Advanced Computational Econometrics: Machine Learning

Chapter 9: Relevance Vector Machines

Thierry Denœux

July-August 2019

Limitations of SVM

- Support vector machines have been used in a variety of classification and regression applications.
- Nevertheless, they suffer from a number of limitations:
  - The outputs of an SVM represent decisions (classification) or point predictions. They are not probabilistic.
  - There is a complexity parameter C (as well as a parameter ϵ in the case of regression), which must be found by cross-validation.
  - Predictions are expressed as linear combinations of kernel functions that are centered on training data points and that are required to be positive definite.

Relevance Vector Machines

- The relevance vector machine or RVM (Tipping, 2001) is a Bayesian sparse kernel technique for regression and classification that shares many of the characteristics of the SVM whilst avoiding its principal limitations.
- Additionally, it typically leads to much sparser models resulting in correspondingly faster performance on test data whilst maintaining comparable generalization error.
- I will present RVM for regression, but the method can be transposed to classification.
- Source:

Tipping, M. E.
Sparse Bayesian learning and the relevance vector machine.
Journal of Machine Learning Research 1, 211–244

Overview

Model

- The RVM for regression is a Bayesian linear regression method, with a modified prior that results in sparse solutions.
- The model defines a conditional distribution for a real-valued target variable Y, given an input vector x, which takes the form

$$Y \sim \mathcal{N}(f(x), \beta^{-1})$$

where $\beta = \sigma^{-2}$ is the noise precision, and

$$f(x) = \sum_{j=1}^{M} w_j \Phi_j(x) = w^T \Phi(x)$$

with fixed nonlinear basis functions $\Phi_j(x)$, which typically include 1 so that the corresponding weight parameter represents the constant term.

Choice of basis functions

- The RVM is a specific instance of this model, which is intended to mirror the structure of the support vector machine.
- In particular, the basis functions are given by kernels, with one kernel associated with each of the data points from the training set:

$$f(x) = \sum_{i=1}^{n} w_i \mathcal{K}(x, x_i) + b,$$

  where $\mathcal{K}$ is a kernel function and b is a constant. We then have $M = n + 1$ parameters.
- However, the subsequent analysis is valid for arbitrary choices of basis functions, and for generality we shall work with the general form.
- In contrast to the SVM, there is no restriction to positive-definite kernels, nor are the basis functions tied in either number or location to the training data points.

Likelihood

- Let $\boldsymbol{X}$ be the matrix with i-th row x_i^T, $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ the vector of target values, and $\boldsymbol{\Phi}$ the $n \times M$ matrix with i-th row $\Phi(x_i)^T$.
- The (conditional) likelihood is

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{X}, w, \beta) &= \prod_{i=1}^{n} p(y_i \mid x_i, w, \beta) \\ &= \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\Phi} w, \beta^{-1} \boldsymbol{I}_n) \end{aligned}$$

Prior on w

- Next we introduce a prior distribution over the parameter vector w. As in standard Bayesian linear regression, we shall consider a zero-mean Gaussian prior.
- However, the key difference in the RVM is that we introduce a separate hyperparameter α_i for each of the weight parameters w_i instead of a single shared hyperparameter:

$$p(w \mid \boldsymbol{\alpha}) = \prod_{j=1}^{M} \mathcal{N}(w_j \mid 0, \alpha_j^{-1}),$$

  where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$.
- We shall see that, when we learn these hyperparameters from the data, a significant proportion of them go to infinity. The corresponding weight parameters have posterior distributions that are concentrated at zero, resulting in a sparse model.

Overview

Overview

Posterior distribution of w

- If we know $\boldsymbol{\alpha}$ and β, we can compute the posterior distribution of w as

$$p(w \mid \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\alpha}, \beta) \propto p(\boldsymbol{y} \mid \boldsymbol{X}, w, \beta) p(w \mid \boldsymbol{\alpha})$$

with

$$p(\boldsymbol{y} \mid \boldsymbol{X}, w, \beta) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\Phi} w, \beta^{-1} \boldsymbol{I}_n)$$

and

$$p(w \mid \boldsymbol{\alpha}) = \mathcal{N}(w \mid 0, \boldsymbol{A}^{-1}),$$

where $\boldsymbol{A} = \text{diag}(\boldsymbol{\alpha})$.

