ML1 Home Assignment 3

id: 12179078

Volodymyr Medentsiy

1

1.1. Data likelihood without independence assumption:

$$p(T, X \mid \theta) = \left[\text{product rule}\right] = p(X \mid T, \theta) \cdot p(T \mid \theta) = \prod_{n=1}^{N} p(t_n \mid \theta).$$

- $$p(x_n \mid t_n, \theta) = \prod_{k=1}^{K} \left(\pi_k \, p(x_n \mid t_n = C_k, \theta)\right)^{\mathbb{1}(t_n = C_k)}$$

Data likelihood with naive assumption:

$$p(X, T \mid \theta) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\pi_i \, p(x_n \mid t_n = C_k, \theta_k)\right)^{\mathbb{1}(t_n = C_k)} = \left[\text{naive Bayes assumption}\right] =$$

$$= \prod_{n=1}^{N} \prod_{k=1}^{K} \prod_{d=1}^{D} \left(\pi_k \cdot p(x_{nd} \mid t_n = C_k, \theta_{dk})\right)^{\mathbb{1}(t_n = C_k)}$$

1.2.

Number of parameters before NB assumption: $K \times (2^D - 1)$

(For each class there is 2^D possible combinations of the features, but the probability of the last combination could be derived from all others $\Rightarrow 2^D - 1$ parameters.)

Number of parameters after NB assumption: $K \times D$

(For each class there are D features, and each of them has its own probability to occur $\Rightarrow D$ parameters.)

This assumption is called naive because it usually does not hold in real life, but conditional independence makes training the model much easier.

Example: in text classification problem we assume that words are independent of a context which obviously does not hold in natural language.

1.3.

$$\ln p(T, X \mid \theta) = \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{d=1}^{D} \mathbb{1}(t_n = C_k) \left(\ln \pi_k + \ln p(x_{nd} \mid t_n = C_k, \theta_{dk})\right) =$$

$$= \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{d=1}^{D} \mathbb{1}(t_n = C_k) \cdot \left(\ln \pi_k + x_{nd} \ln \theta_{dk} + (1 - x_{nd}) \ln (1 - \theta_{dk})\right).$$

1.4. θ_{dk}^{MLE} :

$$\frac{d}{d\theta_{dk}} \ln p(X, T|\theta) = \sum_{n \in C_k}^{N} x_{nd} \frac{1}{\theta_{dk}} - \frac{(1-x_{nd})}{1-\theta_{dk}} = 0 \Rightarrow$$

$$\Rightarrow (1-\theta_{dk}) \sum_{n \in C_k}^{N} x_{nd} - \theta_{dk} \sum_{n \in C_k}^{N} (1-x_{nd}) = 0 \Rightarrow$$

$$\Rightarrow \theta_{dk} \left(- \sum_{n \in C_k}^{N} x_{nd} - \sum_{n \in C_k}^{N} (1-x_{nd}) \right) = - \sum_{n \in C_k}^{N} x_{nd} \Rightarrow$$

$$\Rightarrow \tilde{\theta}_{dk} = \frac{\sum_{n \in C_k}^{N} x_{nd}}{\sum_{n \in C_k}^{N} (x_{nd} + 1 - x_{nd})} = \frac{\sum_{n \in C_k}^{N} x_{nd}}{|C_k|}, \text{ with } |C_k| \text{ - number of documents in class } k.$$

$$\frac{d^2}{d\theta_{dk}^2} \ln p(X, T|\theta) = - \frac{1}{\theta_{dk}^2} \sum_{n \in C_k}^{N} x_{nd} - \frac{1}{(1-\theta_{dk})^2} \sum_{n \in C_k}^{N} (1-x_{nd}) \Bigg|_{\theta_{dk} = \tilde{\theta}_{dk}} =$$

$$= - \frac{(|C_k|)^2}{\sum_{n \in C_k}^{N} x_{nd}} - \frac{\sum_{n \in C_k}^{N} (1-x_{nd})}{\frac{(|C_k| - \sum_{nd}^{N} x_{nd})^2}{|C_k|^2}} = - \frac{|C_k|^2}{\sum_{n \in C_k}^{N} x_{nd}} - \frac{|C_k|^2}{\sum_{n \in C_k}^{N} 1 - x_{nd}} < 0.$$

$$\Rightarrow \theta_{dk}^{MLE} = \frac{\sum_{n \in C_k}^{N} x_{nd}}{|C_k|}.$$

θ_{dk}^{MLE} could be interpreted as the average number of word d per document in class K.

