Statistical Machine Learning 2018

Exercises and answers, week 9

16 November 2018

TUTORIAL

Exercise 1

We look at classification by the perceptron algorithm. The perceptron is an example of a linear discriminant model. It corresponds to a two-class model in which the input vector $\mathbf{x}$ is transformed into a feature vector $\boldsymbol{\phi}(\mathbf{x})$. This feature vector is then used to construct a linear model $y(\mathbf{x}) = f\left(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})\right)$, with bias component $\phi_0(\mathbf{x}) = 1$. The nonlinear activation function $f(\cdot)$ takes the form of a step function: $f(a) = +1$ for $a \geq 0$, $f(a) = -1$ for $a < 0$.

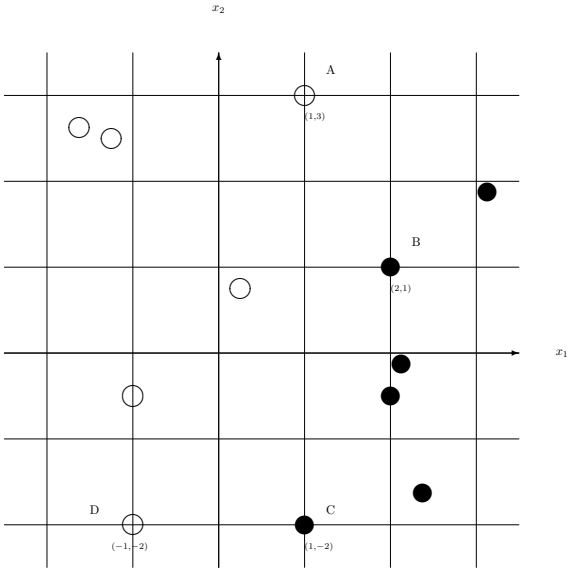

Figure 1: Data points from two classes

We use the $t \in \{-1, +1\}$ target coding scheme to let $t = +1$ denote class $\mathcal{C}_1$ and $t = -1$ denote class $\mathcal{C}_2$. Naturally, the problem now becomes how to determine the weight parameters $\mathbf{w}$ from a given dataset $\{\mathbf{x}_n, t_n\}$ in order to obtain the optimal classification scheme.

1. Explain how the perceptron classifies a (new) data point for a given set of $\mathbf{w}$. Why is it not a good idea to try to learn $\mathbf{w}$ simply from the error function defined by the total number of misclassified patterns in a data set (i.e. $E(\mathbf{w}) = \frac{1}{2}\sum_n |y(\mathbf{x}_n) - t_n|$) ?

   ANSWER: The perceptron assigns a value $y(\mathbf{x}) \in \{-1, +1\}$ to every input vector $\mathbf{x}$. Value $+1$ corresponds to class $\mathcal{C}_1$ and -1 corresponds to class $\mathcal{C}_2$. The value is determined by the

(inner) product of a feature vector $\boldsymbol{\phi}(\mathbf{x})$ with a vector of corresponding weights $\mathbf{w}$: if it is ≥ 0 then the step-function $f(\cdot)$ assigns value $+1$ to $y(\mathbf{x})$, otherwise value -1. The feature vector $\boldsymbol{\phi}(\mathbf{x})$ is a constant, possibly nonlinear, function of $\mathbf{x}$. Note that the dimension of $\boldsymbol{\phi}(\mathbf{x})$ and $\mathbf{x}$ does not have to be equal.
For a given data set, the optimal classifier should have the lowest number of misclassified patterns. As the feature vectors $\boldsymbol{\phi}(\mathbf{x})$ are fixed, the weights $\mathbf{w}$ in the perceptron completely determine the classification. However, the number of misclassified patterns only changes when the decision boundary crosses a data point: as a function of $\mathbf{w}$ it is constant almost everywhere. As a result, we have no information in which direction to shift the weight vector $\mathbf{w}$ in order to improve the classification rate.