- It can be shown that the posterior distribution of w is Gaussian, with parameters that can be computed using the following proposition.

Posterior distribution of w (continued)

Proposition

Given $p(x) = \mathcal{N}(x \mid \mu, \boldsymbol{\Lambda}^{-1})$ *and* $p(y \mid x) = \mathcal{N}(y \mid \boldsymbol{B}x + b, \boldsymbol{L}^{-1})$, *we have*

$$p(y) = \mathcal{N}(y \mid \boldsymbol{B}\mu + b, \boldsymbol{L}^{-1} + B\boldsymbol{\Lambda}^{-1}\boldsymbol{B}^T) \tag{1}$$

$$p(x \mid y) = \mathcal{N}(x \mid \boldsymbol{\Sigma}\left\{\boldsymbol{B}^T\boldsymbol{L}(y - b) + \boldsymbol{\Lambda}\mu\right\}, \boldsymbol{\Sigma}) \tag{2}$$

with $\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \boldsymbol{B}^T\boldsymbol{L}\boldsymbol{B})^{-1}$.

Using (2) with $x = w$, $y = \boldsymbol{y}$, $\boldsymbol{B} = \boldsymbol{\Phi}$, $b = 0$, $\boldsymbol{L} = \beta\boldsymbol{I}_n$, $\mu = 0$, $\boldsymbol{\Lambda} = \boldsymbol{A}$, we get $p(w \mid \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\alpha}, \beta) = \mathcal{N}(w \mid m, \boldsymbol{\Sigma})$ with

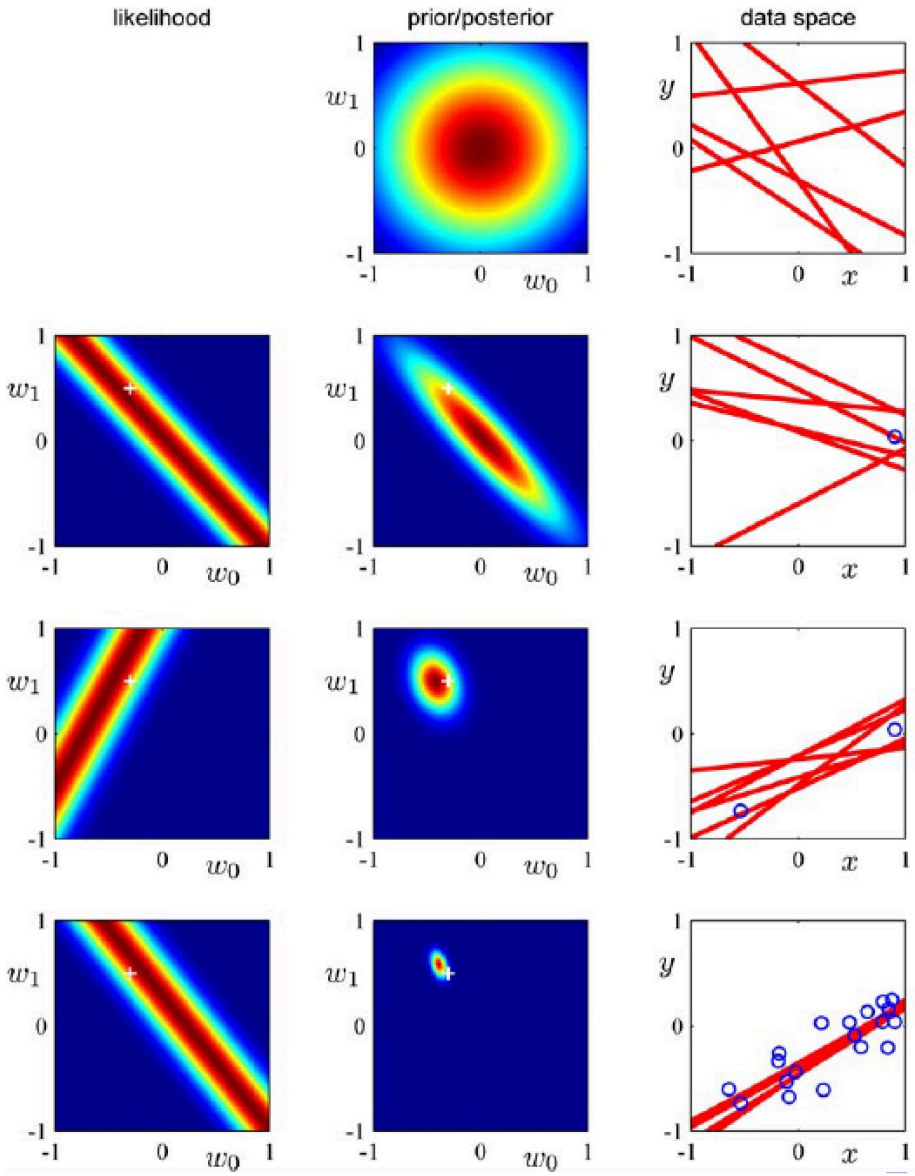
$$\boldsymbol{\Sigma} = (\boldsymbol{A} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}$$

