1 Lecture Outline

Reading: Chapter 7 Sparse Kernel Machines

This lecture reviews the *Relevance Vector Machines Explained* by T. Fletcher

http://www.tristanfletcher.co.uk/RVM%20Explained.pdf

2 Introduction

In the regression problem, we wish to find the set of model parameters, $\mathbf{w}$ so that we can predict y for a given M-dimensional input $\mathbf{x}'$. We assume a non-linear relationship between $\mathbf{x}$ and y, use a basis function, ϕ for the nonlinear mapping, and have incorporated the bias in $\mathbf{w}$. Mathematically, this prediction is given by

$$y = \mathbf{w}^T\phi(\mathbf{x}'). \tag{1}$$

In order to determine $\mathbf{w}$, we have a set of N training points $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and associated target values $\{t_1, \ldots, t_N\}$. We assume each target is representative of the true model y but with the addition of noise

$$\begin{aligned} t_i &= y_i + \epsilon_i \\ &= \mathbf{w}^T\phi(\mathbf{x}_i) + \epsilon_i \end{aligned} \tag{2}$$

where we assume ϵ_i are independent and distributed as $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. This means that

$$\begin{aligned} p(t_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) &= \mathcal{N}(y_i, \sigma^2) \\ &= \mathcal{N}(\mathbf{w}^T\phi(\mathbf{x}_i), \sigma^2). \end{aligned} \tag{3}$$

Looking at the N training points simultaneously and collecting the training points as

$$\begin{aligned} \mathbf{t} &= [t_1, \ldots, \mathbf{t}_N]^T \\ \mathbf{\Phi} &= [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]^T \end{aligned} \tag{4}$$

we have for the likelihood function

$$p(\mathbf{t}|\mathbf{x}_i, \mathbf{w}, \sigma^2) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{w}^T\phi(\mathbf{x}_i), \sigma^2). \tag{5}$$

3 Posterior Probability

When attempting to learn the relationship between $\mathbf{x}$ and y, we wish to constrain the complexity and hence the growth of weights $\mathbf{w}$. To do this, we define an explicit *prior* probability distribution on $\mathbf{w}$

$$p(\mathbf{w}|\alpha_i) = \mathcal{N}(0, \alpha_i^{-1}) \tag{6}$$

where α_i is the precision of each w_i. Looking at the N points simultaneously and letting $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$, the *weight prior* takes the form

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{N} \mathcal{N}(0, \alpha_i^{-1}). \tag{7}$$

This means that there is an *individual* hyperparameter α_i associated with each weight, modifying the strength of the prior.

The *posterior* probability over all the unknown parameters, given the training data, is given by

$$p(\mathbf{w}, \boldsymbol{\alpha}, \beta|\mathbf{t}) = p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}, \beta)p(\boldsymbol{\alpha}, \beta|\mathbf{t}). \tag{8}$$

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

We can show that the first factor of (8) is given by

$$p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}, \beta) = \mathcal{N}(\mathbf{m}, \boldsymbol{\Sigma}) \tag{9}$$

where

$$\begin{aligned} \mathbf{m} &= \beta \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{t} \\ \boldsymbol{\Sigma} &= \left[\text{diag}(\alpha_1, \ldots, \alpha_N) + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}\right]^{-1}. \end{aligned} \tag{10}$$

For the second factor of (8) we have

$$p(\boldsymbol{\alpha}, \beta|\mathbf{t}) \propto p(\mathbf{t}|\boldsymbol{\alpha}, \beta)p(\boldsymbol{\alpha})p(\beta). \tag{11}$$

We want to maximize the *posterior* in (8) with the appropriate choice of $\mathbf{w}$ and hyperparameters $\boldsymbol{\alpha}, \beta$.