1.5. $p(C_1|x)$ for general K classes naive Bayes classifier, $x \in \mathbb{R}^D$

$$p(C_1|x) = \frac{p(x|C_1)\cdot p(C_1)}{p(x)} = \frac{\pi_1 \prod_{d=1}^{D} p(x_d|C_1)}{\sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} p(x_d|\theta_{dk})}$$

1.6. $p(C_1|x)$ for Bernoulli case.

$$p(C_1|x) = \frac{\pi_1 \prod_{d=1}^{D} \theta_{d1}^{x_d} \cdot (1-\theta_{d1})^{1-x_d}}{\sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} \theta_{dk}^{x_d} (1-\theta_{dk})^{1-x_d}}$$

1.7.

$$x \in C_1 \iff p(C_1|x) > p(C_k|x) \quad \forall k \neq 1.$$

$$\pi_1 \prod_{d=1}^{D} \theta_{d1}^{x_d} (1-\theta_{d1})^{1-x_d} > \pi_k \prod_{d=1}^{D} \theta_{dk}^{x_d} (1-\theta_{dk})^{1-x_d}$$

$$\prod_{d=1}^{D} \left(\frac{\theta_{d1}}{\theta_{dk}}\right)^{x_d} \left(\frac{1-\theta_{d1}}{1-\theta_{dk}}\right)^{1-x_d} > \frac{\pi_k}{\pi_1} \implies$$

$$\implies \sum_{d=1}^{D} x_d \ln \frac{\theta_{d1}}{\theta_{dk}} + (1-x_d) \ln\left(\frac{1-\theta_{d1}}{1-\theta_{dk}}\right) > \ln \frac{\pi_k}{\pi_1} \implies$$

$$\implies \sum_{d=1}^{D} x_d \left(\ln \frac{\theta_{d1}}{\theta_{dk}} - \ln\left(\frac{1-\theta_{d1}}{1-\theta_{dk}}\right)\right) > \ln \frac{\pi_k}{\pi_1} - \sum_{d=1}^{D} \ln\left(\frac{1-\theta_{d1}}{1-\theta_{dk}}\right) \implies$$

$$\implies x^T a_k > c_k \quad \forall k \neq 1 \quad \text{with} \quad x^T = (x_1, \dots, x_D),$$

$$a_k = \left(\ln\left(\frac{\theta_{11}}{\theta_{1k}} \cdot \frac{1-\theta_{1k}}{1-\theta_{11}}\right), \dots, \ln\left(\frac{\theta_{D1}}{\theta_{Dk}} \cdot \frac{1-\theta_{Dk}}{1-\theta_{D1}}\right) \right)^T$$

$$c_k = \ln\frac{\pi_k}{\pi_1} - \sum_{d=1}^{D} \ln\left(\frac{1-\theta_{d1}}{1-\theta_{dk}}\right)$$

1.8. Yes, it is possible to consider feature vectors with continuous values. Then those features should be described by continuous random variables. One of the possible example is a Gaussian Bayes classifier, where $p(x \mid C_k) = N(x \mid \mu_k, \delta^2_k)$.
In the document classification task continuous feature could represent the frequency of occurrence of the words in a class and then it could be modelled with a triangular distribution.

№2

2.1.