$$m = \beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\boldsymbol{y}$$

Remarks

1. With infinitely broad priors, $\alpha_i \to 0$, $\boldsymbol{\Sigma} \to \beta^{-1}(\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1} = \text{Var}(\widehat{w})$, and
$$m \to (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y} = \widehat{w},$$
where $\widehat{w}$ is the LS estimate of w.
2. We can write
$$-\log p(w \mid \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\alpha}, \beta) = \frac{\beta}{2}\sum_{i=1}^{n}(y_i - w^T\Phi(x_i))^2 + \frac{1}{2}\sum_{j=1}^{M}\alpha_j w_j^2$$
The mode m of the posterior distribution of w is thus the solution of a penalized least-squares problem (similar to ridge regression, but with one penalization coefficient for each component of w).

Illustration of Bayesian learning ($f(x) = w_0 + w_1 x$, $\alpha_0 = \alpha_1$)



Overview

Predictive distribution of y

- In practice, we are mainly interested in making predictions of y for new values of x.
- Given $\boldsymbol{\alpha}$ and β, the predictive distribution of y is

$$p(y \mid x, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}, \beta) = \int \underbrace{p(y \mid x, w, \beta)}_{\mathcal{N}(\Phi(x)^T w, \beta^{-1}\boldsymbol{I}_n)} \underbrace{p(w \mid \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\alpha}, \beta)}_{\mathcal{N}(m;\boldsymbol{\Sigma})} dw$$

- Again, this is a Gaussian distribution, whose parameters can be obtained from (1) with $y = y$, $x = w$, $\boldsymbol{B} = \Phi(x)^T$, $b = 0$, $\boldsymbol{L} = \beta \boldsymbol{I}_n$, $\mu = m$ and $\boldsymbol{\Lambda}^{-1} = \boldsymbol{\Sigma}$. We get

$$p(y \mid x, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}, \beta) = \mathcal{N}(y \mid \Phi(x)^T m, \sigma_n^2(x))$$

with

$$\sigma_n^2(x) = \frac{1}{\beta} + \Phi(x)^T \boldsymbol{\Sigma} \Phi(x)$$

Remark

- In the expression of the variance

$$\sigma_n^2(x) = \frac{1}{\beta} + \Phi(x)^T \mathbf{\Sigma} \Phi(x),$$

the first term represents the noise on the data whereas the second term reflects the uncertainty associated with the parameters w.