The first factor in (8) leads us to the choice of $\mathbf{w}$ and requires the evaluation of $\mathbf{m}, \boldsymbol{\Sigma}$ which depends on $\boldsymbol{\alpha}, \beta$. The second factor in (8) leads us to the choice of $\boldsymbol{\alpha}, \beta$. We thus focus on maximizing the second factor via (11) and in particular, focus on $p(\mathbf{t}|\boldsymbol{\alpha}, \beta)$ known as the *evidence*. We will assume uniform hyperpriors and thus ignore $p(\boldsymbol{\alpha})$ and $p(\beta)$.

4 Maximizing the Evidence

We can rewrite the expression for the evidence as a *marginal likelihood*

$$p(\mathbf{t}|\boldsymbol{\alpha}, \beta) = \left(\frac{\beta}{2\pi}\right)^{N/2} \left(\frac{1}{2\pi}\right)^{M/2} \prod_{i=1}^{M} \alpha_i^{1/2} \exp\left\{-E(\mathbf{t})\right\} (2\pi)^{M/2} |\boldsymbol{\Sigma}|^{1/2} \tag{12}$$

where

$$E(\mathbf{t}) = \frac{1}{2}\left(\beta \mathbf{t}^T \mathbf{t} - \mathbf{m}^T \boldsymbol{\Sigma}^{-1} \mathbf{m}\right). \tag{13}$$

For purposes of maximizing the evidence or now, maximizing the marginal likelihood, we instead maximize the *log marginal likelihood*

$$\ln p(\mathbf{t}|\boldsymbol{\alpha}, \beta) = \frac{N}{2} \ln \beta - E(\mathbf{t}) - \frac{1}{2} \ln |\boldsymbol{\Sigma}| - \frac{N}{2} \ln(2\pi) + \frac{1}{2} \sum_{i=1}^{M} \ln \alpha_i. \tag{14}$$

Differentiating (14) with respect to α_i and setting to zero to obtain a recursion

$$\begin{aligned} \alpha_i &= \frac{1 - \alpha_i \Sigma_{ii}}{m_i^2} \\ &= \frac{\gamma_i}{m_i^2} \end{aligned} \tag{15}$$

where $\gamma_i = 1 - \alpha_i \Sigma_{ii}$. Differentiating (14) with respect to β and setting to zero yields

$$\beta = \frac{N - \sum_i \gamma_i}{||\mathbf{t} - \boldsymbol{\Phi}\mathbf{m}||^2}. \tag{16}$$

5 Maximization of the Posterior

The hyperparameters, $\boldsymbol{\alpha}$ and β which maximize the evidence or marginal likelihood are found by setting $\boldsymbol{\alpha}$ and β to initial values, finding values for $\mathbf{m}$ and $\boldsymbol{\Sigma}$ in (10), using these to update estimates for $\boldsymbol{\alpha}$ and β, and repeating this process until convergence. Once, $\boldsymbol{\alpha}$ and β are decided, $\mathbf{m}$ and $\boldsymbol{\Sigma}$ are decided in (10).

With this, we have maximized the posterior in (8) since we can now compute (9) and $p(\boldsymbol{\alpha}, \beta|\mathbf{t})$.

6 Predictions and the Predictive Distribution

Referring back to the target distribution for training point $\mathbf{x}_i$ in (3) for a new input $\mathbf{x}'$

$$p(t_i|\mathbf{x}', \boldsymbol{\alpha}, \beta) = \mathcal{N}(\mathbf{m}^T\phi(\mathbf{x}'), \sigma^2(\mathbf{x}')). \tag{17}$$

where $\mathbf{m}$ is the posterior mean of $\mathbf{w}$ in (9) and

$$\sigma^2(\mathbf{x}') = \beta^{-1} + \phi(\mathbf{x}')^T\boldsymbol{\Sigma}\phi(\mathbf{x}'). \tag{18}$$

We use as our point estimate for t the mean of the distribution,

$$y = \mathbf{m}^T\phi(\mathbf{x}'). \tag{19}$$

7 Automatic Relevance Determination

While carrying out the recursions to maximize the evidence many of the α_i will tend to infinity which implies that the associated $w_i = 0$. In this case, the corresponding $\phi(\mathbf{x}_i)$ can be pruned from the design matrix $\boldsymbol{\Phi}$ each each iteration. The $\mathbf{x}_i$ corresponding to the remaining non-zero weights after pruning 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.