A better approach is to define an error function known as the *perceptron criterion*

$$E_{\mathrm{P}}(\mathbf{w}) = -\sum_{n\in\mathcal{M}} \mathbf{w}^T\boldsymbol{\phi}_n t_n \tag{1}$$

with the sum taken over all misclassified patterns. We can use this function in a *stochastic gradient descent* technique: $\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \eta\nabla E_n$, with η a learning rate parameter.

2. Show this results in the following perceptron learning algorithm

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} + \eta\boldsymbol{\phi}(\mathbf{x}_n)t_n \tag{2}$$

ANSWER: For a given misclassified $\mathbf{x}_n$ in (1), the contribution to the error function is

$$E_n = -\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)t_n$$

Taking the derivative of E_n w.r.t. $\mathbf{w}$ and plugging into the equation for stochastic gradient descent then results in (2). This can now be used as a recipe for finding the optimal $\mathbf{w}$ (minimize the error function) by successively considering misclassified patterns and updating the weight vector according to (2) (provided the data set is linearly separable).
Note the crucial role of the $t \in \{-1, +1\}$ values in this equation.

3. Show that in the perceptron learning algorithm, we can set the learning parameter η equal to 1 without loss of generality.

ANSWER: As the step-function $f(a)$ only depends on the sign of a, multiplying $\mathbf{w}$ by an arbitrary (positive) constant (e.g. $1/\eta$) will not affect the classification. In particular if we set $\tilde{\mathbf{w}}^{\tau} = \eta^{-1}\mathbf{w}^{\tau}$ for all τ the perceptron learning algorithm for $\tilde{\mathbf{w}}^{\tau}$ is transformed into

$$\tilde{\mathbf{w}}^{\tau+1} = \eta^{-1}\mathbf{w}^{\tau+1} = \eta^{-1}(\mathbf{w}^{\tau} + \eta\boldsymbol{\phi}(\mathbf{x}_n)t_n) = \tilde{\mathbf{w}}^{\tau} + \boldsymbol{\phi}(\mathbf{x}_n)t_n$$

Therefore the learning rate parameter can be taken as $\eta = 1$.

In Figure 1 a dataset of two classes is depicted: the solid circles belong to class $\mathcal{C}_1$ and the open circles belong to class $\mathcal{C}_2$. We will only look at the subset of points {A,B,C,D}. The features correspond directly to the parameters x_1 and x_2, i.e. $\boldsymbol{\phi}(\mathbf{x}) \equiv [1, x_1, x_2]$.

3. Assume $\mathbf{w}^{(0)} = [0, 1, 0]$. Which of the data points {A,B,C,D} is not classified correctly for this initial weight vector?

ANSWER: The decision boundary is $\mathbf{w}^T\phi(\mathbf{x}) = 0$. For the given $\mathbf{w}^{(0)}$ this results in $(0 \cdot 1 + 1 \cdot x_1 + 0 \cdot x_2) = 0$, corresponding to the line $x_1 = 0$. The step function $f(\cdot)$ maps every point $\mathbf{x} = [x_1, x_2]^T$ with parameter (feature) $x_1 \geq 0$ to class $\mathcal{C}_1$. Therefore only point A will be classified incorrectly.

4. From the given $\mathbf{w}^{(0)}$, iterate over the set of points {A,B,C,D} (in that order) until the perceptron learning algorithm (2) reaches convergence (take $\eta = 1$). What is the final set of weight parameters?

   ANSWER:
   If a point is misclassified then update weight vector; go on to next until all points are classified correctly. As the data set looks linearly separable this procedure is guaranteed to converge in a finite number of steps ...

