Machine Learning 1 - Homework 3

Selene Baez Santamaria

1 Naive Bayes Spam Classification

Answer the following:

1. Write down the likelihood for the general two class naive Bayes classifier.
   Solution:

$$p(\mathbf{T}, \mathbf{X}|\boldsymbol{\theta}) = \prod_{n=1}^{N}(\pi_1 \prod_{d=1}^{D} p(x_n d|\theta_1))^{1-t_n}(\pi_2 \prod_{d=1}^{D} p(x_n d|\theta_2))^{t_n}$$

2. Write down the likelihood for the Poisson model.
   Solution:

$$p(\mathbf{T}, \mathbf{X}|\boldsymbol{\theta}) = \prod_{n=1}^{N}(\pi_1 \prod_{d=1}^{D} \frac{\lambda_{d1}^{x_n d}}{x_n d!}\exp(-\lambda_{d1}))^{1-t_n}(\pi_2 \prod_{d=1}^{D} \frac{\lambda_{d2}^{x_n d}}{x_n d!}\exp(-\lambda_{d2}))^{t_n}$$

3. Write down the log-likelihood for the Poisson model.
   Solution:

$$\ln p(\mathbf{T}, \mathbf{X}|\boldsymbol{\theta}) = \sum_{n \in C_1}^{N} (\ln \pi_1 + \sum_{d=1}^{D} x_{nd} \ln \lambda_{d1} - \ln(x_{nd}!) - \lambda_{d1}) + \sum_{n \in C_2}^{N} (\ln \pi_2 + \sum_{d=1}^{D} x_{nd} \ln \lambda_{d2} - \ln(x_{nd}!) - \lambda_{d2})$$

4. Solve for the MLE estimators for λ_{dk}
   Solution:

$$\frac{\partial \ln p(\mathbf{T}, \mathbf{X}|\boldsymbol{\theta})}{\partial \lambda_{dk}} = \sum_{n \in C_k}^{N} (\frac{x_{nd}}{\lambda_{dk}} - 1) = 0$$
$$\sum_{n \in C_k}^{N} \frac{x_{nd}}{\lambda_{dk}} = \sum_{n \in C_k}^{N} 1$$
$$\lambda_{dk} = \frac{1}{N_k} \sum_{n \in C_k}^{N} x_{nd}$$

   I.e λ_{dk} is the average number of word/token d per email or class k.

5. Write $p(\mathcal{C}_1|\mathbf{x})$ for the general two class naive Bayes classifier.
   Solution:

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{\pi_1 \prod_{d=1}^{D} p(x_{nd}\theta_1)}{\pi_1 \prod_{d=1}^{D} p(x_{nd}\theta_1) + \pi_2 \prod_{d=1}^{D} p(x_{nd}\theta_2)}$$

6. Write $p(\mathcal{C}_1|\mathbf{x})$ for the Poisson model.
   Solution:

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{\pi_1 \prod_{d=1}^{D} \lambda_{d1}^{x_{nd}} \exp(-\lambda_{d1})}{\pi_1 \prod_{d=1}^{D} \lambda_{d1}^{x_{nd}} \exp(-\lambda_{d1}) + \pi_2 \prod_{d=1}^{D} \lambda_{d2}^{x_{nd}} \exp(-\lambda_{d2})}$$

Notice the product over count factorial cancels because it is a constant for each class.

7. Rewrite $p(\mathcal{C}_1|\mathbf{x})$ as a sigmoid $\sigma(a) = \frac{1}{1+\exp(-a)}$; solve for a for the Poisson model.
Solution:

$$
\begin{aligned}
p(\mathcal{C}_1|\mathbf{x}) &= \frac{1}{1 + \frac{\pi_2 \prod_{d=1}^{D} \lambda_{d2}^{x_{nd}} \exp(-\lambda_{d2})}{\pi_1 \prod_{d=1}^{D} \lambda_{d1}^{x_{nd}} \exp(-\lambda_{d1})}} \\
&= \frac{1}{1+\exp(-\alpha)} \\
\alpha &= \underbrace{\sum_{d=1}^{D} x_{nd} \ln \frac{\lambda_{d1}}{\lambda_{d2}}}_{w^T x_n} + \underbrace{\ln \frac{\pi_1}{\pi_2} - \sum_{d=1}^{D} (\lambda_{d1} - \lambda_{d2})}_{w_0}
\end{aligned}
$$