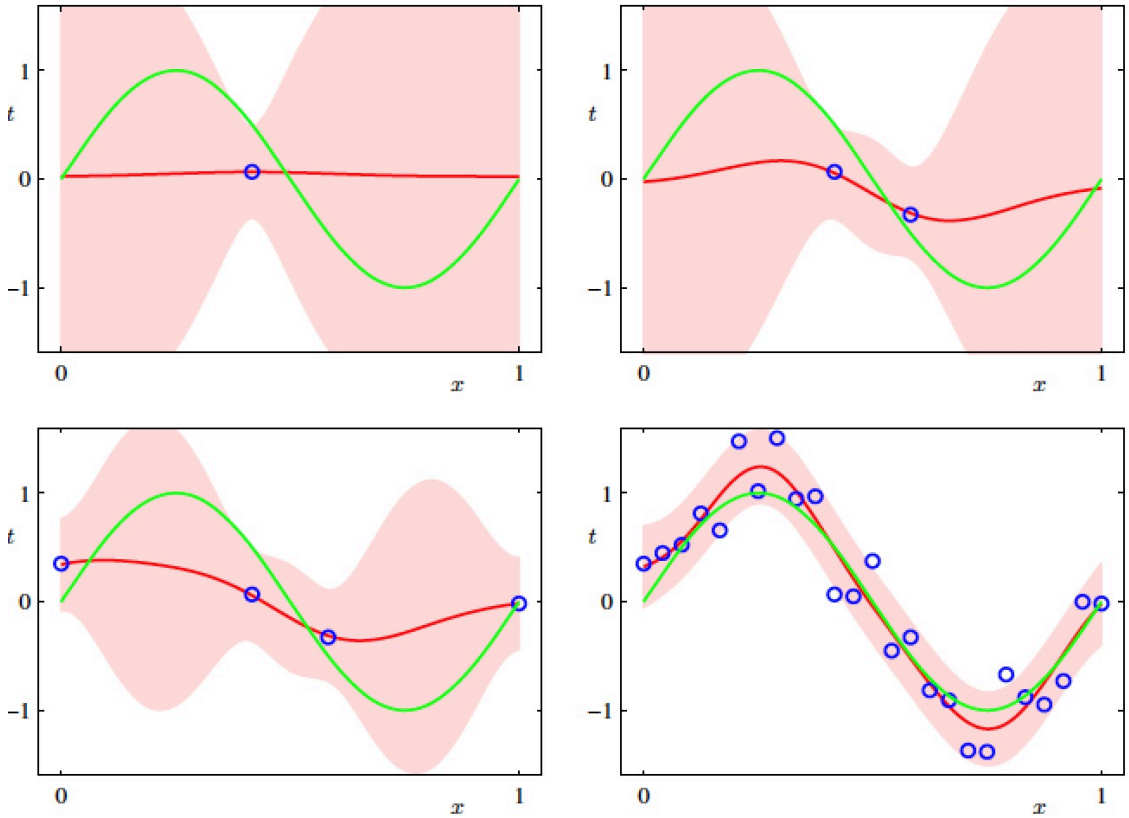
- As additional data points are observed, the posterior distribution becomes narrower. As a consequence it can be shown that

