CS 5350/6350: Machine Learining Fall 2015

Homework 5

Handed out: Nov 20, 2015
Due date: Dec 8, 2015

General Instructions

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down your own solution. Please keep the class collaboration policy in mind.
- Feel free ask the instructor or the TAs questions about the homework.
- Your written solutions should be brief and clear. You need to show your work, not just the final answer, but you do *not* need to write it in glory detail. Your assignment should be **no more than 10 pages**. Every extra page will cost a point.
- Handwritten solutions will not be accepted.
- The homework is due by midnight of the due date. Please submit the homework on Canvas.

1 Naïve Bayes

Consider the Boolean function $f_{TH(3,7)}$. This is a threshold function defined on the 7 dimensional Boolean cube as follows: given an instance x, $f_{TH(3,7)}(x) = 1$ if and only if 3 or more of x's components are 1.

1. [4 points] Show that $f_{TH(3,7)}$ has a linear decision surface over the 7 dimensional Boolean cube.

   Solution: Choose the weight vector to be $\mathbf{w} = [1, 1, 1, 1, 1, 1, 1]^T$ and bias $b = -2.5$. The decision surface is

   $$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 - 2.5 = 0 \tag{1}$$

2. [7 points] Assume that you are given data sampled according to the uniform distribution over the Boolean cube $\{0, 1\}^7$ and labeled according to $f_{TH(3,7)}$. Use naïve Bayes to learn a hypothesis that predicts these labels. What is the hypothesis generated by the naïve Bayes algorithm? (You do not have to implement the algorithm here. You may assume that you have seen all the data required to get accurate estimates of the probabilities).

Solution: Each hypothesis is characterized by a set of probability parameters:

$$p = P(y = 1) \tag{2}$$
$$a_i = P(x_i = 1|y = 1) \tag{3}$$
$$b_i = P(x_i = 1|y = 0) \tag{4}$$

Using naïve Bayes learning we can find such probability parameters from combinatorial calculations since we are assuming we have seen all the data required to get an estimate of the probalities. This leads to the following calculations:

$$p = \sum_{i=3}^{7} \binom{7}{i} / 2^7 = \frac{99}{128} \tag{5}$$

$$a_i = \frac{\sum_{n=3}^{7} \binom{6}{n-1}}{\sum_{n=3}^{7} \binom{7}{n}} = \frac{19}{33} \tag{6}$$

$$b_i = \frac{\sum_{n=1}^{2} \binom{6}{n-1}}{\sum_{n=0}^{2} \binom{7}{n}} = \frac{7}{29} \tag{7}$$

The hypothesis function generated by naive Bayes learning is

$$h(\mathbf{x}) = \text{argmax}_y P(y) \prod_{i=1}^{7} P(x_i|y) \tag{8}$$

Suppose the input $\mathbf{x}$ has n features with value 1, and $7 - n$ features with value 0. We have

$$P(y) \prod_{i=1}^{7} P(x_i|y) = \begin{cases} pa^n(1-a)^{7-n}, & \text{if } y = 1 \\ (1-p)b^n(1-b)^{7-n}, & \text{if } y = 0 \end{cases} \tag{9}$$

The example $\mathbf{x}$ will get label 1 if

$$\frac{pa^n(1-a)^{7-n}}{(1-p)b^n(1-b)^{7-n}} > 1 \tag{10}$$

Substituing $n = 0, 1, 2, \ldots, 7$ we will find that the above inequality holds when $n \geq 2$. So the final prediction of the hypothesis is

$$h(\mathbf{x}) = \begin{cases} 1, & \text{if } n \geq 2, \text{ where } n \text{ is the number of non-zero features} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

3. [4 points] Show that the hypothesis produced in the previous question does not represent this function.

Solution: From previous discussion we find that the learned hypothesis is actually $f_{TH(2,7)}$ instead of $f_{TH(3,7)}$. It cannot represent the original function. If an input $\mathbf{x}$ has 2 non-zeros features, then the learned hypothesis will predict it to be $y = 1$, but the original function will predict $y = 0$.