8. Assume $a = \mathbf{w}^T x + w_0$; solve for $\mathbf{w}$ and $w0$.
Solution:

See above. $w_d = \ln \frac{\lambda_{d1}}{\lambda_{d2}}$

9. Is the decision boundary a linear function of $\mathbf{x}$? Why?
Solution:

Yes: α is a linear function of x (w_0 is a constant for all x)

2 Multi-class Logistic Regression

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 $a_k = \mathbf{w}_k^T \boldsymbol{\phi}$

Answer the following:

1. Derive $\frac{\partial y_k}{\partial \mathbf{w}_j}$. Bishop uses an indicator function $\mathbf{I}_{kj}$, entries of the identity matrix; previously we used $[k = j]$ —they are the same thing.
Solution:

$$\begin{aligned}
\frac{\partial y_k(\boldsymbol{\phi})}{\partial \mathbf{w}_j} &= \frac{\partial}{\partial \mathbf{w}_j}\left(\frac{\exp(a_k)}{\sum_i \exp(a_i)}\right) \\
&= \frac{\exp(a_k)}{\sum_i \exp(a_i)}\frac{\partial a_k}{\partial \mathbf{w}_j} - \frac{\exp(a_k)\exp(a_j)}{\left(\sum_i \exp(a_i)\right)^2}\frac{\partial a_k}{\partial \mathbf{w}_j} \\
&= [k=j]\frac{\exp(a_k)}{\sum_i \exp(a_i)}\boldsymbol{\phi} - \frac{\exp(a_k)}{\sum_i \exp(a_i)}\frac{\exp(a_j)}{\sum_i \exp(a_i)}\boldsymbol{\phi} \\
&= \frac{\exp(a_k)}{\sum_i \exp(a_i)}\left([k=j] - \frac{\exp(a_j)}{\sum_i \exp(a_i)}\right)\boldsymbol{\phi} \\
&= y_k(\boldsymbol{\phi})\left(\mathbf{I}_{kj} - y_j(\boldsymbol{\phi})\right)\boldsymbol{\phi}
\end{aligned}$$

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.
Solution:
The likelihood is written as:

$$\begin{aligned}
p(\mathbf{T}|\boldsymbol{\phi}, \mathbf{W}) &= \prod_{n=1}^{N}\prod_{k=1}^{K} p(\mathcal{C}_k|\boldsymbol{\phi}_n)^{t_{nk}} \\
&= \prod_{n=1}^{N}\prod_{k=1}^{K} \left(y_k(\boldsymbol{\phi})\right)^{t_{nk}} \\
&= \prod_{n=1}^{N}\prod_{k=1}^{K} \left(\frac{\exp(a_k)}{\sum_i^K \exp(a_i)}\right)^{t_{nk}}
\end{aligned}$$

The log-likelihood is written as:

$$\ln p(\mathbf{T}|\boldsymbol{\phi}, \mathbf{W}) = \ln \left(\prod_{n=1}^{N} \prod_{k=1}^{K} p(\mathcal{C}_k|\boldsymbol{\phi}_n)^{t_{nk}} \right)$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} (t_{nk} \ln y_k(\boldsymbol{\phi}))$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \left(t_{nk} \ln \frac{\exp(a_k)}{\sum_i^K \exp(a_i)} \right)$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \left(t_{nk} \ln \exp(a_k) - \ln \sum_{i}^{K} \exp(a_i) \right)$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \left(t_{nk} \left(a_k - \ln \sum_{i}^{K} \exp(a_i) \right) \right)$$