1) $$p(T \mid \varphi, w_1, \dots, w_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(C_k \mid \varphi_n)^{t_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} y_k(\varphi_n)^{t_{nk}}$$

2) $$\ln p(T \mid \varphi, w_1, \dots, w_K) = \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(\varphi_n).$$

3) $$\frac{\partial y_k(\varphi_n)}{\partial w_j}$$

$$j \neq k: \quad \frac{\partial y_k(\varphi_n)}{\partial w_j} = \exp(w_k^T \varphi_n)\left(-\sum_i \exp(w_i^T \varphi_n)\right)^{-2} \exp(w_j^T \varphi_n) \cdot \varphi_n^T =$$

$$= -\frac{\exp(w_k^T \varphi_n)}{\sum_i \exp(w_i^T \varphi_n)} \cdot \frac{\exp(w_j^T \varphi_n)}{\sum_i \exp(w_i^T \varphi)} \cdot \varphi_n^T = -y_k(\varphi_n) \cdot y_j(\varphi_n) \cdot \varphi_n^T.$$

$$j = k \quad \frac{\partial y_k(\varphi_n)}{\partial w_j} = \frac{\exp(w_k^T \varphi_n)}{\sum_i \exp(w_i^T \varphi_n)} \cdot \varphi_n^T - \frac{\exp(w_k^T \varphi_n) \cdot \exp(w_k^T \varphi_n)}{\left(\sum_i \exp(w_i^T \varphi_n)\right)^2} \cdot \varphi_n^T =$$

$= y_k(\varphi_n)\,(1 - y_k(\varphi_n))\,\varphi_n^T$

So in general case: $\frac{\partial y_k(\varphi_n)}{\partial w_j} = y_k(\varphi_n)\,(I_{kj} - y_j(\varphi_n)) \cdot \varphi_n^T$,

with $I_{kj} = \begin{cases} 1, & k = j \\ 0, & k \neq j. \end{cases}$

4) $\nabla_{w_j} \ln p(T \mid \Phi, w_1, \dots, w_K) = \left[y_{nk} := y_k(\varphi_n) \right] = \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \frac{\partial \ln y_{nk}}{\partial w_j} =$

$= \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{t_{nk}}{y_{nk}} \frac{\partial y_{nk}}{\partial w_j} = \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{t_{nk}}{y_{nk}} \cdot y_{nk}(I_{kj} - y_{nj})\varphi_n^T =$

$= \sum_{n=1}^{N} \varphi_n^T \cdot \left(\sum_{k=1}^{K} t_{nk} \cdot I_{kj} - y_{nj} \underbrace{\sum_{k=1}^{K} t_{nk}}_{\| \; 1} \right) =$

$= \sum_{n=1}^{N} \varphi_n^T \cdot (t_{nj} - y_{nj})$

5) $\nabla_{w_j} \ln p(T \mid \Phi, w_1, \dots, w_K) = d_j^T \Phi$

with $d_j = (t_{1j} - y_{1j}, \dots, t_{nj} - y_{nj})^T$,

(if $W \in \mathbb{R}^{M \times K}$ - weight matrix, then we can also consider:

$\nabla_W \ln p(T \mid \Phi, w_1, \dots, w_K) = (T - Y)^T \Phi \in \mathbb{R}^{K \times M}$)

It is useful because matrix multiplications are more computationally efficient.

2.2.

$$E = -\ln p(T|q, w_1, \dots, w_K) = \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(q_n).$$

E – is a cross-entropy.

The relationship between E and log-likelihood: Instead of maximizing log-likelihood we need to minimize E.

What changes in terms of weight update?

Because we minimize E and maximize log-likelihood:

$w^{\tau+1} = w^{\tau} + \eta \cdot \nabla_w (\text{log-likelihood})^T$, but

$w^{\tau+1} = w^{\tau} - \eta \nabla_w E^T$

2.3.

B – the batch size, n_B – number of batches.

