Machine Learning 1 - Homework 3

Pascal Mattia Esser

September 27th, 2017

1 Naive Bayes Spam Classification

1.1

data likelihood, for the general three class

$$p(\boldsymbol{T}, \boldsymbol{X}|\boldsymbol{\Theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n, C_1)^{\mathbb{I}(t_n=1)} \prod_{n=1}^{N} p(\boldsymbol{x}_n, C_2)^{\mathbb{I}(t_n=2)} \prod_{n=1}^{N} p(\boldsymbol{x}_n, C_3)^{\mathbb{I}(t_n=3)} \tag{1}$$

with

$$p(\boldsymbol{x}|C_k) = \prod_{d=1}^{D} p(\boldsymbol{x}_d|C_k) \tag{2}$$

and

$$p(\boldsymbol{x}_n, C_i) = p(C_i)p(\boldsymbol{x}|C_i) \tag{3}$$

we can rewrite $p(\boldsymbol{T}, \boldsymbol{X}|\boldsymbol{\Theta})$ as:

$$p(\boldsymbol{T}, \boldsymbol{X}|\boldsymbol{\Theta}) = \prod_{k=1}^{K} \prod_{n=1}^{N} \left(p(C_k) \prod_{d=1}^{D} p(x_{nd}|C_k, \Theta_{dk} \right)^{\mathbb{I}(t_n=k)} \tag{4}$$

1.2

data likelihood for the Poisson model

$$p(\boldsymbol{T}, \boldsymbol{X}|\boldsymbol{\Gamma}) = \prod_{k=1}^{K} \prod_{n=1}^{N} \left(p(C_k) \prod_{d=1}^{D} \left(\frac{\lambda_{dk}^{x_{nd}}}{x_{nd}!} exp(-\lambda_{dk}) \right) \right)^{\mathbb{I}(t_n=k)} \tag{5}$$

1.3

log-likelihood for the Poisson model write

$$p(C_k) = \pi_k \tag{6}$$

and use this in the calculation for the log:

$$\ln p(\boldsymbol{T}, \boldsymbol{X}|\boldsymbol{\Gamma}) = \sum_{k=1}^{K}\sum_{n=1}^{N} \mathbb{I}(t_n = k)\left(\ln \pi_k + \sum_{d=1}^{D} x_{nd} \ln \lambda_{dk} - \ln(x_{nd}!) - \lambda_{dk}\right) \tag{7}$$

1.4

Solve for the MLE estimators

$$\frac{\partial \ln p(\boldsymbol{T}, \boldsymbol{X}|\boldsymbol{\Gamma})}{\partial \lambda_{dk}} = \frac{\partial}{\partial \lambda_{dk}} \sum_{k=1}^{K}\sum_{n=1}^{N} \mathbb{I}(t_n = k) \sum_{d=1}^{D} x_{nd} \ln \lambda_{dk} - \lambda_{dk} \tag{8}$$

$$= \sum_{k=1}^{K}\sum_{n=1}^{N} \mathbb{I}(t_n = k) \sum_{d=1}^{D} \frac{x_{nd}}{\lambda_{dk}} - 1 \tag{9}$$

solving for λ_{dk}:

$$\lambda_{dk} = \frac{1}{N_k} \sum_{n=1}^{N} \mathbb{I}(t_n = k) x_{nd} \tag{10}$$

1.5

Write p(C1|x) for the general three class naive Bayes classifier

$$p(C_1|\boldsymbol{x}) = \frac{p(C_1)p(\boldsymbol{x}|C_1)}{\sum_{i=1}^{3} p(C_i)p(\boldsymbol{x}|C_i)} \tag{11}$$

1.6

Write p(C1|x) for the Poisson model

$$p(C_1|\boldsymbol{x}) = \frac{\pi_1 \prod_{d=1}^{D}\left(\frac{\lambda_{d1}^{x_{nd}}}{x_{nd}!} exp(-\lambda_{d1}\right)}{\sum_{k=1}^{3} \pi_k \prod_{d=1}^{D}\left(\frac{\lambda_{dk}^{x_{nd}}}{x_{nd}!} exp(-\lambda_{dk})\right)} \tag{12}$$

1.7

express the conditions(inequalities) of the region where x is predicted to be in C1

The prove for x belonged to C_1 and not to $C_k, k \neq 1$ follows without loss of generality from the prove below for C_1 and C_2.