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

$$\frac{\partial \ln p(\mathbf{T}|\boldsymbol{\phi}, \mathbf{W})}{\partial \mathbf{w}_j} = \frac{\partial \left(\sum_{n=1}^{N} \sum_{k=1}^{K} \left(t_{nk} \ln y_k(\boldsymbol{\phi}_n) \right) \right)}{\partial \mathbf{w}_j}$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \left(t_{nk} \frac{1}{y_k(\boldsymbol{\phi}_n)} \frac{\partial y_k(\boldsymbol{\phi}_n)}{\partial \mathbf{w}_j} \right)$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \left(\frac{t_{nk}}{y_k(\boldsymbol{\phi}_n)} y_k(\boldsymbol{\phi}_n) \left(\mathbf{I}_{kj} - y_j(\boldsymbol{\phi}_n) \right) \boldsymbol{\phi}_n \right)$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \left(t_{nk} \left(\mathbf{I}_{kj} - y_j(\boldsymbol{\phi}_n) \right) \boldsymbol{\phi}_n \right)$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \left(t_{nk} \mathbf{I}_{kj} \boldsymbol{\phi}_n \right) - \sum_{n=1}^{N} \sum_{k=1}^{K} \left(t_{nk} y_j(\boldsymbol{\phi}_n) \boldsymbol{\phi}_n \right)$$

$$= \sum_{n=1}^{N} \boldsymbol{\phi}_n \sum_{k=1}^{K} \left(t_{nk} \mathbf{I}_{kj} \right) - \sum_{n=1}^{N} y_j(\boldsymbol{\phi}_n) \boldsymbol{\phi}_n \sum_{k=1}^{K} t_{nk}$$

$$= \sum_{n=1}^{N} \boldsymbol{\phi}_n t_{nj} - \sum_{n=1}^{N} y_j(\boldsymbol{\phi}_n) \boldsymbol{\phi}_n$$

$$= \sum_{n=1}^{N} \left(\left(t_{nj} - y_j(\boldsymbol{\phi}_n) \right) \boldsymbol{\phi}_n \right)$$

4. What is the objective function we minimize that is equivalent to maximizing the log-likelihood?
   Solution:
   The objective function is the *cross-entropy error* ($E(\mathbf{W})$) and is equal

to the negative log-likelihood. I.e.:

$$E(\mathbf{W}) = -\ln p(\mathbf{T}|\mathbf{\Phi}, \mathbf{W}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_k(\phi_n)$$

Sometimes we may write y_{nk} for $y_k(\phi_n)$; it is useful sometimes to give clutter-free solutions. Minimizing the cross-entropy requires the same gradients as maximizing the log-likelihood, except there is a change in sign:

$$\begin{aligned}\frac{\partial E(\mathbf{W})}{\partial \mathbf{w}_j} &= \sum_{n=1}^{N}\sum_{k=1}^{K}(y_{nj} - t_{nj})\phi_n \\ &= \sum_{n=1}^{N} e_n\end{aligned}$$

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.
   Solution:
   The single step update for SGD is:

$$w_j^{t+1} = w_j^t - \eta^t \nabla e_n$$

   (a) Initialize $\mathbf{W}$
   (b) Initialize η
   (c) For $t = 1$ to T do:
      (i) Randomly choose n from [1,N]
      (ii) $\mathbf{w}_j = \mathbf{w}_j - \eta \nabla e_n(forall j)$
      (iii) Decrease η
   (d) Return $\mathbf{W}$

6. Explain why is this a stochastic optimization procedure?.
   Solution:
   This is a stochastic procedure because we choose a random data vector n. This causes the full gradient (over all the data vectors) to be replaced by the gradient of the single random vector. Thus, every step introduces noise.

Nonetheless, it is an optimizing procedure because the algorithm is guaranteed to converge to a local minimum, given enough steps ($T \to \infty$) and a process for annealing the learning rate to 0.

7. Logistic regression is not free from overfitting. How would you modify the cross-entropy error to regularize your weights? Write down the new objective. If we optimized for $\mathbf{w}$, would this be the maximum likelihood estimator or the maximum-a-posterior estimator?
Solution:
In order to regularize the cross-entropy error we can use the regularization methods for the log-likelihood. This last one penalizes large values of $\mathbf{W}$ by subtracting the regularization term:

$$\frac{\lambda}{2}\|\mathbf{W}\|^2$$

Since the cross entropy error is the negative log-likelihood, the regularization needs to `ADD` the same term. The new objective is:

$$E(\mathbf{W}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_k(\phi_n) + \frac{\lambda}{2}\|\mathbf{W}\|^2$$

Optimizing the cross entropy for $\mathbf{W}$ is equivalent to optimizing the log-likelihood, which is also the log posterior. As such, optimizing the previous objective can be seen as the maximum-a-posterior estimator.