4. [5 points] Are the naïve Bayes assumptions satisfied by $f_{TH(3,7)}$? Justify your answer.

   Solution: The assumption of naive Bayes is that

$$P(\mathbf{x}|y) = \prod_i P(x_i|y) \tag{12}$$

   This is not satisfied by $f_{TH(3,7)}$. A counter example is for input $\mathbf{x} = [0,0,0,0,0,0,0]$

$$P(\mathbf{x} = [0,0,0,0,0,0,0]|y = 1) = 0 \tag{13}$$

   but

$$\prod_i P(x_i|y = 1) = (1-a)^7 \neq 0. \tag{14}$$

   An intuitive way of thinking is that $y = 1$ requires at least 3 features to be 1. So if for a positive example we only see 2 non-zero features in the first six features, then the 7th feature must be 1. Clearly features are correlated in this case. The naive Bayes assumption is not justified.

2 EM Algorithm

There are two grocery stores in the neighborhood of the U: Smith's and Trader Joe's. Each store has n checkout lanes. The number of customers for each lane per unit time, say one day, is distributed according to Poisson distribution with parameter λ. That is, for the i'th checkout lane,

$$P(\#\text{ of customers for lane } i = x_i|\lambda) = \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$$

with parameters λ_S and λ_T for Smith's and Trader Joe's, respectively.

1. [10 points] Given a record of customer counts $(x_1, \cdots, x_n)$, where x_i denotes the number of customers went through lane i, what is the most likely value of λ?

   Solution: We need tocompute the maximum likelihood estimate for a Poisson distribution. Note that we need to assume that each lane is independent of all others given that the record comes from one store. With this assumption, we have the likelihood of the data in terms of λ as

$$P((x_1, \cdots, x_n)|\lambda) = \prod_i P(x_i|\lambda) = \prod_i \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}.$$

   As usual, we will work with the log likelihood, which is

$$\log P((x_1, \cdots, x_n)|\lambda) = \sum_i \left(x_i \log \lambda - \lambda - \log(x_i!)\right).$$

   Our goal is to find the lambda that maximizes the log likelihood. So we take the derivative of the log likelihood with respect to λ and set it to zero. So we get

$$\begin{aligned}\frac{d}{d\lambda}\log P((x_1,\cdots,x_n)|\lambda) &= 0\\ \frac{1}{\lambda}\sum_i x_i - n &= 0\\ \lambda &= \tfrac{\sum_i x_i}{n}.\end{aligned}$$

2. [10 points] Assume now that you are given a collection of m records $\{(x_{j1},\cdots x_{jn})\}$, where $j = 1,\cdots,m$. You do not know which record is from Smith's and which is from Trader Joe's. Assume that the probability of a record is from the Smith's is η. In other words, it means that the probability that a record is from Trader Joe's is $1-\eta$. Explain the generative model that governs the generation of this data collection. In doing so, name the parameters that are required in order to fully specify the model.

   Solution: This question is simply asking you to explicitly state the probabilistic model that generates the complete data – namely, a labeled record. That is, we need to specify $P((x_{j1},\cdots x_{jn}), y|\eta,\lambda_S,\lambda_T)$, where y is one of Trader Joe's or Smith's. We can write this as:

$$\begin{aligned}P((x_{j1},\cdots x_{jn}), y|\eta,\lambda_S,\lambda_T) &= P(y|\eta,\lambda_S,\lambda_T)P\left((x_{j1},\cdots x_{jn})|y,\eta,\lambda_S,\lambda_T\right)\\ &= P(y||\eta,\lambda_S,\lambda_T)\prod_i P(x_{ji}|y,\eta,\lambda_S,\lambda_T)\\ &= \begin{cases}\eta\prod_i \frac{\lambda_S^{x_i}e^{-\lambda_S}}{x_i!} & \text{if } y = \text{Smith's}\\ (1-\eta)\frac{\lambda_T^{x_i}e^{-\lambda_T}}{x_i!} & \text{if } y = \text{Trader Joe's}\end{cases}\end{aligned}$$

   This can be interpreted with the following generative story: First a Bernoulli random variable with bias η (i.e. a coin toss) is sampled to get the identity of the store. Then, each lane of the store is sampled independently from the Poisson distribution that is parameterized by the identity of the store (i.e either λ_S or λ_T).