point classified	$\phi(\mathbf{x}_n)t_n$	$\mathbf{w}^{\tau+1}$	decision boundary
$A_{-1}(+1,+3) \Rightarrow +1$	$[-1,-1,-3]$	$\mathbf{w}^{(1)} = [-1,+0,-3]$	$x_2 = -1/3$
$B_{+1}(+2,+1) \Rightarrow -1$	$[+1,+2,+1]$	$\mathbf{w}^{(2)} = [+0,+2,-2]$	$x_2 = x_1$
$C_{+1}(+1,-2) \Rightarrow +1$	Ok	$-$	$-$
$D_{-1}(-1,-2) \Rightarrow -1$	$[-1,+1,+2]$	$\mathbf{w}^{(3)} = [-1,+3,+0]$	$x_1 = 1/3$
$A_{-1}(+1,+3) \Rightarrow +1$	$[-1,-1,-3]$	$\mathbf{w}^{(4)} = [-2,+2,-3]$	$x_2 = 2/3x_1 - 2/3$
$B_{+1}(+2,+1) \Rightarrow -1$	$[+1,+2,+1]$	$\mathbf{w}^{(5)} = [-1,+4,-2]$	$x_2 = 2x_1 - 1/2$

   After the fifth step all points are classified correctly. Note that, for example, after the first update the previously correctly assigned point B becomes misclassified.

Exercise 2

In *logistic regression* we start from the general form of the posterior probability of class $\mathcal{C}_1$, as a logistic sigmoid

$$\sigma(a) = \frac{1}{1+\exp(-a)} \tag{3}$$

acting on a linear function of the feature vector $\boldsymbol{\phi}$.

$$p(\mathcal{C}_1|\boldsymbol{\phi}) = y(\boldsymbol{\phi}) = \sigma(\mathbf{w}^T\boldsymbol{\phi}) \tag{4}$$

1. From the definition in (3), show that

$$\frac{d\ln\sigma}{da} = (1-\sigma) \tag{5}$$

   ANSWER:

$$\begin{aligned}
\frac{d\ln\sigma(a)}{da} &= \frac{1}{\sigma(a)}\frac{d\sigma(a)}{da} \\
&= \frac{1}{\sigma(a)}\frac{\exp(-a)}{(1+\exp(-a))^2} \\
&= \frac{\exp(-a)}{1+\exp(-a)} \\
&= \frac{(1-1)+\exp(-a)}{1+\exp(-a)} \\
&= \frac{(1+\exp(-a))-1}{1+\exp(-a)} \\
&= \frac{1+\exp(-a)}{1+\exp(-a)} - \frac{1}{1+\exp(-a)} \\
&= (1-\sigma)
\end{aligned}$$

2. For a data set $\{\boldsymbol{\phi}_n, t_n\}$, with $t_n \in \{0, 1\}$, the likelihood function can be written as

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n} \{1 - y_n\}^{1-t_n} \tag{6}$$

where $y_n = p(\mathcal{C}_1|\boldsymbol{\phi}_n)$. Define the *cross entropy* error as $E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w})$. Use the result (5) to show that the gradient of this error function w.r.t. $\mathbf{w}$ is given by

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\boldsymbol{\phi}_n \tag{7}$$

ANSWER:
From (6), the definition of the cross entropy error and $y_n = \sigma(\mathbf{w}^T\boldsymbol{\phi}_n)$ we have

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln \sigma(\mathbf{w}^T\boldsymbol{\phi}_n) + (1 - t_n) \ln(1 - \sigma(\mathbf{w}^T\boldsymbol{\phi}_n))\} \tag{8}$$

Taking the gradient w.r.t. $\mathbf{w}$ and using result (5) this gives

$$\begin{aligned}
\nabla E(\mathbf{w}) = \frac{\partial E}{\partial \mathbf{w}} &= \sum_{n=1}^{N} \{-t_n(1 - \sigma(\mathbf{w}^T\boldsymbol{\phi}_n))\boldsymbol{\phi}_n + (1 - t_n)\sigma(\mathbf{w}^T\boldsymbol{\phi}_n))\boldsymbol{\phi}_n\} \\
&= \sum_{n=1}^{N} \{-t_n\boldsymbol{\phi}_n + \sigma(\mathbf{w}^T\boldsymbol{\phi}_n)\boldsymbol{\phi}_n\} \\
&= \sum_{n=1}^{N} (\sigma(\mathbf{w}^T\boldsymbol{\phi}_n) - t_n)\boldsymbol{\phi}_n \\
&= \sum_{n=1}^{N} (y_n - t_n)\boldsymbol{\phi}_n
\end{aligned}$$

3. Describe how this can be used to obtain a gradient descent algorithm to learn the weight vector $\mathbf{w}$.

ANSWER:
As the expression for the gradient of the cross entropy error in (8) comprises of a sum of individual contributions from each data point, we can use gradient descent (Bishop, 3.22) to get

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \eta \sum_{n=1}^{N} (y_n - t_n)\phi_n \tag{9}$$

with η a suitably chosen learning rate parameter.

