Machine Learning Exercise Sheet 05

Linear Classification

Homework

Linear classification

Problem 1: We want to create a generative binary classification model for classifying *nonnegative* one-dimensional data. This means, that the labels are binary ($y \in \{0, 1\}$) and the samples are $x \in [0, \infty)$.

We place a uniform prior on y

$$p(y = 0) = p(y = 1) = \frac{1}{2}.$$

As our samples x are nonnegative, we use exponential distributions (and not Gaussians) as class conditionals:

$$p(x \mid y = 0) = \text{Expo}(x \mid \lambda_0) \qquad \text{and} \qquad p(x \mid y = 1) = \text{Expo}(x \mid \lambda_1),$$

where $\lambda_0 \neq \lambda_1$. Assume, that the parameters λ_0 and λ_1 are known and fixed.

a) What is the name of the posterior distribution $p(y \mid x)$? You only need to provide the name of the distribution (e.g., "normal", "gamma", etc.), not estimate its parameters.

Bernoulli.

Remark: y can only take values in $\{0, 1\}$, so obviously Bernoulli is the only possible answer.

b) What values of x are classified as class 1?
(As usual, we assume that the classification decision is $y_{predicted} = \arg\max_k p(y = k \mid x)$)

Sample x is classified as class 1 if $p(y = 1 \mid x) > p(y = 0 \mid x)$. This is the same as saying

$$\frac{p(y = 1 \mid x)}{p(y = 0 \mid x)} \overset{!}{>} 1 \quad \text{or equivalently} \quad \log \frac{p(y = 1 \mid x)}{p(y = 0 \mid x)} \overset{!}{>} 0.$$

Upload a single PDF file with your homework solution to Moodle by 17.11.2019, 23:59 CET. We recommend to typeset your solution (using LaTeX or Word), but handwritten solutions are also accepted. If your handwritten solution is illegible, it won't be graded and you waive your right to dispute that.

$$\begin{aligned}
\log \frac{p(y=1 \mid x)}{p(y=0 \mid x)} &= \log \frac{p(x \mid y=1)p(y=1)}{p(x \mid y=0)p(y=0)} \\
&= \log \frac{p(x \mid y=1)}{p(x \mid y=0)} \\
&= \log \frac{\lambda_1 \exp(-\lambda_1 x)}{\lambda_0 \exp(-\lambda_0 x)} \\
&= \log \frac{\lambda_1}{\lambda_0} + \lambda_0 x - \lambda_1 x = \log \frac{\lambda_1}{\lambda_0} + (\lambda_0 - \lambda_1)x
\end{aligned}$$

The inequality that we are interested in solving is

$$\begin{aligned}
(\lambda_0 - \lambda_1)x + \log \frac{\lambda_1}{\lambda_0} &> 0 \\
(\lambda_0 - \lambda_1)x &> \log \frac{\lambda_0}{\lambda_1}
\end{aligned}$$

We have to be careful, because if $(\lambda_0 - \lambda_1) < 0$, diving by it will flip the inequality sign. Hence the answer is

$$\begin{cases}
x \in \left(\frac{\log \lambda_0 - \log \lambda_1}{\lambda_0 - \lambda_1}, \infty\right) & \text{if} \quad \lambda_0 > \lambda_1; \\
x \in \left[0, \frac{\log \lambda_0 - \log \lambda_1}{\lambda_0 - \lambda_1}\right) & \text{if} \quad \lambda_0 < \lambda_1.
\end{cases}$$

Problem 2: Assume you have a linearly separable data set. What properties does the maximum likelihood solution for the decision boundary $\boldsymbol{w}$ of a logistic regression model have? Assume that $\boldsymbol{w}$ includes the bias term.

What is the problem here and how do we prevent it?

Finding a separating hyperplane puts all training points on the correct side (and thus ensures posterior class probabilities > 0.5 for each training point). Taking the magnitude of $\boldsymbol{w}$ to infinity then assigns each training point a posterior class probability of 1. This happens because of the shape of the logistic sigmoid function—it becomes a heaviside (step) function in this particular case.

We can prevent this by adding a weight regularization term to the loss function, which will make the loss function bounded.