$$\sigma_{n+1}^2(x) \leq \sigma_n^2(x)$$

and

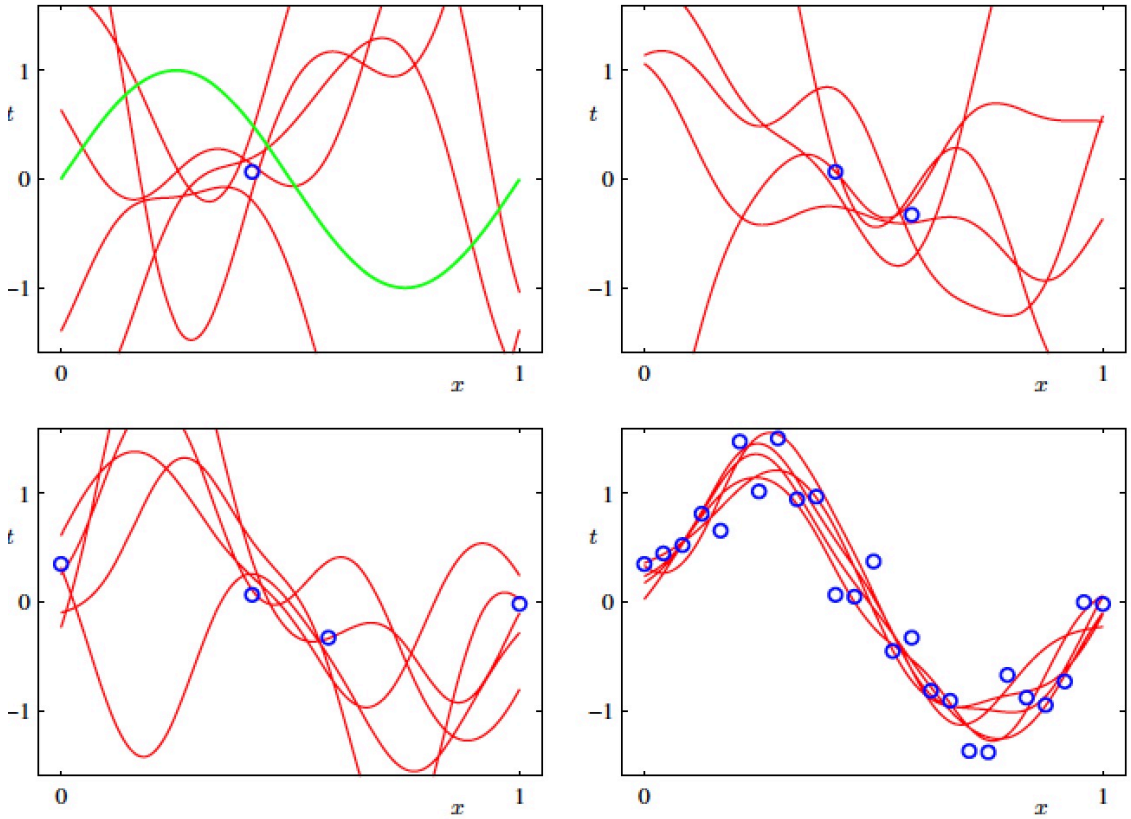
$$\lim_{n \to \infty} \sigma_n^2(x) = \frac{1}{\beta}$$

Example



Examples of predictive distribution for a model with 9 Gaussian basis functions. The green curves correspond to the function $\sin(2\pi x)$ with Gaussian noise. The red curve shows the mean of the corresponding Gaussian predictive distribution and the red shaded region spans one standard deviation either side of the mean.

Example (continued)



In order to gain insight into the covariance between the predictions at different values of x, we can draw samples from the posterior distribution over w, and then plot the corresponding functions $f(x, w)$.

Overview

Overview

Integrating out $\boldsymbol{\alpha}$ and β

- Until now, we have assumed that hyperparameters $\boldsymbol{\alpha}$ and β are known.
- In a fully Bayesian treatment of the linear basis function model, we would introduce prior distributions over $\boldsymbol{\alpha}$ and β and make predictions by marginalizing with respect to these hyperparameters as well as with respect to the parameters w:

$$p(y \mid x, \boldsymbol{X}, \boldsymbol{y}) = \iiint p(y \mid x, w, \beta) \, p(w \mid \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}, \beta) p(\boldsymbol{\alpha}, \beta \mid \boldsymbol{X}, \boldsymbol{y}) dw \, d\boldsymbol{\alpha} \, d\beta$$

- However, although we can integrate analytically over either w or over the hyperparameters, the complete marginalization over all of these variables is analytically intractable.