3. [10 points] Assume that you are given the parameters of the model described above. How would you use it to cluster records to two groups, the Smith's and the Trader Joe's?

   Solution: We are looking for $P\left(y|(x_1,\cdots x_n),\eta,\lambda_S,\lambda_T\right)$. For brevity, let us use $\mathbf{x}$ to denote the record $(x_1,\cdots x_n)$ and θ to denote the parameters η,λ_S,λ_T. By the definition of conditional probability, we have

$$P\left(y|\mathbf{x},\theta\right) = \frac{P\left(y,\mathbf{x}|\theta\right)}{P\left(y=\text{Smith's},\mathbf{x}|\theta\right)+P\left(y=\text{Trader Joe's},\mathbf{x}|\theta\right)}.$$

   Using the definition of the joint probability from the previous question, we can write

down $P(y = \text{Smith's}|\mathbf{x}, \theta)$ as

$$\begin{aligned} P(y = \text{Smith's}|\mathbf{x}, \theta) &= \frac{\eta \prod_{i=1}^{n} \frac{\lambda_S^{x_i} e^{-\lambda_S}}{x_i!}}{\eta \prod_{i=1}^{n} \frac{\lambda_S^{x_i} e^{-\lambda_S}}{x_i!} + (1-\eta) \prod_{i=1}^{n} \frac{\lambda_T^{x_i} e^{-\lambda_T}}{x_i!}} \\ &= \frac{\eta \prod_i \lambda_S^{x_i} e^{-\lambda_S}}{\eta \prod_i \lambda_S^{x_i} e^{-\lambda_S} + (1-\eta) \prod_i \lambda_T^{x_i} e^{-\lambda_T}}. \end{aligned}$$

Using this, we also get $P(y = \text{Trader Joe's}|\mathbf{x}, \theta)$ as $1 - P(y = \text{Smith's}|\mathbf{x}, \theta)$.

4. [10 points] Given the collection of m records without labels of which store they came from, derive the update rule of the EM algorithm. Show all of your work.

Solution: The answer to the previous question is the E Step, with $Q_j(y) = P(y|\mathbf{x}_j, \theta^t)$. Note that this function Q_j is a function of the previous set of parameters.

We only need to work out the M step to get the full algorithm. The M step requires us to write down the value of the expected log likelihood. Let us first focus on one example $\mathbf{x}_j$

$$\begin{aligned} E_{y \sim Q_j^t}[\log P(y, \mathbf{x}_j|\theta)] &= Q_j(\text{Smith's}) \log P(y = \text{Smith's}, \mathbf{x}_j|\theta) \\ &\quad + Q_j(\text{Trader Joe's}) \log P(y = \text{Trader Joe's}, \mathbf{x}_j|\theta) \\ &= Q_j(\text{Smith's}) \log \eta \prod_{i=1}^{n} \frac{\lambda_S^{x_{ji}} e^{-\lambda_S}}{x_{ji}!} \\ &\quad + Q_j(\text{Trader Joe's}) \log(1-\eta) \prod_{i=1}^{n} \frac{\lambda_T^{x_{ji}} e^{-\lambda_T}}{x_{ji}!} \\ &= Q_j(\text{Smith's}) \left(\log \eta + \sum_{i=1}^{n} (x_{ji} \log \lambda_S - \lambda_S - \log(x_{ji}!)) \right) \\ &\quad + Q_j(\text{Trader Joe's}) \left(\log(1-\eta) + \sum_{i=1}^{n} (x_{ji} \log \lambda_T - \lambda_T - \log(x_{ji}!)) \right) \end{aligned}$$

The expected log likelihood over the data is simply the sum of this expression over the entire data, namely $\sum_j E_{y \sim Q_j^t}[\log P(y, \mathbf{x}_j|\theta)]$. Let us call this sum $f(\theta)$ to simplify notation below.