x belonged to C_1 and not to C_2 iff:

$$p(C_1|\boldsymbol{x}) > p(C_2|\boldsymbol{x}) \tag{13}$$

which means:

$$p(\boldsymbol{x}|C_1)p(C_1) > p(\boldsymbol{x}|C_2)p(C_2) \tag{14}$$

$$p(\boldsymbol{x}|C_1) > p(\boldsymbol{x}|C_2)\frac{p(C_2)}{p(C_1)} \tag{15}$$

$$\prod_{d=1}^{D} \frac{\lambda_{d1}^{x_{nd}}}{x_{nd}!} exp(-\lambda_{d1}) > \prod_{d=1}^{D} \frac{\lambda_{d2}^{x_{nd}}}{x_{nd}!} exp(-\lambda_{d2})\frac{\pi_2}{\pi_1} \tag{16}$$

$$\prod_{d=1}^{D} \left(\frac{\lambda_{d1}}{\lambda_{d2}}\right)^{x_d} > \frac{\pi_2}{\pi_1} \prod_{d=1}^{D} exp(\lambda_{d1} - \lambda_{d2}) \tag{17}$$

$$\sum_{d=1}^{D} \ln \frac{\lambda_{d1}}{\lambda_{d2}} > \ln\left(\frac{\pi_2}{\pi_1}\right) \sum_{d=1}^{D} (\lambda_{d1} - \lambda_{d2}) \tag{18}$$

now the term

$$ln\left(\frac{\pi_2}{\pi_1}\right) \sum_{d=1}^{D} (\lambda_{d1} - \lambda_{d2}) \tag{19}$$

does not depend on directly on x, so we set

$$c_{1,2} = ln\left(\frac{\pi_2}{\pi_1}\right) \sum_{d=1}^{D} (\lambda_{d1} - \lambda_{d2}) \tag{20}$$

and we get for

$$a_{1,2_k} = \ln \frac{\lambda_{d1}}{\lambda_{d2}} \tag{21}$$

write as a matrix

$$\boldsymbol{a}_{12} = \begin{bmatrix} \ln \frac{\lambda_{11}}{\lambda_{12}} \\ \ln \frac{\lambda_{21}}{\lambda_{22}} \\ \vdots \\ \ln \frac{\lambda_{d1}}{\lambda_{d2}} \end{bmatrix} \tag{22}$$

so we can write the in equation

$$\boldsymbol{x}^T \boldsymbol{a}_{12} > c_{12} \tag{23}$$

as

$$\boldsymbol{x}^T \begin{bmatrix} \ln \frac{\lambda_{11}}{\lambda_{12}} \\ \ln \frac{\lambda_{21}}{\lambda_{22}} \\ \vdots \\ \ln \frac{\lambda_{d1}}{\lambda_{d2}} \end{bmatrix} > n\left(\frac{\pi_2}{\pi_1}\right) \sum_{d=1}^{D} (\lambda_{d1} - \lambda_{d2}) \tag{24}$$

1.8

Is the region where x is predicted to be in C1 convex? because we showed in the previous task, that

$$\boldsymbol{x}^T \boldsymbol{a} > c \tag{25}$$

is only linear in $\boldsymbol{x}$, we can show convexity as follows:

$$\hat{\boldsymbol{x}} = \lambda x_1 + (1-\lambda)x_2 \tag{26}$$

with $\lambda \in [0, 1]$

$$p(x_1|C_1)p(C_1) > p(x_1|C_k)p(C_k) \quad (27)$$
$$p(x_2|C_1)p(C_1) > p(x_2|C_k)p(C_k) \quad (28)$$

$$\begin{aligned} x_1^T a_{1k} &= (\lambda x_1 + (1-\lambda)x_2)^T a_{1k} && (29) \\ &= \lambda x_1^T a_{1k} + (1-\lambda)x_2^T a_{1k} && (30) \\ &> \lambda c_{1k} + (1-\lambda)C_{1k} && (31) \\ &= c_{1k} && (32) \end{aligned}$$

giving us:

$$x_1^T a_{1k} > c_{1k} \quad (33)$$

which shows, that the region is convex, because for an arbitrary point $\hat{x}$, it can be shown, that it is on the line between x_1 and x_2.

1.9

Give a concrete example with a specific application where it is helpful to make algorithms ask humans' help for ambiguous predictions.

medical decisions: if the algorithm diagnoses something it would be good to have a doctor double check the results. this is especially important, if the results are with a low certainty. Also the fact, that the misclassification of an algorithm is somethings very different form the on of a human makes it likely, that a human, can spot an error, that the machine would not.