Approximation

- As an approximation, we can set the hyperparameters to specific values determined by maximizing the marginal likelihood function obtained by first integrating over the parameters w.
- This framework is known as
  - Empirical Bayes or generalized maximum likelihood in the statistics literature
  - Evidence approximation in the machine learning literature.
- In this approach, the hyperparameters are determined directly from the data: we do not need to tune them by cross-validation, as for the SVMs.

Evidence approximation

- If the posterior distribution $p(\boldsymbol{\alpha}, \beta \mid \boldsymbol{X}, \boldsymbol{y})$ is sharply peaked around values $\boldsymbol{\alpha}^*$ and β^*, then the predictive distribution is obtained simply by marginalizing over w in which $\boldsymbol{\alpha}$ and β are fixed to the values $\boldsymbol{\alpha}^*$ and β^*, so that

$$p(y \mid x, \boldsymbol{X}, \boldsymbol{y}) \approx p(y \mid x, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^*, \beta^*) = \int p(y \mid x, w, \beta^*) p(w \mid \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^*, \beta^*) dw$$

- We choose as $\boldsymbol{\alpha}^*$ and β^* the values that maximize

$$p(\boldsymbol{\alpha}, \beta \mid \boldsymbol{X}, \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\alpha}, \beta) p(\boldsymbol{\alpha}, \beta)$$

- If the prior is relatively flat, then the values $\boldsymbol{\alpha}^*$ and β^* are obtained by maximizing the marginal likelihood function $p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\alpha}, \beta)$ called the evidence function.

Overview

Expression of the evidence function

- We have

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\alpha}, \beta) = \int p(\boldsymbol{y} \mid \boldsymbol{X}, w, \beta) p(w \mid \boldsymbol{\alpha}) dw$$

- Once again, we can use Eq. (1) to compute this integral, with $y = \boldsymbol{y}$, $x = w$, $\boldsymbol{B} = \boldsymbol{\Phi}$, $\boldsymbol{L} = \beta \boldsymbol{I}_n$, $\boldsymbol{\Lambda} = \boldsymbol{A}$, $\mu = 0$. We get

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\alpha}, \beta) = \mathcal{N}(\boldsymbol{y} \mid 0, \boldsymbol{C})$$

where $\boldsymbol{C}$ is the $n \times n$ matrix

$$\boldsymbol{C} = \beta^{-1} \boldsymbol{I}_n + \boldsymbol{\Phi} \boldsymbol{A}^{-1} \boldsymbol{\Phi}^T.$$

Maximization of the evidence function

- The log-likelihood is

$$\log p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\alpha}, \beta) = -\frac{1}{2}\left(n \log(2\pi) + \log|\boldsymbol{C}| + \boldsymbol{y}^T \boldsymbol{C}^{-1} \boldsymbol{y}\right)$$

- To maximize $\log p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\alpha}, \beta)$ we compute the derivatives with respect to the hyperparameters $\boldsymbol{\alpha}$ and β.
- After some tedious calculation, we find

$$\frac{\partial \log p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\alpha}, \beta)}{\partial \alpha_j} = -\frac{1}{2}\left(\frac{1}{\alpha_j} - \Sigma_{jj} - m_j^2\right)$$

$$\frac{\partial \log p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\alpha}, \beta)}{\partial \beta} = \frac{1}{2}\left(\frac{n}{\beta} - \|\boldsymbol{y} - \boldsymbol{\Phi} m\|^2 - \frac{1}{\beta}\text{trace}(\boldsymbol{I}_M - \boldsymbol{A}\boldsymbol{\Sigma})\right)$$

Maximization of the evidence function (continued)

- Setting these derivatives to zero, we get update equations for $\boldsymbol{\alpha}$ and β:

$$\frac{1}{\alpha_j} - \Sigma_{jj} - m_j^2 \Leftrightarrow \alpha_j = \frac{\gamma_i}{m_j^2},$$