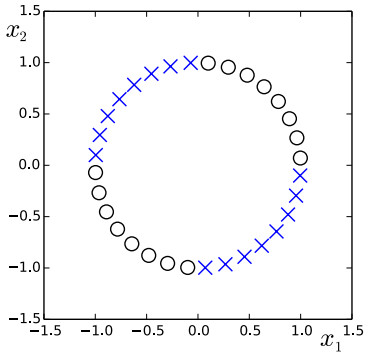
Problem 3: Show that the softmax function is equivalent to a sigmoid in the 2-class case.

$$\begin{aligned}\frac{\exp(\boldsymbol{w}_1^T\boldsymbol{x})}{\exp(\boldsymbol{w}_1^T\boldsymbol{x}) + \exp(\boldsymbol{w}_0^T\boldsymbol{x})} &= \frac{1}{1 + \exp(\boldsymbol{w}_0^T\boldsymbol{x})/\exp(\boldsymbol{w}_1^T\boldsymbol{x})} \\ &= \frac{1}{1 + \exp(\boldsymbol{w}_0^T\boldsymbol{x} - \boldsymbol{w}_1^T\boldsymbol{x})} \\ &= \frac{1}{1 + \exp(-(\boldsymbol{w}_1 - \boldsymbol{w}_0)^T\boldsymbol{x})} \\ &= \sigma(\hat{\boldsymbol{w}}^T\boldsymbol{x})\end{aligned}$$

where $\hat{\boldsymbol{w}} = \boldsymbol{w}_1 - \boldsymbol{w}_0$.

One conclusion we can draw from this is that if we have C parameter vectors $\boldsymbol{w}_c$ for C classes, the logistic regression model is unidentifiable. This means that adding the same constant $k \in \mathbb{R}$ to each vector $\boldsymbol{w}_c := \boldsymbol{w}_c + k$ would lead to the same logistic regression model. We can fix this issue by adding a constraint $\boldsymbol{w}_1 = \mathbf{0}$, which is what is done implicitly when we use sigmoid (instead of 2-class softmax) in binary classification.

Problem 4: Which basis function $\phi(x_1, x_2)$ makes the data in the example below linearly separable (crosses in one class, circles in the other)?



There are many choices of basis functions to achieve this goal. One example is

$$\phi(x_1, x_2) = x_1 x_2$$

In-class Exercises

Multi-Class Classification

Problem 5: Consider a generative classification model for C classes defined by prior class probabilities $p(y = c) = \pi_c$ and general class-conditional densities $p(\boldsymbol{x}|y = c, \boldsymbol{\theta}_c)$ where $\boldsymbol{x}$ is the input feature vector and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_c\}_{c=1}^C$ are further model parameters. Suppose we are given a training set $\mathcal{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^N$ where $y^{(n)}$ is a binary target vector of length C that uses the 1-of-C(one-hot) encoding scheme, so that it has components $y_c^{(n)} = \delta_{ck}$ if pattern n is from class $y = k$. Assuming that the data points are iid, show that the maximum-likelihood solution for the prior probabilities is given by

$$\pi_c = \frac{N_c}{N}$$

where N_c is the number of data points assigned to class $y = c$.

The likelihood function of the parameters $\{\pi_c, \boldsymbol{\theta}_c\}_{c=1}^C$ is given by

$$p(\mathcal{D}|\{\pi_c, \boldsymbol{\theta}_c\}_{c=1}^C) = \prod_{n=1}^N \prod_{c=1}^C (p(\boldsymbol{x}^{(n)}|\boldsymbol{\theta}_c)\pi_c)^{y_c^{(n)}}$$

so the log-likelihood is given by

$$\log p(\mathcal{D}|\{\pi_c, \boldsymbol{\theta}_c\}_{c=1}^C) = \sum_{n=1}^N \sum_{c=1}^C y_c^{(n)} \log \pi_c + \text{const in } \pi_c$$

In order to maximize the log likelihood with respect to π_c we need to preserve the constraint $\sum_c \pi_c = 1$. This can be done by introducing a Lagrange multiplier λ and maximizing

$$\sum_{n=1}^N \sum_{c=1}^C y_c^{(n)} \log \pi_c + \lambda \left(\sum_{c=1}^C \pi_c - 1 \right).$$

Setting the derivative with respect to π_c equal to zero, we obtain

$$\lambda = \frac{1}{\pi_c} \sum_{n=1}^N y_c^{(n)} = \frac{1}{\pi_c} N_c.$$

Setting the derivative with respect to λ equal to zero, we obtain our constraint

$$\sum_{c=1}^C \pi_c = 1.$$

where we can now insert the previous result $\pi_c = \frac{N_c}{\lambda}$ and obtain

$$\lambda = \sum_c N_c = N.$$

So overall we obtain

$$\pi_c = \frac{N_c}{N}.$$

Problem 6: Using the same classification model as in the previous question, now suppose that the class-conditional densities are given by Gaussian distributions with a shared covariance matrix, so that

$$p(x|y = c, \boldsymbol{\theta}_c) = p(x|\boldsymbol{\theta}_c) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}).$$

Show that the maximum likelihood solution for the mean of the Gaussian distribution for class C_c is given by

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{\{n|\boldsymbol{x}^{(n)} \in C_c\}} \boldsymbol{x}^{(n)}$$

which represents the mean of those feature vectors assigned to class C_c.

