Task 1.1. Supervised Learning: Standard Classifier

Garoe Dorta-Perez
CM50246: Machine Learning and AI

November 12, 2014

1 Introduction

Given pictures from the world and been asked to classify them in several groups, we are faced with a problem of multi-class classification. One of the options would be to create N one-against-all binary classifiers. Using x to denote our data, ω to for the world state, and *lambda* for the probability of observing the given class. However a better one involves using a categorical distribution to model our world. Where $\boldsymbol{\lambda}$ is a vector that contains a λ for each class.

$$Pr(\omega|\mathbf{x}) = Cat_w[\boldsymbol{\lambda}[\mathbf{x}]] \tag{1}$$

2 The problem

As stated in the introduction, we are going to fit a Categorical probability model into our data. Denoting I as the total number of data points that we are given. Then, using Bayes' rule we have:

$$Pr(\theta|x_{1\cdots I}) = \frac{\prod_{i=1}^{I} Pr(\omega = k_n|x,\theta)Pr(\theta)}{Pr(x_{1\cdots I})} \tag{2}$$

We need N activation functions to enforce the constrains. Since we are solving for multi-class classification a logistic sigmoid function as activation will not be valid. Therefore a softmax function is used instead for each activation a_n.

$$a_n = \phi_n^T x \tag{3}$$

$$\lambda_n = softmax_n[a_1, a_2 \cdots a_N] = \frac{exp[a_n]}{\sum_{m=1}^{N} exp[a_m]} \tag{4}$$

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:

$$L = -log \sum_{i=1}^{I} y_{in} + \frac{1}{2\sigma^2}\phi^T\phi \tag{5}$$

With gradient and Hessian updates being:

$$\begin{aligned} \frac{\delta L}{\delta \phi_n} &= \sum_{i=1}^{I} \left(y_{in} - \delta\left[\omega_i - n\right]\right)\mathbf{x}_i + \frac{\phi}{\sigma^2} \\ \frac{\delta^2 L}{\delta \phi_m \phi_n} &= \sum_{i=1}^{I} y_{im}\left(\delta\left[m-n\right] - y_{in}\mathbf{x}_i\mathbf{x}_i^T\right) + \frac{\delta\left[m-n\right]}{\sigma^2} \end{aligned} \tag{6}$$

To make the predictions we evaluate a new sample doing a Laplace approximation and then a Monte Carlo integration

$$predictions = \int y_{in}\mathcal{N}_a\left(\mu_a, \Sigma_a\right) da \tag{7}$$

3 Results

We tested the classification algorithm with two different data sets. The first one consists of hand drawn images of single digits from zero to nine, stored as 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 value.

Below are the prediction accuracy results for different variances for the normal distribution in the prior and for the initial ϕ vector.

Digits data set							
Prior Variance	1	10	100	1000	1	1	1
Initial ϕ	0.1	0.1	0.1	0.1	-1	1	2.5
Prediction Accuracy	88%	86%	77%	48%	87%	87%	83%

ETH-80-HoG data set							
Prior Variance	1	10	100	1000	1000	1000	1000
Initial ϕ	0.125	0.125	0.125	0.125	-10	-1	10
Prediction Accuray	67%	74%	84%	89%	89%	89%	89%

For the *Digits* data set initial ϕ vector values smaller than -10 and bigger than 5 would give NaN.