1 Introduction

Our intuition for the project comes from the paper "Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit" by Raykar, Yu, etc. The paper analyzed a classification problem where instead of observing a "true" classification of each data point, we observe some classification from several experts. The project will attempt to solve a variation of the problem in which the features are binary instead of real-valued. In addition, we generalized the problem to do a N-class classification instead of a binary classification.

2 Model Description

2.1 Training Data

The data set contains m training examples: \(((\vec{x}^{(i)}, \vec{y}^{(i)}); i = 1, ..., m\}, where \(\vec{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, ... x_k^{(i)})\) with \(x_j^{(i)} \in \{0, 1\}\) and \(\vec{y}^{(i)} \in A^R\) with \(A = \{1, 2, ..., N\}\) representing the N different classes, i.e. the feature space is k-dimensional and there are \(R\) experts providing estimates of the true \(y^{(i)}\).

2.2 Model Assumptions

2.2.1 Naive Bayes Assumption

Similar to the spam classification example given in class, we make the Naive Bayes assumption. Assume that for the \(i\)th training example, the \(x_j^{(i)}\)'s are conditionally independent given the true \(y^{(i)}\). Then we have the following property which is convenient:

\[
p(x_1^{(i)}, x_2^{(i)}, ..., x_k^{(i)} | y^{(i)}) = \prod_{j=1}^{k} p(x_j^{(i)} | y^{(i)}). \quad (1)
\]

2.2.2 Characteristic Matrix for Each Customer

For the \(r\)th expert, we define his/her characteristic matrix to be \(M^{(r)}\), where \(M_{p,q}^{(r)} = P(y_r = q | y = p)\) for \(p, q \in \{1, 2, ..., N\}\) i.e. the entry on the \(p\)th row and \(q\)th column is the probability that the \(r\)th expert gives classification \(q\) given that the true classification is \(p\). Notice that each row of this matrix has to add up to one, hence the degree of freedom is \(N(N - 1)\) instead of \(N^2\), i.e. we should really describe each customer using a \(N \times (N - 1)\) matrix instead of a \(N \times N\) matrix, but for symmetry and simplicity we keep it that way for now.
3 Single Expert Case: A Generalization to Spam Classification

We use two sets of parameters to model this problem:

\[ \phi_y = P(y^{(i)} = y) \]
\[ \phi_{j|y} = P(x_j^{(i)} = 1|y^{(i)} = y). \]

Note that we only consider \( \phi_1, \phi_2, \ldots, \phi_{N-1} \) as parameters, \( \phi_N \) can be calculated as \( \phi_N = 1 - \sum_{p=1}^{N-1} \phi_p. \)

The joint likelihood is:

\[
L(\phi_y, \phi_{j|y}) = \prod_{i=1}^{m} p(x_1^{(i)}, x_2^{(i)}, \ldots, x_k^{(i)}, y^{(i)}) \\
= \prod_{i=1}^{m} p(y^{(i)}) \prod_{j=1}^{k} p(x_j^{(i)}) \\
= \prod_{i=1}^{m} \phi_y^{(i)} \prod_{j=1}^{k} \phi_{j|y}^{(i)} (1 - \phi_{j|y}^{(i)})^{1-x_j^{(i)}}.
\]

Set the partial derivatives of \( L \) to 0 and we derive the maximum likelihood estimators:

\[
\phi_y = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = y\}}{m} \\
\phi_{j|y} = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = y\} x_j^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = y\}}.
\]

4 Multiple Expert Case

4.1 Likelihood Function

We need to use the characteristic matrices \( M^{(r)} \) as well as the parameters used in the single expert case \( (\phi_y, \phi_{j|y}) \). Let \( \Theta = (M^{(r)}, \phi_y, \phi_{j|y}) \). We can calculate the likelihood function:
\[
L(\Theta) = \prod_{i=1}^{m} P(y_{1}^{(i)} , \ldots , y_{R}^{(i)} , x^{(i)}; \Theta) \\
= \prod_{i=1}^{m} \sum_{n=1}^{N} P(y_{1}^{(i)} , \ldots , y_{R}^{(i)} | y^{(i)} = n, x^{(i)}; \Theta) P(x^{(i)}|y^{(i)} = n; \Theta) P(y^{(i)} = n; \Theta) \\
= \prod_{i=1}^{m} \sum_{n=1}^{N} \left( \prod_{r=1}^{R} M_{n,y_r^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_{j|n}^{x_j^{(i)}} (1 - \phi_{j|n}^{y_j^{(i)}}) \right) \phi_n.
\]