For the EM update, we need to maximize the expected log likelihood by taking its derivative with respect to each parameter (η, λ_S and λ_T) and setting them to zero.

(a) Update for η:

$$\frac{df}{d\eta} = \sum_j \frac{Q_j(\text{Smith's})}{\eta)} - \frac{Q_j(\text{Trader Joe's})}{1-\eta}$$

Setting this to zero gives us

$$\eta = \frac{\sum_j Q_j(\text{Smith's})}{\sum_j (Q_j(\text{Smith's}) + Q_j(\text{Trader Joe's}))}.$$

(b) Update for λ_S:

$$\frac{df}{d\lambda_S} = \sum_j \left(Q_j(\text{Smith's}) \sum_{i=1}^{n} \left(\frac{x_{ji}}{\lambda_S} - 1 \right) \right)$$

Rearranging and solving for λ_S gives us

$$\lambda_S = \frac{\sum_j Q_j(\text{Smith's}) \sum_{i=1}^{n} x_{ji}}{n \sum_j Q_j(\text{Smith's})}.$$

(c) Update for λ_T: The derivative is very similar to the above case, giving us

$$\lambda_T = \frac{\sum_j Q_j(\text{Trader Joe's}) \sum_{i=1}^{n} x_{ji}}{n \sum_j Q_j(\text{Trader Joe's})}.$$

3 Experiment

We looked maximum a posteriori learning of the logistic regression classifier in class. In particular, we showed that learning the classifier is equivalent to the following optimization problem:

$$\min_{\mathbf{w}} \left\{ \sum_{i=1}^{m} \log(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)) + \frac{1}{\sigma^2} \mathbf{w}^T \mathbf{w} \right\}$$

In this question, you will derive the stochastic gradient descent algorithm for the logistic regression classifier, and also implement it with cross-validation. Detailed instructions on cross-validation procedure can be found in homework 1, and instructions on SGD can be found in homework 4.

1. [5 points] What is the derivative of the function $\log(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i))$ with respect to the weight vector?

   Solution: Without loss of generality, take the derivative with respect to a particular w_j.

   $$\begin{aligned} \frac{d\ell\ell}{dw_j} &= \frac{-y_i x_{ij} \exp(-y_i \mathbf{w} \mathbf{x}_i)}{1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)} \\ &= \frac{-y_i x_{ij}}{1 + \exp(y_i \mathbf{w}^T \mathbf{x}_i)} \end{aligned}$$

   Which implies that:

   $$\frac{d\ell\ell}{d\mathbf{w}} = \frac{-y_i \mathbf{x}_i}{1 + \exp(y_i \mathbf{w}^T \mathbf{x}_i)}$$

2. [5 points] The inner most step in the SGD algorithm is the gradient update where we use a single example instead of the entire dataset to compute the gradient. Write

down the objective where the entire dataset is composed of a single example, say $(\mathbf{x}_i, y_i)$. Derive the gradient of this objective with respect to the weight vector.

Solution:

$$\nabla f = \frac{-y_i \mathbf{x}_i}{1 + \exp(y_i \mathbf{w}^T \mathbf{x}_i)}$$

3. [10 points] Write down the pseudo code for the stochastic gradient algorithm using the gradient from previous part.

   Solution:

Algorithm 1 SGD

Initialize $\mathbf{w} = 0$
for epoch = 1 to T **do**
    Shuffle data
    for each training example $(\mathbf{x}_i, y_i)$ **do**
        update $\mathbf{w} : \mathbf{w} = \mathbf{w} + r \frac{y_i \mathbf{x}_i}{1+\exp(y_i \mathbf{w}^T \mathbf{x}_i)}$
return w

4. [20 points] Implement your pseduo code as a training algorithm with cross-validation on the σ parameter. This parameter basically helps trade off between generalizability and model fit.

   Use the two `astro` data sets (original and scaled) from homework 4 to train and evaluate the learner. In your writeup, please report on the accuracy of your system, what value of sigma you chose based on cross validation, how many epochs you chose to run SGD, and a plot of the NEGATIVE log likelihood after each epoch of SGD.