1. Initialize: η – learning rate, $w_1, \dots, w_K$ – weights
2. For $\tau = 1$ to n_B
   i. Randomly sample a minibatch M of size B.
   ii. $w_j = w_j - \eta \left(\nabla_{w_j} \sum_{i \in M} E_i\right)^T \quad \forall j = \overline{1, K}$

   with $E_i = -\sum_{k=1}^{K} t_{ik} \ln y_k(q_i)$

   iii. (optional) decrease η.
3. Return $w_1, \dots, w_K$.

Advantage in comparison with SGD with single data point: the estimate of the gradient is less noisy.

Comparison with full batch gradient descent:

Though full batch provide a more accurate estimate of the gradient, it will be more computationally efficient to use mini-batch GD. Some argue that small batches can also provide a regularizing effect.

2.4. Multiclass logistic regression:

$$y_j = \frac{\exp(\tilde{w}_j^T \varphi(x))}{\sum_{i=1}^{3} \exp(\tilde{w}_i^T \varphi(x))}$$

$\tilde{w}_1 = (w_5 \quad w_6)^T$, $\tilde{w}_2 = (w_7 \quad w_8)^T$, $\tilde{w}_3 = (w_9 \quad w_{10})^T$

The activation function on the last layer should be softmax.

$$\varphi(x) = (f_1(w_1 x_1 + w_2 x_2) \quad f_2(w_3 x_1 + w_4 x_2))^T,$$

Because $\varphi(x)$ - fixed basis function $\Rightarrow$ w_1, w_2, w_3, w_4 - should be fixed and non-zero, but activation functions f_1, f_2 — could be of any choice.

(Note:

However if we need a bias component $\varphi_1(x_1, x_2) = 1$, then as an activation function f_1 we can use $f_1(x_1, x_2) \equiv 1$.)

2.5. $\vec{y} = \text{Softmax}(W_2 \cdot \text{ReLU}(W_1 \vec{x}))$

1. Forward step. $x = (0{,}3 \quad 0{,}7)^T$ $y = (0 \quad 0 \quad 1)^T$

$$W_1 x = \begin{pmatrix} 0{,}4 & 0{,}87 \\ 0{,}58 & 0{,}34 \end{pmatrix} \begin{pmatrix} 0{,}3 \\ 0{,}7 \end{pmatrix} = \begin{pmatrix} 729/1000 \\ 103/250 \end{pmatrix} = \begin{pmatrix} 0{,}729 \\ 0{,}412 \end{pmatrix}$$

$$\text{ReLU}\begin{pmatrix} 729/1000 \\ 103/250 \end{pmatrix} = \begin{pmatrix} 0{,}729 \\ 0{,}412 \end{pmatrix}$$

$$W_2 \,\text{ReLU}(W_1 x) = \begin{pmatrix} 0{,}12 & 0{,}87 \\ 0{,}82 & 0{,}31 \\ 0{,}77 & 0{,}9 \end{pmatrix} \begin{pmatrix} 0{,}729 \\ 0{,}412 \end{pmatrix} = \begin{pmatrix} 2787/6250 \\ 1451/2000 \\ 93213/100000 \end{pmatrix} \approx$$

$$\approx \begin{pmatrix} 0{,}446 \\ 0{,}726 \\ 0{,}932 \end{pmatrix}$$

$$\text{Softmax}(W_2\,\text{ReLU}(W_1 x)) = \begin{bmatrix} e^{0,446} + e^{0,726} + e^{0,932} \approx \\ \approx 6,168 \end{bmatrix} =$$

$$= \begin{pmatrix} e^{0,446}/6,168 \\ e^{0,726}/6,168 \\ e^{0,932}/6,168 \end{pmatrix} \approx \begin{pmatrix} 0,253 \\ 0,335 \\ 0,412 \end{pmatrix}$$

