Task 1.1. Supervised Learning: Standard Classifier

Garoe Dorta-Perez
CM50246: Machine Learning and AI

November 12, 2014

1 Introduction

Given some pictures objects and having been asked to classify them in several groups, we are faced with a multi-class classification problem. One of the options would be to create N one-against-all binary classifiers. However an approach that naturally handles the multi-class nature of the problem is to use a categorical distribution to model our world.

2 The problem

As stated in the introduction, we are going to fit a Categorical probability model into our data. Using Bayes' rule we have:

$$Pr(\theta|x_{1\cdots I}) = \frac{\prod_{i=1}^{I} Pr(\omega = k_n|x,\theta)Pr(\theta)}{Pr(x_{1\cdots I})} \tag{1}$$

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 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:

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

With gradient and Hessian updates been:

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

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 28x28 matrix 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.

Below are the prediction accuracy results 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, ϕ_0 vector values smaller than -10 and bigger than 5 would give NaN. Regarding the prior variance it is clear that a smaller value, decreasing our confidence in the prior, yields better predictions. Thus indicating that normal distribution centred at zero is a reasonable guess. On the other hand, there are not 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. On the other hand, ϕ_0 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.