However, \( L(\Theta) \) is quite difficult to maximize because of the summation in the formula (and hence taking the log-likelihood does not simplify the problem very much). The solution is to use the EM algorithm with \( \tilde{y} = (y^{(1)} , \ldots , y^{(m)}) \) as latent variables. Now consider the new likelihood function:

\[
L(\tilde{y}, \Theta) = \prod_{i=1}^{m} P(y_{1}^{(i)} , \ldots , y_{R}^{(i)} , x^{(i)} , y^{(i)}; \Theta) \\
= \prod_{i=1}^{m} p(y_{1}^{(i)} , \ldots , y_{R}^{(i)} | y^{(i)} , x^{(i)}; \Theta) p(x^{(i)}|y^{(i)}; \Theta) p(y^{(i)}; \Theta) \\
= \prod_{i=1}^{m} \left( \prod_{r=1}^{R} M_{y_r^{(i)},y_r^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_{j|y_j^{(i)}}^{x_j^{(i)}} (1 - \phi_{j|y_j^{(i)}}) \right) \phi_{y^{(i)}}.
\]

### 4.2 The EM Algorithm

#### 4.2.1 E-step

We need \( Q_i(y^{(i)}) \propto p(y_{1}^{(i)} , \ldots , y_{R}^{(i)} | y^{(i)} , x^{(i)}; \Theta) p(x^{(i)}|y^{(i)}; \Theta) p(y^{(i)}; \Theta) \), and thus

\[
Q_i(y^{(i)}) = \frac{p(y_{1}^{(i)} , \ldots , y_{R}^{(i)} | y^{(i)} , x^{(i)}; \Theta) p(x^{(i)}|y^{(i)}; \Theta) p(y^{(i)}; \Theta)}{\sum_{n=1}^{N} P(y_{1}^{(i)} , \ldots , y_{R}^{(i)} | y^{(i)} = n, x^{(i)}; \Theta) P(x^{(i)}|y^{(i)} = n; \Theta) P(y^{(i)} = n; \Theta)} \\
= \left( \prod_{r=1}^{R} M_{y_r^{(i)},y_r^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_{j|y_j^{(i)}}^{x_j^{(i)}} (1 - \phi_{j|y_j^{(i)}}) \right) \phi_{y^{(i)}} \\
= \frac{\sum_{n=1}^{N} \left( \prod_{r=1}^{R} M_{n,y_r^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_{j|n}^{x_j^{(i)}} (1 - \phi_{j|n}) \right) \phi_n}{\sum_{n=1}^{N} \left( \prod_{r=1}^{R} M_{n,y_r^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_{j|n}^{x_j^{(i)}} (1 - \phi_{j|n}) \right) \phi_n} \\
= \frac{\sum_{n=1}^{N} \left( \prod_{r=1}^{R} M_{n,y_r^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_{j|n}^{x_j^{(i)}} (1 - \phi_{j|n}) \right) \phi_n}{\sum_{n=1}^{N} \left( \prod_{r=1}^{R} M_{n,y_r^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_{j|n}^{x_j^{(i)}} (1 - \phi_{j|n}) \right) \phi_n}.
\]

To initialize (because during the first E-step, we do not have \( \Theta \) to let us calculate \( Q_i(y^{(i)}) \)), we can set \( Q_i(y^{(i)}) = \frac{1}{R} \sum_{r=1}^{R} 1\{y_r^{(i)} = y^{(i)}\} \).
4.2.2 M-step

Given $Q_i(y^{(i)})$ calculated in the E-step, we want to maximize

$$I(\Theta) = \sum_{i=1}^{m} \sum_{n=1}^{N} Q_i(n) \log \left( \left( \prod_{r=1}^{R} M_{n,y^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_{j|n}^{x^{(i)}_j} (1 - \phi_{j|n})^{1-x^{(i)}_j} \right) \phi_n \right)$$

$$= \sum_{i=1}^{m} \sum_{n=1}^{N} Q_i(n) \left( \sum_{r=1}^{R} \log M_{n,y^{(i)}}^{(r)} + \log \phi_n + \sum_{j=1}^{k} (x^{(i)}_j \log \phi_{j|n} + (1 - x^{(i)}_j) \log(1 - \phi_{j|n})) \right)$$

Setting $\frac{\partial I}{\partial M_{n,p}^{(r)}} = 0$ for $p = 1, 2, ..., N - 1$ (bear in mind that $M_{n,N} = 1 - \sum_{p=1}^{N-1} M_{n,p}$ is not a free parameter), we get

$$M_{n,p}^{(r)} = \frac{\sum_{i=1}^{m} Q_i(n) 1\{y_r^{(i)} = p\}}{\sum_{i=1}^{m} Q_i(n)}.$$  \hspace{1cm} (2)

Setting $\frac{\partial I}{\partial \phi_{n}} = 0$ for $n = 1, 2, ..., N - 1$ (again, $\phi_N = 1 - \sum_{n=1}^{N-1} \phi_n$ is not a free parameter), we get

$$\phi_n = \frac{\sum_{i=1}^{m} Q_i(n)}{\sum_{p=1}^{N} \sum_{i=1}^{m} Q_i(p)} = \frac{\sum_{i=1}^{m} Q_i(n)}{m}.$$  \hspace{1cm} (3)