The loss: $E = -\ln 0,412 \approx 0,887$

2) Compute the derivative:

$$\frac{dE}{dw_5} = \frac{d}{dw_5}\left(-(t_1 \ln y_1(p) + t_2 \ln y_2(p) + t_3 \ln y_3(p))\right) =$$

$$= \frac{d}{dw_5}\Big\{ -\Big(t_1 \cdot (w_5 p_1 + w_6 p_2) + t_2 (w_7 p_1 + w_8 p_2) + t_3 (w_9 p_1 + w_{10} p_2) -$$

$$- \underbrace{\sum_{i=1}^{3} t_i}_{=1} \ln\left(\exp(w_5 p_1 + w_6 p_2) + \exp(w_7 p_1 + w_8 p_2) + \exp(w_9 p_1 + w_{10} p_2)\right)\Big) =$$

$$= -t_1 p_1 + \frac{\exp(w_5 p_1 + w_6 p_2) \cdot p_1}{\exp(w_5 p_1 + w_6 p_2) + \exp(w_7 p_1 + w_8 p_2) + \exp(w_9 p_1 + w_{10} p_2)},$$

with $p_1 = \text{ReLU}(w_1 x_1 + w_2 x_2) = 0,729$, $p_2 = \text{ReLU}(w_3 x_1 + w_4 x_2) = 0,412$.

$$\frac{dE}{dw_5} = 0,729 \cdot 0,253 \approx 0,184.$$

3)

$$w_i = w_i - 0,05 \cdot \frac{\partial E}{\partial w_i}$$

$$w_1 = 0,4 + 0,05 \cdot 0,044 \approx 0,402$$

$$w_2 = 0,87 + 0,05 \cdot 0,103 \approx 0,875$$

$$w_3 = 0,58 + 0,05 \cdot 0,062 \approx 0,583$$

$w_4 = 0{,}34 + 0{,}05 \cdot 0{,}144 \approx 0{,}347$

$w_5 = 0{,}12 - 0{,}05 \cdot 0{,}184 \approx 0{,}111$

$w_6 = 0{,}87 - 0{,}05 \cdot 0{,}104 \approx 0{,}865$

$w_7 = 0{,}82 - 0{,}05 \cdot 0{,}144 \approx 0{,}808$

$w_8 = 0{,}31 - 0{,}05 \cdot 0{,}138 \approx 0{,}303$

$w_9 = 0{,}77 + 0{,}05 \cdot 0{,}429 \approx 0{,}791$

$w_{10} = 0{,}9 + 0{,}05 \cdot 0{,}242 \approx 0{,}912$

4) $W_1 x = \begin{pmatrix} 0{,}402 & 0{,}875 \\ 0{,}583 & 0{,}347 \end{pmatrix} \begin{pmatrix} 0{,}3 \\ 0{,}7 \end{pmatrix} = \begin{pmatrix} 7331/10000 \\ 2089/5000 \end{pmatrix} \approx \begin{pmatrix} 0{,}733 \\ 0{,}418 \end{pmatrix}$

$ReLU(W_1 x) = \begin{pmatrix} 0{,}733 \\ 0{,}418 \end{pmatrix}$

$W_2\, ReLU(W_1 x) = \begin{pmatrix} 0{,}111 & 0{,}865 \\ 0{,}808 & 0{,}303 \\ 0{,}791 & 0{,}912 \end{pmatrix} \begin{pmatrix} 0{,}733 \\ 0{,}418 \end{pmatrix} \approx \begin{pmatrix} 0{,}443 \\ 0{,}719 \\ 0{,}961 \end{pmatrix}$

$Softmax(W_2\, ReLU(W_1 x)) = \left[e^{0{,}443} + e^{0{,}719} + e^{0{,}961} \approx 6{,}224 \right] =$

$= \begin{pmatrix} e^{0{,}443}/6{,}224 \\ e^{0{,}719}/6{,}224 \\ e^{0{,}961}/6{,}224 \end{pmatrix} \approx \begin{pmatrix} 0{,}25 \\ 0{,}33 \\ 0{,}42 \end{pmatrix}$

Loss: $E = -\ln(0{,}42) \approx 0{,}868 \Rightarrow$ the loss decreased.