Similarly, show that the maximum likelihood solution for the shared covariance matrix is given by

$$\boldsymbol{\Sigma} = \sum_{c=1}^{C} \frac{N_c}{N} \mathbf{S}_c$$

where

$$\mathbf{S}_c = \frac{1}{N_c} \sum_{\{n|\boldsymbol{x}^{(n)} \in C_c\}} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^T.$$

Thus $\boldsymbol{\Sigma}$ is given by a weighted average of the covariances of the data associated with each class, in which the weighting coefficients N_c/N are the prior probabilities of the classes.

If we substitute $p(\boldsymbol{x}|\boldsymbol{\theta}_c) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma})$ into $\log p(\mathcal{D}|\{\pi_c, \boldsymbol{\theta}_c\}_{c=1}^C)$ and then use the definition of the multivariate Gaussian, we obtain

$$\log p(\mathcal{D}|\{\pi_c, \boldsymbol{\theta}_c\}_{c=1}^C) = \frac{-1}{2} \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} (\log |\boldsymbol{\Sigma}| + (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c) + \log \pi_c)$$

Dropping terms independent of $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}$ we get

$$\log p(\mathcal{D}|\{\boldsymbol{\theta}_c\}_{c=1}^C) = \frac{-1}{2} \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} (\log |\boldsymbol{\Sigma}| + (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c))$$

Setting the derivative of the above equation w.r.t $\boldsymbol{\mu}_c$, (obtained using $\frac{\partial}{\partial x}(x^T a) = \frac{\partial}{\partial x}(a^T x) = a$), to zero we get

$$\sum_{n=1}^{N} y_c^{(n)} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c) = 0.$$

Making use of what we learned in the last problem, i.e.

$$\sum_{n=1}^{N} y_c^{(n)} = N_c,$$

we can re-arrange this to obtain

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{n=1}^{N} y_c^{(n)} \boldsymbol{x}^{(n)}.$$

Using the trace trick ($a = \mathrm{Tr}(a)$ for $a \in \mathbb{R}$ and $\mathrm{Tr}(\boldsymbol{ABC}) = \mathrm{Tr}(\boldsymbol{BCA})$) we can rewrite our original expression $\log p(\mathcal{D}|\{\boldsymbol{\theta}_c\}_{c=1}^C) = \frac{-1}{2} \sum_{n=1}^N \sum_{c=1}^C y_c^{(n)} (\log|\boldsymbol{\Sigma}| + (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c))$ as

$$\log p(\mathcal{D}|\{\boldsymbol{\theta}_c\}_{c=1}^C) = \frac{-1}{2} \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} (\log|\boldsymbol{\Sigma}| + \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^T).$$

We can now use $\frac{\partial}{\partial \boldsymbol{A}} Tr(\boldsymbol{AB}) = \boldsymbol{B}^T$ and $\frac{\partial}{\partial \boldsymbol{A}} \ln|\boldsymbol{A}| = (\boldsymbol{A}^{-1})^T$ and $\ln|\boldsymbol{A}^{-1}| = -\ln|\boldsymbol{A}|$ to calculate the derivative w.r.t. $\boldsymbol{\Sigma}^{-1}$. Setting this to zero we obtain

$$\frac{1}{2} \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} (\boldsymbol{\Sigma} - (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^T) = 0.$$

Making use of $\sum_{n=1}^N y_c^{(n)} = N_c$, we can re-arrange this to obtain

$$\boldsymbol{\Sigma} = \sum_{c=1}^{C} \frac{N_c}{N} \mathbf{S}_c$$

,

where

$$\mathbf{S}_c = \frac{1}{N_c} \sum_{n=1}^{N} y_c^{(n)} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^T.$$

Note that we do not enforce that $\boldsymbol{\Sigma}$ should be symmetric, but the solution shows that it is automatically symmetric.