2 Multi-class Logistic Regression

2.1

Derive after w start with

$$y_k(\phi) = p(C_k|\phi) = \frac{exp(a_k)}{\sum exp(a_i)} \quad (34)$$

and use quotient rule to derive:

$$\begin{aligned} \frac{\partial y_k}{\partial \boldsymbol{w}_j} &= \frac{exp(a_k)\frac{\partial a_k}{\partial \boldsymbol{w}_j}(\sum exp(a_i)) - exp(a_k)exp(a_j)\frac{\partial a_i}{\partial \boldsymbol{w}_j}}{(\sum exp(a_i))^2} && (35) \\ &= \frac{exp(a_k)\frac{\partial a_k}{\partial \boldsymbol{w}_i}}{\sum exp(a_i)} - \frac{exp(a_k)exp(a_j)\frac{\partial a_i}{\partial \boldsymbol{w}_j}}{(\sum exp(a_i))^2} && (36) \\ &= \frac{exp(a_k)}{\sum exp(a_i)}\phi^{\mathbb{I}(i=k)} - \frac{exp(a_k)}{\sum exp(a_i)}\frac{exp(a_j)}{\sum exp(a_i)}\frac{\partial a_i}{\partial \boldsymbol{w}_j} && (37) \end{aligned}$$

now with:

$$\frac{exp(a_k)}{\sum exp(a_i)} = y_k(\phi) \quad (38)$$

$$\frac{exp(a_j)}{\sum exp(a_i)} = y_j(\phi) \tag{39}$$

we get

$$\frac{\partial y_k}{\partial \boldsymbol{w}_j} = y_k(\phi)\phi^{\mathbb{I}(j=k)} - y_k(\phi)y_j(\phi)\phi \tag{40}$$

$$= y_k(\phi)(\mathbb{I}(j=k) - y_j(\phi))\phi \tag{41}$$

2.2

likelihood and log-likelihood

$$p(\boldsymbol{T}|\boldsymbol{w},\boldsymbol{\phi}) = \prod_{n=1}^{N}\prod_{k=1}^{K} y_k(\boldsymbol{\phi}_n)^{t_{nk}} \tag{42}$$

log likelihood

$$\ln p(\boldsymbol{T}|\boldsymbol{w},\boldsymbol{\phi}) = \sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\ln y_k(\boldsymbol{\phi}_n) \tag{43}$$

2.3

Derive the gradient with respect to $\boldsymbol{w}_i$

$$\nabla \ln p(\boldsymbol{T}|\boldsymbol{w},\boldsymbol{\phi}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \frac{t_{nk}}{y_k(\boldsymbol{\phi}_n)}\frac{\partial y_k}{\partial \boldsymbol{w}_i} \tag{44}$$

$$= \sum_{n=1}^{N}\sum_{k=1}^{K} \frac{t_{nk}}{y_k(\boldsymbol{\phi}_n)} y_k(\phi)(\mathbb{I}(j=k) - y_{nj}(\phi))\phi \tag{45}$$

$$= \sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}(\mathbb{I}(j=k) - y_{nj}(\phi))\phi) \tag{46}$$

$$= \sum_{n=1}^{N}(t_{nj} - y_{nj}(\phi))\phi \tag{47}$$

2.4

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

The negative logarithm builds the cross-entropy error function as:

$$E(\boldsymbol{w}) = \sum_{n=1}^{N}\sum_{k=1}^{K} - \ln p(\boldsymbol{t}_n|\boldsymbol{w},\boldsymbol{\phi}) \tag{48}$$

$$E(\boldsymbol{w}) = \sum_{n=1}^{N} E_D(\boldsymbol{w}) \tag{49}$$

$$E_D(\boldsymbol{w}) = - \ln p(\boldsymbol{t}_n|\boldsymbol{w},\boldsymbol{\phi}) \tag{50}$$

This function is later used for the stochastic gradient algorithm as:

$$\nabla E_D(\boldsymbol{w}) = -(t_{nj} - y_n j(\phi))\phi \tag{51}$$

2.5

stochastic gradient algorithm for logistic regression using this objective function

Algorithm 1 stochastic gradient algorithm for logistic regression

initialize learning rate η
initialize $\boldsymbol{w}^{(0)}$

for k in K **do**
  while $||\boldsymbol{w}_k^{(\tau-1)} - \boldsymbol{w}_k^{(\tau)}|| > \varepsilon$ **do**
    randomly select $(\boldsymbol{x}_n, t)$
    $\boldsymbol{w}_k^{(\tau+1)} = \boldsymbol{w}_k^{(\tau)} + \eta(t_{nj} - y_j(\phi))\phi$

2.6

potential weakness of above algorithm and/or suggest a possible improvement upon it

if the given data is not linear separable, the algorithm will not converge. this could be solved by stopping after a given number of iterations I_{max}

Algorithm 2 stochastic gradient algorithm for logistic regression with iteration limits

initialize learning rate η
initialize $\boldsymbol{w}^{(0)}$
set the maximum number of Iterations to I_{max}

for k in K **do**
  while $||\boldsymbol{w}^{(\tau-1)} - \boldsymbol{w}^{(\tau)}|| > \varepsilon$ AND $\tau < I_{max}$ **do**
    randomly select $(\boldsymbol{x}_n, t)$
    $\boldsymbol{w}_k^{(\tau+1)} = \boldsymbol{w}_k^{(\tau)} + \eta(t_{nj} - y_j(\phi))\phi$