8 Regression Example

Figure 9 shows an example of the RVM applied to the sinusoidal regression data set. We see that the number of relevance vectors in the RVM is significantly smaller than the number of support vectors in the SVM. For a wide range of regression and classification tasks, the RVM is found to give models that are typically an order of magnitude more compact than the corresponding SVM, resulting in significant improvement in the sped of processing on test data. Remarkably, this greater sparsity is achieved with little or no reduction in generalization error compared with the corresponding SVM.

The principal disadvantage of the RVM compared to the SVM is that the training involves optimizing a nonconvex function, and training times can be longer than for a comparable SVM. Note that there exists an alternative approach to the two described in the text that significantly improves training speed.

9 Classification Example

See text for development of the RVM for classification. Figure 10 shows the RVM applied to a synthetic classification data set. We see that the relevance vectors tend not to lie in the region of the decision boundary, in contrast to the SVM. One of the potential advantages of the RVM compared with the SVM is that it makes probabilistic predictions.

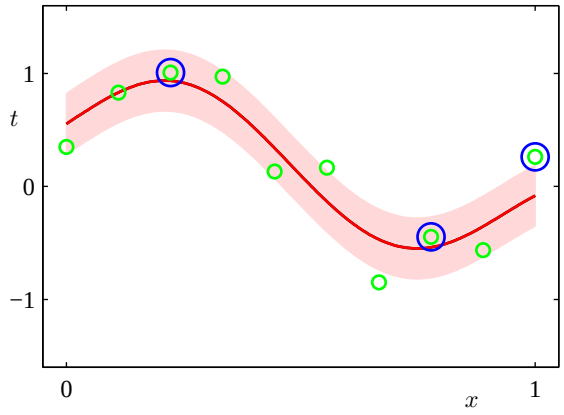
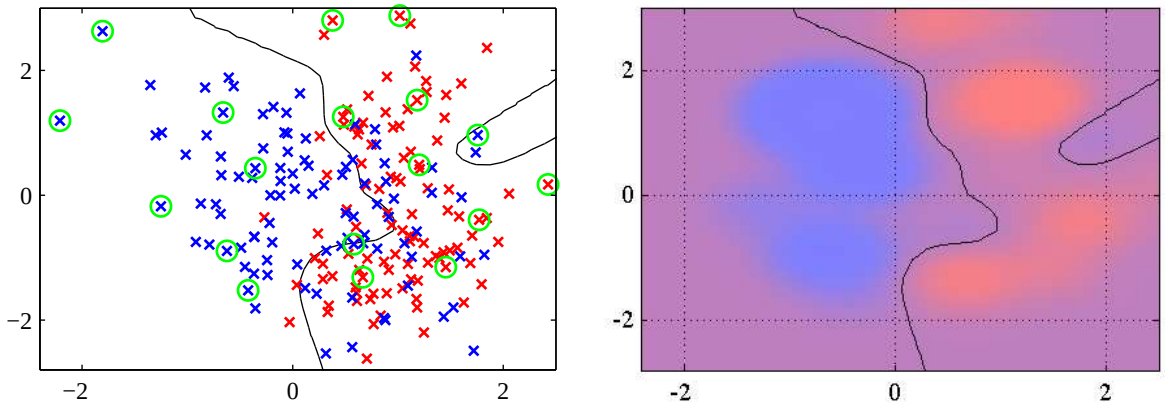


Figure 9:



(a) (b)

Figure 10:

See http://www.vectoranomaly.com/downloads/downloads.htm

for MATLAB implementation of RVM.