HW5

1 Reject option in classifiers

(Source: [**?** , Q2.13])
In many classification problems one has the option either of assigning $\mathbf{x}$ to class j or, if you are too uncertain, of choosing the **reject option**. If the cost for rejects is less than the cost of falsely classifying the object, it may be the optimal action. Let α_i mean you choose action i, for $i = 1 : C + 1$, where C is the number of classes and $C + 1$ is the reject action. Let $Y = j$ be the true (but unknown) **state of nature**. Define the loss function as follows

$$\lambda(\alpha_i|Y = j) = \begin{cases} 0 & \text{if } i = j \text{ and } i, j \in \{1, \ldots, C\} \\ \lambda_r & \text{if } i = C + 1 \\ \lambda_s & \text{otherwise} \end{cases} \tag{1}$$

In otherwords, you incur 0 loss if you correctly classify, you incur λ_r loss (cost) if you choose the reject option, and you incur λ_s loss (cost) if you make a substitution error (misclassification).

1. Show that the minimum risk is obtained if we decide $Y = j$ if $p(Y = j|\mathbf{x}) \geq p(Y = k|\mathbf{x})$ for all k (i.e., j is the most probable class) *and* if $p(Y = j|\mathbf{x}) \geq 1 - \frac{\lambda_r}{\lambda_s}$; otherwise we decide to reject.
2. Describe qualitatively what happens as λ_r/λ_s is increased from 0 to 1 (i.e., the relative cost of rejection increases).

2 Setting hyper-parameters for the beta prior

1. Let $\theta \sim \text{Beta}(a, b)$. Sometimes our prior knowledge is not in the form of pseudo counts, so it is not immediately clear how to set a and b. But we may be able to express our prior in terms of an expected value, $E\ \theta = m$, and a variance, Var $\theta = v$, which is like a measure of confidence. Use the following properties of the Beta distribution to solve for a and b in terms of m and v.

$$E\ \theta = m = \frac{a}{a+b} \tag{2}$$

$$\text{Var}\ \theta = v = \frac{m(1-m)}{a+b+1} = \frac{ab}{(a+b)^2(a+b+1)} \tag{3}$$

2. Suppose θ is beta with mean 0.7 and standard deviation 0.2. What are the values of the hyper-parameters a and b that correspond to this?

3 Posterior predictive distribution for a batch of data with the dirichlet-multinomial model

In Section **??**, we showed that the posterior predictive distribution for a single multinomial trial, using a dirichlet prior, is

$$p(X = j|\mathcal{D}, \boldsymbol{\alpha}) = \frac{\alpha_j + N_j}{N + \sum_k \alpha_k} \tag{4}$$

Now consider predicting a *batch* of new data, $\tilde{\mathcal{D}} = (X_1, \ldots, X_m)$, consisting of m single multinomial trials (think of predicting the next m words in a sentence, assuming they are drawn iid). Derive an expression for

$$p(\tilde{\mathcal{D}}|\mathcal{D}, \alpha) \tag{5}$$

Your answer should be a function of $\boldsymbol{\alpha}$, and the old and new counts (sufficient statistics), defined as

$$N_k^{old} = \sum_{i \in \mathcal{D}} I(x_i = k), \quad N_k^{new} = \sum_{i \in \tilde{\mathcal{D}}} I(x_i = k) \tag{6}$$

Hint: recall that, for a vector of counts, $N_{1:K}$, the marginal likelihood (evidence) is given by

$$p(\mathcal{D}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha)}{\Gamma(N + \alpha)} \prod_k \frac{\Gamma(N_k + \alpha_k)}{\Gamma(\alpha_k)} \tag{7}$$

where $\alpha = \sum_k \alpha_k$ and $N = \sum_k N_k$.

4 Gaussian posterior credible interval

(Source: DeGroot)
Let $X \sim \mathcal{N}(\mu, \sigma^2 = 4)$ where μ is unknown but has prior $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2 = 9)$. The posterior after seeing n samples is $\mu \sim \mathcal{N}(\mu_n, \sigma_n^2)$. (This is called a credible interval, and is the Bayesian analog of a confidence interval.) How big does n have to be to ensure

$$p(\ell \leq \mu_n \leq u|D) \geq 0.95 \tag{8}$$

where (ℓ, u) is an interval (centered on μ_n) of width 1 and D is the data. Hint: recall that 95% of the probability mass of a Gaussian is within $\pm 1.96\sigma$ of the mean.

5 MAP estimation for 1D Gaussians

(Source: Jaakkola)
Consider samples $x_1, \ldots, x_n$ from a Gaussian random variable with known variance σ^2 and unknown mean μ. We further assume a prior distribution (also Gaussian) over the mean, $\mu \sim \mathcal{N}(m, s^2)$, with fixed mean m and variance s^2. (We will assume $s^2 > 0$, although it may be small.)
Throughout the following questions, we consider the "true" parameters μ and σ^2 as fixed. We will also fix m but consider the effects of varying the prior variance $s^2 > 0$ and the sample size n.