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

$$\frac{n}{\beta} - \|\boldsymbol{y} - \boldsymbol{\Phi} m\|^2 - \frac{1}{\beta} \underbrace{\text{trace}(\boldsymbol{I}_M - \boldsymbol{A}\boldsymbol{\Sigma})}_{\sum_{j=1}^M \gamma_j} = 0 \Leftrightarrow \beta = \frac{n - \sum_{j=1}^M \gamma_j}{\|\boldsymbol{y} - \boldsymbol{\Phi} m\|^2}.$$

- Remark: these equations are implicit, because both γ_j and m depend on $\boldsymbol{\alpha}$ and β.
- Algorithm: initialize $\boldsymbol{\alpha}$ and β, compute m and $\boldsymbol{\Sigma}$, update $\boldsymbol{\alpha}$ and β, and iterate until convergence.

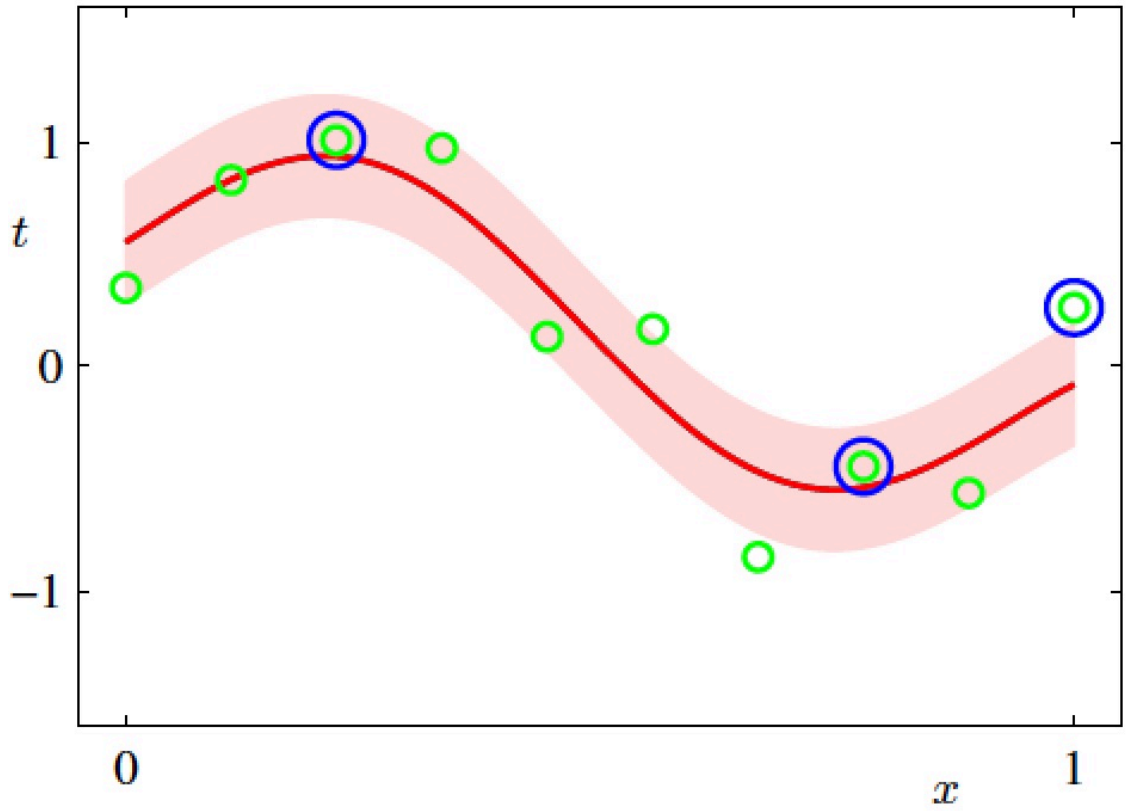
Interpretation of the γ_i

- Each γ_j can be interpreted as a measure of how "well-determined" its corresponding parameter w_j is by the data.
- For α_j large, w_j is highly constrained by the prior, $\Sigma_{jj} \approx \alpha_j^{-1}$ and $\gamma_j \approx 0$.
- Conversely, when α_j is small and w_j fits the data, $\gamma_j \approx 1$.

