Task 3. Sparse Supervised Learning: Relevance Vector Machine

Garoe Dorta-Perez
CM50246: Machine Learning and AI

December 10, 2014

1 Introduction

SVM are a popular choice in classification problems. However they are inherently not probabilistic. A technique that uses the same intuition is the Relevance Vector Machine. It uses a full Bayesian approach with a prior that encourages sparseness.

2 The problem

First a dual model is used, previously we would model the data as a normal distribution. Where w is a one dimensional array with the world state, x is the data point, ϕ are the parameters of a linear function of the data and σ is the standard deviation. We now apply a transformation on the parameters as shown in Equation 1, so ϕ is now a weighted sum over the observed data points, where ψ is a vector with the weights. If we are working with few data of high dimensionality, this sum is faster to calculate, as it has less terms, that the original linear dependency.

$$Pr(w_i|\mathbf{x}_i) = Norm_{x_i}[\phi^T\mathbf{x}_i, \sigma^2], \tag{1}$$

$$\phi = \mathbf{X}\psi, \tag{2}$$

Using the dual parameters ψ, we encourage sparsernes in the model by using the prior defined in Equation 3. Where I is the number of data points, *Stud* is a Student's t-distribution with ν degrees of freedom and Γ is the Gamma function. Using a product of t-distributions produces the desired sparseness since the areas with higher probability density are in the origin and along the axis.

$$Pr(\psi) = \prod_{i=1}^{I} Stud_{\psi_i}[0, 1, \nu], \tag{3}$$

$$= \prod_{i=1}^{I} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{\psi_i^2}{\nu}\right)^{-(\nu+1)/2}, \tag{4}$$

A t-distribution is not conjugate to a Normal distribution, so there is no simple closed form solution for the posterior. The solution will be to approximate the t-distributions by maximizing with respect to their hidden variables h, introduced in Equation 5, where $\mathbf{w}$ is a one dimensional array with the world state. This leads to the marginal likelihood shown in Equation 6, where $\mathbf{I}$ is the identity matrix, $\mathbf{X}$ is a matrix with the data points and $\mathbf{H}$ is a matrix with all the hidden variables.

$$Pr(\psi) = \prod_{i=1}^{I} \int Norm_{\mathbf{w}}[0, 1/h_i] Gam_{h_i}[\nu/2, \nu/2] dh_i, \tag{5}$$

$$Pr(\mathbf{w}|\mathbf{X}, \sigma^2) \approx \max_{\mathrm{H}} \left[Norm_{\mathbf{w}}\left[\mathbf{0}, \mathbf{X}^T\mathbf{X}\mathbf{H}^{-1}\mathbf{X}^T\mathbf{X} + \sigma^2\mathbf{I}\right] \prod_{d=1}^{D} Gam_{h_d}\left[\nu/2, \nu/2\right] \right], \tag{6}$$

The optimization is performed in three steps:

1. Optimize the marginal likelihood with respect to the hidden variables, using Equation 7.

$$h_i^{new} = \frac{1 - h_i \sum_{ii} + \nu}{\mu_i^2 + \nu}, \tag{7}$$

2. Update $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, using Equation 8.

$$\begin{aligned} \boldsymbol{\mu} &= \frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}^T\mathbf{X}\mathbf{w}, \\ \boldsymbol{\Sigma} &= \mathbf{A}^{-1}, \\ \mathbf{A} &= \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{X} + \mathbf{H}, \end{aligned} \tag{8}$$

3. Optimize the marginal likelihood with respect to the variance parameter, using Equation 9.

$$(\sigma^2)^{new} = \frac{1}{\mathbf{I} - \sum_i (1 - h_i \sum_{ii})} \left(\mathbf{w} - \mathbf{X}^T\mathbf{X}\boldsymbol{\mu}\right)^T \left(\mathbf{w} - \mathbf{X}^T\mathbf{X}\boldsymbol{\mu}\right), \tag{9}$$

After the training process, we only take the data points whose hidden variable h_i is smaller than a threshold. Since a larger one means a small ψ_i, and hence no significance in the weighted sum. In the implementation, we use a Gaussian kernel $k[x_i, x_j]$, as shown in Equation 10, instead of calculating the inner products $\mathbf{X}^T\mathbf{X}$, where λ controls the scale of the output. This is more efficient than computing the inner product.

$$k[x_i, x_j] = exp\left[-\frac{1}{2}\left(\frac{(x_i - x_j)^T(x_i - x_j)}{\lambda^2}\right)\right], \tag{10}$$

Since our world states $\mathbf{w_i}$ are multivariate, a separate regressor is used for each dimension in $\mathbf{w_i}$. However the hidden variables matrix $\mathbf{H}$ is shared among all of them.

3 Results

In Figures 1 and 2 we show results for a font data set. The RVM is trained with pictures of n in different fonts and their corresponding m in the same font. For testing, n with new fonts are used.

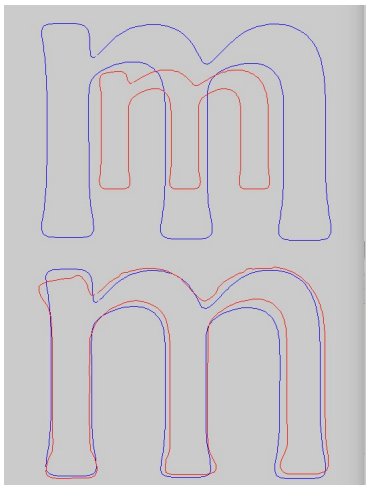

Figure 1: Blue is ground truth and red is RVM result. Top image corresponds to a kernel $\lambda = 2$, bottom image has $\lambda = 7$.

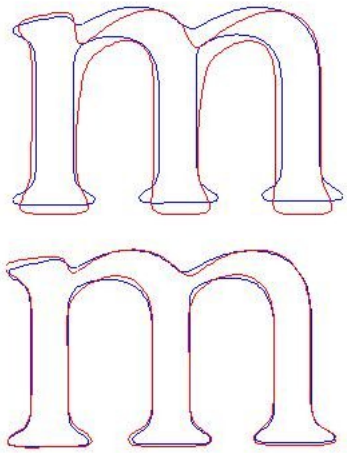

Figure 2: Blue is ground truth and red is RVM result. Top image corresponds $\nu = 0.001$, bottom image has $\nu = 1$

In Figure 1 we see the effects of changing the value of λ. Since it controls the scale, we can see how the m varies in size. In Figure 2 variations in ν are shown. Smaller ν encourages more sparsity, and so less relevance vectors been used. For the top image 5 relevance vectors are used, while for the bottom one 10 vectors are used. Less vectors means a more general and sparse model used. However the error in the prediction increases as we take fewer relevance vectors.