Lastly, we set $\frac{\partial I}{\partial \phi_{j|n}} = \sum_{i=1}^{m} Q_i(n) \left( \frac{x^{(i)}_j}{\phi_{j|n}} - \frac{1-x^{(i)}_j}{1-\phi_{j|n}} \right)$ equal to zero. And we get

$$\phi_{j|n} = \frac{\sum_{i=1}^{m} Q_i(n) x^{(i)}_j}{\sum_{i=1}^{m} Q_i(n)}.$$  \hspace{1cm} (4)

It is worth noting that $\phi_n, \phi_{j|n}$ derived here in the M-step is the same as MLE in the single expert case, except all the $1\{y^{(i)} = n\}$ terms are substituted with $Q_i(n)$.

4.3 Missing Labels

One of the technical details that we need to deal with is the missing labels, meaning that not all experts give classifications to all training examples, i.e. $y_r^{(i)}$ does not necessarily exist for all $i$ and $r$. It turns out that we can make a small change to our algorithm to take care of this issue. Let $R_i$ be the set of experts who classified training example $i$. Then the likelihood function becomes

$$L(\Theta) = \prod_{i=1}^{m} \sum_{n=1}^{N} \left( \prod_{r \in R_i} M_{n,y^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_{j|n}^{x^{(i)}_j} (1 - \phi_{j|n})^{1-x^{(i)}_j} \right) \phi_n.$$
Consequently, the E-step can be modified to

\[ Q_i(y^{(i)}) = \frac{\left( \prod_{r \in R_i} M_{y^{(i)}, y_r^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_j^{x_j^{(i)}} (1 - \phi_{j|y^{(i)}}) \right)}{\sum_{n=1}^{N} \left( \prod_{r \in R_i} M_{n,y_r^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_j^{x_j^{(i)}} (1 - \phi_{j|n}) \right) \phi_n}. \]

with the initial step to be \( Q_i(y^{(i)}) = \frac{1}{|R_i|} \sum_{r \in R_i} 1\{y_r^{(i)} = y^{(i)}\} \).

For the M-step, the update formulae for \( \phi_n \) and \( \phi_{j|n} \) does not change, the formula for \( M_{n,p}^{(r)} \) can be rewritten as

\[ M_{n,p}^{(r)} = \frac{\sum_{i: r \in R_i} Q_i(n) 1\{y_r^{(i)} = p\}}{\sum_{i: r \in R_i} Q_i(n)}. \]

### 4.4 Laplace Smoothing

Another technical detail that we may encounter is that the denominators in the E-step and M-step formulae may be zero. As a result, we need to apply Laplace smoothing. In the M-step, since \( \sum_{i=1}^{n} Q_i(n) \) and \( \sum_{i: r \in R_i} Q_i(n) \) might be zero (consider the case where nobody ever gave a classification of \( n \) in the training set, and the first M-step right after the initial E-step which is to set \( Q_i(y^{(i)}) = \frac{1}{|R_i|} \sum_{r \in R_i} 1\{y_r^{(i)} = y^{(i)}\} \)), the formulae can be replaced with

\[ M_{n,p}^{(r)} = \frac{\sum_{i: r \in R_i} Q_i(n) 1\{y_r^{(i)} = p\} + 1}{\sum_{i: r \in R_i} Q_i(n) + N} \]

\[ \phi_n = \frac{\sum_{i=1}^{m} Q_i(n) + 1}{m + N} \]

\[ \phi_{j|n} = \frac{\sum_{i=1}^{m} Q_i(n) x_j^{(i)} + 1}{\sum_{i=1}^{m} Q_i(n) + 2}. \]

One can also sanity-check that \( \sum_{p=1}^{N} M_{n,p}^{(r)} = 1 \) and \( \sum_{n=1}^{N} \phi_n = 1 \) using the smoothed formulae.

In the E-step, applying Laplace smoothing might be difficult since each

\( \left( \prod_{r \in R_i} M_{y^{(i)}, y_r^{(i)}}^{(r)} \right) \left( \prod_{j=1}^{k} \phi_j^{x_j^{(i)}} (1 - \phi_{j|y^{(i)}}) \right) \phi_{y^{(i)}} \)

term can be very small and it is hard to estimate the order of magnitude of these terms. As a result, adding 1 to the numerator and \( N \) to the denominator, or generally
adding some pre-determined constant \( c \) to the numerator and \( Nc \) to the denominator, may destroy most of the meaningful information in the \( Q_i(y^{(i)}) \) distribution, making them all equal to \( \frac{1}{N} \). However, the good news is that because of the smoothing applied in the M-step, one is guaranteed that \( M^{(r)}_{n,p}, \phi_n, \phi_{jn} \in (0, 1) \) and the denominator will be non-zero, hence there is no need to apply smoothing in the E-step.