Machine Learning 1 - Practice exercise 3

1 Naive Bayes Spam Classification

Naive Bayes (NB) is a particular form of classification that makes strong independence assumptions regarding the features of the data, conditional on the classes (see Bishop section 4.2.3). Specifically, NB assumes each feature is independent given the class label. In contrast, when we looked at probabilistic generative models for classification in the lecture, we used a full-covariance Gaussian to model data from each class, which incorporates correlation between all the input features (i.e. they are not conditionally independent).

If correlated features are treated independently, the evidence for a class will be overcounted. However, Naive Bayes is very simple to construct, because by ignoring correlations the *class-conditional likelihood*, $p(\mathbf{x}|\mathcal{C}_k)$, is a product of D univariate distributions, each of which is simple to learn:

$$p(\mathbf{x}|\mathcal{C}_k) = \prod_{d=1}^{D} p(x_d|\mathcal{C}_k) \tag{1}$$

Consider a spam filter that classifies your emails into three classes $\mathcal{C}_1$ (spam), $\mathcal{C}_2$ (non-spam) and $\mathcal{C}_3$ (ambiguous). If the classifier is not certain whether the email is spam or non-spam, it will assign it to the $\mathcal{C}_3$ (ambiguous) case, indicating it needs help from humans. To do this you first make a *bag-of-words* (BoW) representation of your entire training set (a bunch of spam emails and non-spam emails). A BoW is a vector of dimension D of word counts, one for each document (i.e. the words go into a bag and are shaken, losing their order so only their count matters). You can think of D as the vocabulary size of the training set, but it may also contain tokens or special features you think are important for spam detection. Your training set is therefore consists of an N by D matrix of word counts $\mathbf{X}$, and the target vector $\mathbf{t} = (t_1, ..., t_N)^T$, such that $t_n = 1$, if $n \in \mathcal{C}_1$, and $t_n = 2$, if $n \in \mathcal{C}_2$ and $t_n = 3$ if $n \in \mathcal{C}_3$. Assume we know $p(\mathcal{C}_1) = \pi_1$, $p(\mathcal{C}_2) = \pi_2$ and $p(\mathcal{C}_3) = \pi_3$ (with the constraint $\pi_1 + \pi_2 + \pi_3 = 1$). We can model the word counts using different distributions, but for this question we will use a Poisson distribution model:

$$\begin{aligned} p(x_d|\mathcal{C}_k, \theta_{dk}) &= \mathcal{P}\left(x_d|\lambda_{dk}\right) \\ &= \frac{\lambda_{dk}^{x_d}}{x_d!}\exp(-\lambda_{dk}) \end{aligned}$$

with distribution parameters $\theta_{dk} = \lambda_{dk}$.

With this information answer the following questions:

1. Write down the data likelihood, $p(\mathbf{T}, \mathbf{X}|\boldsymbol{\theta})$, for the *general* three class naive Bayes classifier. Use the indicator function $\mathbb{I}(x = a) = \begin{cases} 1 & \text{if} \quad x = a \\ 0 & \text{otherwise} \end{cases}$. You should write the likelihood in terms of $p(x_d|\mathcal{C}_k, \theta_{dk})$, meaning you should not assume the explicit Poisson distribution.
2. Write down the data likelihood $p(\mathbf{T}, \mathbf{X}|\boldsymbol{\theta})$ for the Poisson model.
3. Write down the log-likelihood for the Poisson model.
4. Solve for the MLE estimators for λ_{dk}.
5. Write $p(\mathcal{C}_1|\mathbf{x})$ for the *general* three class naive Bayes classifier.
6. Write $p(\mathcal{C}_1|\mathbf{x})$ for the Poisson model.
7. For the Poisson model, express the conditions (inequalities) of the region where $\mathbf{x}$ is predicted to be in $\mathcal{C}_1$. Make each inequalities as linear inequalities such as $\mathbf{x}^T\mathbf{a} > c$
8. Is the region where $\mathbf{x}$ is predicted to be in $\mathcal{C}_1$ convex? Why?
9. Give a concrete example with a specific application where it is helpful to make algorithms ask humans' help for ambiguous predictions.

2 Multi-class Logistic Regression

In class we saw the binary classification version of logistic regression. Here you will derive the gradients for the general case $K > 2$. Much of the preliminaries are in Bishop 4.3.4. This will be useful for Lab 2.

For $K > 2$ the posterior probabilities take a generalized form of the sigmoid called the softmax:

$$y_k(\boldsymbol{\phi}) = p(\mathcal{C}_k|\boldsymbol{\phi}) = \frac{\exp(a_k)}{\sum_i \exp(a_i)}$$

where $a_k = \mathbf{w}_k^T \boldsymbol{\phi}$ and $\boldsymbol{\phi}$ is short for $\boldsymbol{\phi}(\mathbf{x}) = (\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), ..., \phi_{M-1}(\mathbf{x}))^T$ with $\phi_0(\mathbf{x}) = 1$. Note that the posterior for class k depends on all the other classes i; keep this in mind when working out the derivatives for $\mathbf{w}_k$. The training set is a pair of matrices $\mathbf{\Phi}$ and $\mathbf{T}$. Each row of $\mathbf{T}$ uses a one-hot encoding of the class labeling for that training example, meaning that the n-th row contains a row vector $\mathbf{t}_n^T$ with all entries zero except for the k-th entry which is equal to 1 if datapoint n belongs to class $\mathcal{C}_k$.

Answer the following questions:

1. Derive $\frac{\partial y_k}{\partial \mathbf{w}_j}$. Bishop uses an indicator function I_{kj}, which you can also view as the element at position (k, j) of the identity matrix; previously we used $\mathbb{I}[k = j]$—they are the same thing.

2. Write down the likelihood as a product over N and K then write down the log-likelihood. Use the entries of $\mathbf{T}$ as selectors of the correct class.

3. Derive the gradient of the log-likelihood with respect to $\mathbf{w}_j$.

4. What is the objective function we minimize that is equivalent to maximizing the log-likelihood?

5. Write a stochastic gradient algorithm for logistic regression using this objective function. Make sure to include indices for time and to define the learning rate. The gradients may differ in sign switching from maximizing to minimizing; don't overlook this.

6. (**Bonus**) In practice, the above vanilla SGD is not effective. Point out a potential weakness of the above algorithm and/or suggest a possible improvement upon it.