.

Inference - Making predictions

Having found the maximizing values $\boldsymbol{\alpha}_{MP}$ and σ^2_{MP}, we can now compute predictions

$$p(t_*|\boldsymbol{t}, \boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) =$$

$$= \int p(t_*|\boldsymbol{w}, \sigma^2_{MP}) \cdot p(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) d\boldsymbol{w}$$

Both terms in the integrand are Gaussian, thus

Inference - Making predictions

the result can be readily computed to be

$$p(t_*|\boldsymbol{t}, \boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) = \mathcal{N}(t_*|y_*, \sigma^2_*)$$

with

$$y_* = \boldsymbol{\mu}^T \boldsymbol{\phi}(\boldsymbol{x}_*)$$
$$\sigma^2_* = \sigma^2_{MP} + \boldsymbol{\phi}(\boldsymbol{x}_*)^T \boldsymbol{\Sigma} \boldsymbol{\phi}(\boldsymbol{x}_*)$$

RVM example

Conclusions

- Generalization is typically very good.
- Learned models are typically highly sparse.
- There are no constraints imposed on the basis functions.
- Different input scales for input variable are possible.

References

- M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211-244, June 2001.
- Chris Bishop, “Probabilistic graphical models and their role in machine learning”, NATO ASI - LTP 2002 tutorial, Leuven, Belgium.