Task 3. Relevance Vector Machine

Garoe Dorta-Perez
CM50246: Machine Learning and AI

December 9, 2014

1 Introduction

SVM are a quite 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

Using the dual parameters ψ, we encourage sparsernes in the model by using the prior defined in Equation 1. Where I is the number of data points, Stud is the Student's t distribution with v degrees of freedom.

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

Assuming the data can be classified using linear functions, N activations are needed to enforce the constrains. Since we are solving for multi-class classification, a logistic sigmoid function as activation will not be valid. Instead a softmax is used for each a_n.

$$a_n = \phi_n^T x \tag{2}$$

$$\lambda_n = softmax_n[a_1, a_2 \cdots a_N] = \frac{exp[a_n]}{\sum_{m=1}^{N} exp[a_m]} \tag{3}$$

For the prior we are going to use a normal distribution with zero mean and σ^2 variance. In order to simplify the calculations we are going to minimise the log of the probability, where y_{in} is the softmax expression for class n and data i:

$$L = -log \sum_{i=1}^{I} y_{in} + \frac{1}{2\sigma^2} \phi^T \phi \tag{4}$$

We will use the gradient and Hessian updates shown below.

$$\begin{aligned}\frac{\partial L}{\partial \phi_n} &= \sum_{i=1}^{I} \left(y_{in} - \delta\left[\omega_i - n\right]\right)\mathbf{x}_i + \frac{\phi}{\sigma^2} \\ \frac{\partial^2 L}{\partial \phi_m \phi_n} &= \sum_{i=1}^{I} y_{im}\left(\delta\left[m-n\right] - y_{in}\mathbf{x}_i\mathbf{x}_i^T\right) + \frac{\delta\left[m-n\right]}{\sigma^2}\end{aligned} \tag{5}$$

Predictions are calculated through Laplace approximation and Monte Carlo integration.

$$predictions = \int y_{in}\mathcal{N}_a\left(\mu_a, \Sigma_a\right) da \tag{6}$$

3 Results

We tested the classification algorithm with two different datasets. The first one consists of hand drawn images of single digits from zero to nine, stored as a 784 vector of pixel values. While the second one consists of pictures of eight different objects against a blue background, stored as a 576 vector of pixel values. We present two tables showing prediction accuracy using different variances for the normal distribution prior and for several values of the ϕ_0 vector.

Digits dataset							
Prior Variance	1	10	100	1000	1	1	1
ϕ_0	0.1	0.1	0.1	0.1	-1	1	2.5
Prediction Accuracy	88%	86%	77%	48%	87%	87%	83%

ETH-80-HoG dataset							
Prior Variance	1	10	100	1000	1000	1000	1000
ϕ_0	0.125	0.125	0.125	0.125	-10	-1	10
Prediction Accuray	67%	74%	84%	89%	89%	89%	89%

For the *Digits* dataset is clear that a smaller variance value, yields better predictions. Thus indicating that a spiked normal distribution centred at zero is a reasonable guess. With respect to the numerical minimization, we do not see significant improvements when using different ϕ_0 values. Except for 2.5 which gives a decrease in performance, then indicating the existence of a worse local minima in that region.

While for the *ETH-80* dataset, the trend is better accuracy as the variance increases. Therefore the prior believe must be erroneous. The effect of the initial guess for ϕ does not seem to affect the end result. However, if set to zeros it does not converge or at least it does so quite slowly. Which could be an indicator of a valley where the gradient is zigzagging.