As mentioned in previous homeworks, you may use any programming language for your implementation. Upload your code along with a script so the TAs can run your solution in the CADE environment.

4 HAPPY HOLIDAYS Extra Credit

25pts You've seen stochastic gradient decent applied to logistic regression. Now we ask why this is a viable strategy for optimizing this objective. Prove that SGD will find the optimal value for this function. This can be done by demonstrating that the objective is convex. There are many ways to prove this. One of the most straightforward ways to show this is demonstrate that the Hessian is positive-semidefninite.

1. [5 ppoins] Find the Gradient of:

$$\sum_{i=1}^{m} \log(1 + \exp(-y_i w^T x_i)) + \frac{1}{\sigma^2} w^T w$$

Solution: The solution is the same as in (3) but because of the linearity of the derivative operator we can take the derivative of $\frac{1}{\sigma^2}w^T w$ separately which results in:

$$\frac{d\ell\ell}{d\mathbf{w}} = \frac{-y_i\mathbf{x}_i}{1+\exp(y_i\mathbf{w}^T\mathbf{x}_i)} + \frac{\mathbf{w}}{\sigma^2}$$

2. [5 points] Find the Hessian of (a).

Solution: We just take the second derivative of (a) with respect to a different index of w for example, w_k

$$\begin{aligned}
\frac{\partial^2\ell\ell}{\partial w_j \partial w_k} &= \frac{\partial\ell\ell}{\partial w_k}\frac{-y_i x_{ij}}{1+\exp(y_i\mathbf{w}^T\mathbf{x}_i)} + \frac{\partial\ell\ell}{\partial w_k}\frac{w_j}{\sigma^2} \\
&= \frac{\partial\ell\ell}{\partial w_k}\frac{-y_i x_{ij}}{1+\exp(y_i\mathbf{w}^T\mathbf{x}_i)} \\
&= \frac{(1+\exp(y_i\mathbf{w}^T\mathbf{x}_i)(0) + y_i x_{ij}\exp(y_i\mathbf{w}^T\mathbf{x}_i)(y_i x_{ik})}{(1+\exp(y_i\mathbf{w}^T\mathbf{x}_i))^2} \\
&= \frac{y_i^2 x_{ij}x_{ik}\exp(y_i\mathbf{w}^T\mathbf{x}_i)}{(1+\exp(y_i\mathbf{w}^T\mathbf{x}_i))^2}
\end{aligned}$$

Therefore the element in the i^{th} row and k^{th} column correspond to the above value.

3. [10 points] prove that the Hessian is positive semidefinite.

Proof. There are a number of ways to prove that this matrix is PSD. We will demonstrate that $z^t H z \geq 0$ where H is the Hessian. To start recognize that for any data point x_i we can create a new constant:

$$\alpha_i = \frac{y_i^2\exp(y_i\mathbf{w}^T\mathbf{x}_i)}{(1+\exp(y_i\mathbf{w}^T\mathbf{x}_i))^2} \geq 0$$

Because these elements do not change as we move through the matrix. Now our hessian has been simplified too:

$$H_{jk} = \alpha_i x_{ij} x_{ik}$$

Now we examine $z^t H z$ for some particular data point (x_i, y_i)

$$\begin{aligned}
z^t H z &= \sum_{j=1}^{d}\sum_{k=1}^{d} z_j z_k H_{jk} \\
&= \sum_{j=1}^{d}\sum_{k=1}^{d} z_j z_k \alpha_i x_{ij} x_{ik} \\
&= \alpha_i z^t x x^t z \\
&= \alpha (z^t x)^2 \\
&\geq 0
\end{aligned}$$

□

4. [5 points] Why does this prove that we are GAURANTEED to have within $\epsilon > 0$ of the right answer?

   Solution: The log likelihood is concave which means that there is only one global maximum therefore our algorithm will converge to some point which is epsilon close to the correct maximum given a sufficient number of iterations.