1. Calculate the MAP estimate $\hat{\mu}_{MAP}$. You can state the result without proof. Alternatively, with a lot more work, you can compute derivatives of the log posterior, set to zero and solve.
2. Show that as the number of samples n increase, the MAP estimate converges to the maximum likelihood estimate
3. Suppose n is small and fixed. What does the MAP estimator converge to if we increase the prior variance s^2?
4. Suppose n is small and fixed. What does the MAP estimator converge to if we decrease the prior variance s^2?

6 Logistic regression vs naive Bayes (Matlab)

For this question, make sure you download the latest version of BLT (8 Oct 08 or newer). Also download `NBLRcode.zip`. Extend your naive Bayes code from hw4 to handle multiple classes. Assume the features are binary. Use the posterior mean estimate of the class-conditional density parameters θ_{jc}, under a Beta(α, α) prior as before. For the class prior $\boldsymbol{\pi}$, compute the MLE. Your interface should be as follows

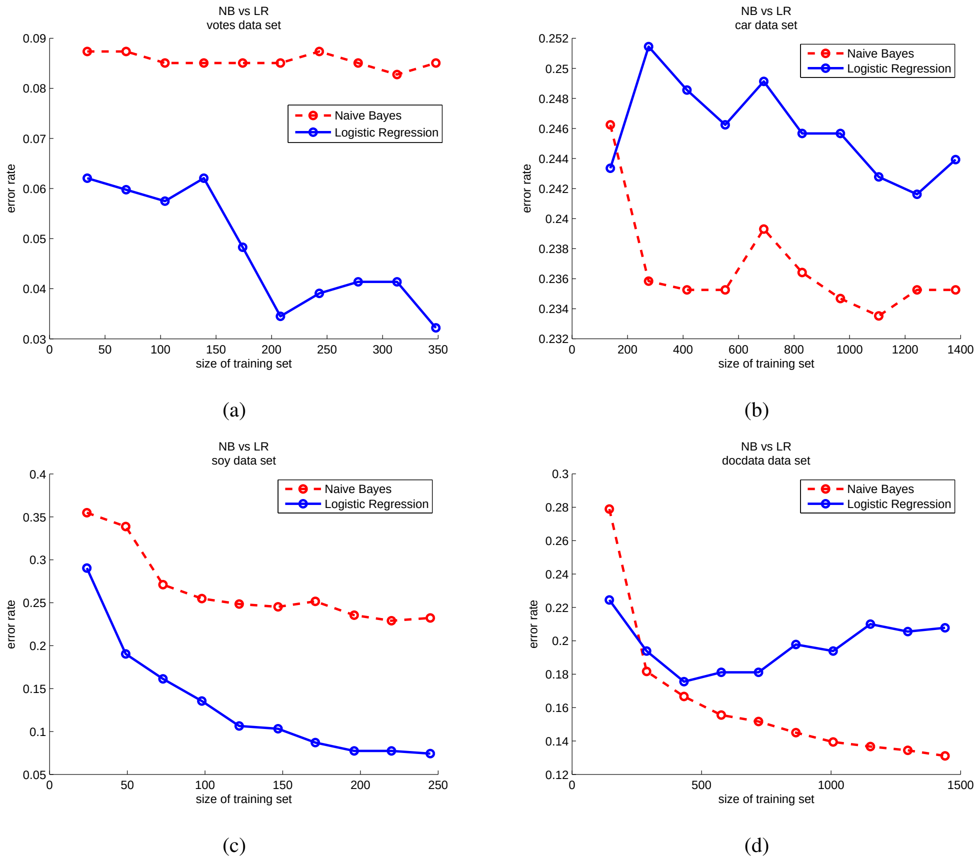

Figure 1: (a) votes $n = 435$, $d = 16$, $C = 2$; (b) car $n = 1728$,$d = 6$, $C = 3$; (c) soy $n = 307$, $d = 35$, $C = 3$; (d) docdata $n = 1800$,$d = 600$,$C = 2$

```
function [theta,classPrior] = NBtrainMulticlass(X,Y,alpha)
% X(i,j) = 0 or 1  i=1:n, j=1:d
% Y(i) = 1,2,3,...C
%
% theta(j,c) = prob of feature j being on in class c
% classPrior(c) = prior prob of class c

function yhat = NBapplyMulticlass(X,theta,classPrior)
% % X(i,j) = 0 or 1  i=1:n, j=1:d
% theta(j,c) = prob of feature j being on in class c
% classPrior(c) = prior prob of class c
%
% yhat(i) = 1 or 2 or .. C (most probable class)
```

Then run `NBLRscript`. You should get the plots shown in Figure 1. Turn in your code and plots.
Bonus (optional): try changing α and/or the L2 regularizer λ in logistic regression, to see what difference it makes (try cross validation). Also, can you explain why the test error increases for logistic regression as the training set increases in size?