Note: a (much) more efficient method for this is the so called 'Iterative Reweighted Least Squares' method, or 'IRLS' for short, see Bishop, §4.3.3.

4. Show that if the classes are linearly separable, then the magnitude of $\mathbf{w}$ of the ML solution is unbounded (i.e. goes to infinity).

ANSWER:
If the classes are linearly separable, then there are weights $\mathbf{w}$ such that $\mathbf{w}^T\boldsymbol{\phi}_n < 0$ if $t_n = 0$

and $\mathbf{w}^T\boldsymbol{\phi}_n > 0$ if $t_n = 1$. If we take these weights, we find $y_n \equiv \sigma(\mathbf{w}^T\boldsymbol{\phi}_n) < 0.5$ if $t_n = 0$ and $y_n > 0.5$ if $t_n = 1$. Let us write the negative loglikelihood in terms of these y's,

$$E(\mathbf{w}) = -\sum_{t_n=1} \ln y_n - \sum_{t_n=0} \ln(1 - y_n) \tag{10}$$

If we now take $\mathbf{w}' = 2\mathbf{w}$, we see that the sigma's are all more saturated, i.e. all the y'_n's are closer to their corresponding t_n's. This means that the y'_n's in the first term $-\sum_{t_n=1} \ln y_n$ are bigger than the original y_n's, so the first term $-\sum_{t_n=1} \ln y_n$ gets smaller, and the y'_n's in the second term, $-\sum_{t_n=0} \ln(1 - y_n)$ are smaller than the original y_n's, so this second term in the negative loglikelihood gets also smaller. So the whole negative loglikelihood gets smaller. We can repeat the multiplication of the weights by two forever. In each step the negative loglikelihood decreases. So there is no bounded solution for $\mathbf{w}$.

Exercise 3

Laplace approximation (Bishop §4.4). Consider the probability density $p(t)$ defined as

$$p(t) = \frac{1}{Z} f(t) \tag{11}$$

with

$$f(t) = \begin{cases} t^2 \exp(-\lambda t) & , \quad t \geq 0 \\ 0 & , \quad t < 0 \end{cases} \tag{12}$$

in which λ is a positive constant. Z is an (unknown) normalizing constant.

1. Show that the mode t^* of $p(t)$ is located at $t = 2/\lambda$.

   ANSWER: Mode corresponds to the maximum of $f(t)$, so take derivative of (12) w.r.t. t and set equal to zero to give

$$\begin{aligned} f'(t) &= 2t\exp(-\lambda t) - \lambda t^2 \exp(-\lambda t) = Z^{-1}(2t - \lambda t^2)\exp(-\lambda t) = 0 \\ &\Rightarrow (2 - \lambda t) = 0 \\ &\Rightarrow t^* = 2/\lambda \end{aligned}$$

2. Create a second order Taylor expansion of $\ln f(t)$ around $t^* > 0$. Take the exponential to obtain an approximation to $f(t)$ in the form of an unnormalized Gaussian with mode t^*.

   ANSWER: We have: $\ln f(t) = 2\ln t - \lambda t$. The derivative of this is: $(\ln f(t))' = \frac{2}{t} - \lambda$, and the second derivative becomes: $(\ln f(t))'' = -\frac{2}{t^2}$.
   The Taylor expansion of $\ln f(t)$ around t^* up to second order is then given as

$$\begin{aligned} \ln f(t) &\simeq \ln f(t^*) + (t - t^*)(\ln f(t^*))' + \frac{1}{2}(t - t^*)^2(\ln f(t^*))'' \\ &= \ln f(t^*) - \frac{1}{2}\frac{\lambda^2}{2}(t - t^*)^2 \end{aligned}$$

   Where the contribution of the first order term is zero since we are in a maximum. Taking the exponential gives

$$f(t) \simeq f(t^*) \exp\left[-\frac{1}{2}\left(\frac{\lambda^2}{2}\right)(t - t^*)^2\right] \tag{13}$$

   which corresponds to a Gaussian potential as it has a quadratic term in the exponent.