Sparsity

- As a result of the optimization, we find that a proportion of the hyperparameters α_j are driven to large (in principle infinite) values, and so the weight parameters w_j corresponding to these hyperparameters have posterior distributions with mean and variance both zero.
- Thus those parameters, and the corresponding basis functions $\Phi_j(x)$, are removed from the model and play no role in making predictions for new inputs.
- In the case of models of the form $\Phi_i(x) = \mathcal{K}(x, x_i)$, the inputs x_i corresponding to the remaining nonzero weights are called relevance vectors, because they are identified through the mechanism of automatic relevance determination, and are analogous to the support vectors of an SVM.

Example



The mean of the predictive distribution for the RVM is shown by the red line, and the one standard- deviation predictive distribution is shown by the shaded region. Also, the data points are shown in green, and the relevance vectors are indicated by blue circles.

Summary of the Algorithm

1. Select a suitable kernel function for the data set and relevant parameters. Create the design matrix $\mathbf{\Phi}$.
2. Set thresholds ϵ_α, ϵ_β and τ
3. Choose starting values for $\boldsymbol{\alpha}$ and β.
4. Calculate

$$\mathbf{\Sigma} = (\boldsymbol{A} + \beta \mathbf{\Phi}^T \mathbf{\Phi})^{-1} \quad \text{and} \quad m = \beta \mathbf{\Sigma} \mathbf{\Phi}^T \boldsymbol{y}.$$

5. Update

$$\alpha_j^{new} = \frac{\gamma_j}{m_j^2} \quad \text{and} \quad \beta^{new} = \frac{n - \sum_{j=1}^{M} \gamma_j}{\|\boldsymbol{y} - \mathbf{\Phi} m\|^2}$$

6. Prune the basis functions $\Phi_j(x)$ for all j such that $\alpha_j > \tau$.
7. Repeat (4) to (6) until $|\beta^{new} - \beta^{old}| < \epsilon_\beta$ and $\|\alpha_j^{new} - \alpha_j^{old}\| < \epsilon_\alpha$

Complexity

- The principal disadvantage of the RVM compared to the SVM is that training involves optimizing a nonconvex function, and training times can be longer than for a comparable SVM.
- For a model with M basis functions, the RVM requires inversion of a matrix of size $M \times M$, which in general requires $O(M^3)$ computation.
- In the specific case of the SVM-like model, we have $M = n + 1$. In contrast, there are techniques for training SVMs whose cost is roughly quadratic in n. However:
  - In the case of the RVM we can start with a smaller number of basis functions than $n + 1$.
  - In the RVM the parameters governing complexity and noise variance are determined automatically from a single training run, whereas in the support vector machine the parameters C and ϵ are generally found using cross-validation, which involves multiple training runs.

Overview

Application in R

```
>library('kernlab')
>library('MASS')

>fit<-rvm(accel ~ times,data=mcycle,kernel="rbfdot",kpar=list(sigma=0.1))

> fit
Relevance Vector Machine object of class "rvm"
Problem type: regression

Gaussian Radial Basis kernel function.
Hyperparameter : sigma = 0.1

Number of Relevance Vectors : 5
Variance : 497.0264
Training error : 483.173922435
```

Application in R (continued)

```
> alpha(fit)
-113.70005 38.65428 -109.91277 22.32886 22.32418
> RVindex(fit)
58 64 66 94 95
```

```
mcycle.test<-data.frame(times=seq(0,60,0.1))
ytest <- predict(fit, newdata=mcycle.test)

plot(mcycle$times, mcycle$accel, type ="p",
xlab='time',ylab='acceleration')
lines(mcycle.test$times, ytest, col="red",lwd=2)
points(mcycle$times[RVindex(fit)], mcycle$accel[RVindex(fit)],pch=16)
```

Result