3. Rewrite the approximation into a standard Gaussian $q(t)$, and use the normalization factor to compute an estimate for the unknown constant Z in (12). Compare to the true value of $Z = 2$ for $\lambda = 1$.

   ANSWER: From the coefficient in the exponential in (13) we know that the corresponding normalized Gaussian distribution now takes the form

$$q(t) = \left(\frac{A}{2\pi}\right)^{1/2} \exp\left\{-\frac{A}{2}(t - t^*)^2\right\} \tag{14}$$

   with $A = \lambda^2/2$.
   If $p(t)$ originally corresponded to a normalized distribution, then the approximation (13), divided by Z, should match the normalized Gaussian $q(t)$. In other words, the factors in front of the exponent should match, giving

$$\frac{1}{Z} f(t^*) = \left(\frac{A}{2\pi}\right)^{1/2}$$

   and so for the normalization constant

$$\begin{aligned} Z &= f(t^*)\left(\frac{2\pi}{A}\right)^{1/2} \\ &= \left(\frac{4}{\lambda^2}\exp(-2)\right)\left(\frac{4\pi}{\lambda^2}\right)^{1/2} \\ &= \frac{8\sqrt{\pi}\exp(-2)}{\lambda^3} \\ &\simeq \frac{1.919}{\lambda^3} \end{aligned}$$

   Filling in $\lambda = 1$ we see that the resulting estimated normalization coefficient $Z = 1.919$ matches the true value $Z = 2$ pretty close ... even though the distributions are qualitatively quite different (see plot in Matlab).

   Note: direct calculation of the normalization constant for the integral in (12) results in a factor of $\dfrac{2}{\lambda^3}$, indicating the approximation holds for other values of λ as well.

Exercise 4

Consider a data set $\mathcal{D}$ and a set of models $\{\mathcal{M}_i\}$ with parameters $\{\boldsymbol{\theta}_i\}$. In a Bayesian selection of the 'best' model, we would like to compare the posterior distribution of the models given the data: $p(\mathcal{M}_i|\mathcal{D})$. In doing so, the so called *model evidence*, quantifying the preference of the data for different models, plays an important role. From Bayes' theorem this is given by (eq.4.136)

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_i) p(\boldsymbol{\theta}|\mathcal{M}_i) \mathrm{d}\boldsymbol{\theta} \tag{15}$$

1. Show why (and under what assumptions) the model evidence is a good measure for comparing the model posteriors $p(\mathcal{M}_i|\mathcal{D})$. (see Bishop, §3.4)

   ANSWER: From Bayes' theorem, the relation between the model posterior $p(\mathcal{M}_i|\mathcal{D})$ and the model evidence $p(\mathcal{D}|\mathcal{M}_i)$ is

$$p(\mathcal{M}_i|\mathcal{D}) = \frac{1}{p(\mathcal{D})} p(\mathcal{D}|\mathcal{M}_i) p(\mathcal{M}_i)$$

Here the probability of the data $p(\mathcal{D})$ is just a normalizing constant (it does not change if we consider different models), so the posterior distribution is proportional to

$$p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M}_i)p(\mathcal{M}_i) \tag{16}$$

If we assume that the probability $p(\mathcal{M}_i)$ for each of the models is the same (we have no a priori reason to favour one over another) this implies that the model evidence is proportional to the model posterior: $p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M}_i)$.
Given this assumption, the model evidence is a good measure for comparing the posterior for different models, even though numerically the two may be very different.

Using Laplace approximation (§4.4) we found a general result for the approximation of the normalizing constant of a distribution

$$Z = \int f(\mathbf{z})\mathrm{d}\mathbf{z} \simeq f(\mathbf{z}_0) \int \exp\left\{-\frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^{\mathrm{T}}\mathbf{A}(\mathbf{z}-\mathbf{z}_0)\right\}\mathrm{d}\mathbf{z} = f(\mathbf{z}_0)\frac{(2\pi)^{M/2}}{|\mathbf{A}|^{1/2}} \tag{17}$$

2. Match this with (15) to derive the following approximation to the log model evidence

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\boldsymbol{\theta}_{\mathrm{MAP}}) + \ln p(\boldsymbol{\theta}_{\mathrm{MAP}}) + \frac{M}{2}\ln 2\pi - \frac{1}{2}\ln|\mathbf{A}| \tag{18}$$

Explain how the last three terms ('Occam factor', eq.4.137) penalize model complexity.

ANSWER: The Laplace approximation is based on fitting a Gaussian to the mode (maximum) $\mathbf{z}_0$ of an (unnormalized) distribution $f(\mathbf{z})$, with *inverse* covariance matrix $\mathbf{A}$ equal to the Hessian of $-\ln f(\mathbf{z})$ at $\mathbf{z}_0$, so the curvatures match up to second order at that point. Note that, as is often done in the Bishop book, the (obvious) explicit dependence on the model $\mathcal{M}_i$ in (18) is dropped in order to reduce 'notational clutter'.

Matching the elements we have for the mode $\mathbf{z}_0 \to \boldsymbol{\theta}_{\mathrm{MAP}}$ and so $f(\mathbf{z}_0) \to p(\mathcal{D}|\boldsymbol{\theta}_{\mathrm{MAP}})p(\boldsymbol{\theta}_{\mathrm{MAP}})$. Substituting in (17) gives

$$p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \tag{19}$$

$$\approx p(\mathcal{D}|\boldsymbol{\theta}_{\mathrm{MAP}})p(\boldsymbol{\theta}_{\mathrm{MAP}})\frac{(2\pi)^{M/2}}{|\mathbf{A}|^{1/2}} \tag{20}$$

where $\mathbf{A} = -\nabla\nabla \ln p(\mathcal{D}|\boldsymbol{\theta}_{\mathrm{MAP}})p(\boldsymbol{\theta}_{\mathrm{MAP}})$. Taking the log this reduces to the approximation (18) for the log model evidence.

To see how the last three components penalize models for complexity it is instructive to think of the Laplace approximation of the integral (17) as the product of the height of the Gaussian times the 'width' of the peak, which is roughly proportional to twice the variance. So, the last term in (20), corresponding to the last two terms in (18), is the *posterior* uncertainty in the parameters $\boldsymbol{\theta}$ for the given model, while the second term, $p(\boldsymbol{\theta}_{\mathrm{MAP}})$ corresponds to the inverse of the *prior* uncertainty in the parameters (roughly uniform over a wide range of possible parameters for the model). In other words, the 'Occam factor' corresponds to the ratio of the posterior accessible parameter space to the prior accessible volume: the more the data restricts the parameters for a given model, the smaller the Occam factor and with it the model evidence.
Specifically

- $\ln p(\boldsymbol{\theta}_{\mathrm{MAP}})$ - the more parameters in a model, the larger the accessible volume in parameter space and so the lower the average probability. Therefore for a complex model the log of this term will have a large negative score, heavily penalizing the model evidence.

- $\frac{M}{2} \ln 2\pi$ - this term actually becomes larger for models with more parameters (dimensions) , so in itself it does *not* penalize the model evidence. However, for more data, this term is completely dominated by the third term:
- $-\frac{1}{2} \ln |\mathbf{A}|$ - the more data, the lower the posterior variance and so the higher the negative Hessian (curvature) at $\boldsymbol{\theta}_{\mathrm{MAP}}$ and with it the value of the elements in $\mathbf{A}$. However, for the determinant $|\mathbf{A}|$ this change depends exponentially on the number of parameters/dimensions M in the model: it is multiplied for each parameter. Therefore, complex models with high numbers of parameters will be heavily penalized.