CS-E3210- Machine Learning Basic Principles
Home Assignment 3 - "Classification"

Your solutions to the following problems should be submitted as one single pdf which does not contain any personal information (student ID or name). The only rule for the layout of your submission is that each problem has to correspond to one single page, which has to include the problem statement on top and your solution below. You are welcome to use the LaTeX-file underlying this pdf, available under `https://version.aalto.fi/gitlab/junga1/MLBP2017Public`, and fill in your solutions there.

Problem 1: Logistic Regression - I

Consider a binary classification problem where the goal is classify or label a webcam snapshot into "winter" ($y = -1$) or "summer" ($y = 1$) based on the feature vector $\mathbf{x} = (x_{\mathrm{g}}, 1)^T \in \mathbb{R}^2$ with the image greenness x_{g}. A particular classification method is logistic regression, where we classify a datapoint as $\hat{y} = 1$ if $h^{(\mathbf{w})}(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) > 1/2$ and $\hat{y} = -1$ otherwise. Here, we used the sigmoid function $\sigma(z) = 1/(1 + \exp(-z))$.

The predictor value $h^{(\mathbf{w})}(\mathbf{x})$ is interpreted as the probability of $y = 1$ given the knowledge of the feature vector $\mathbf{x}$, i.e., $P(y = 1|\mathbf{x}; \mathbf{w}) = h^{(\mathbf{w})}(\mathbf{x})$. Note that the conditional probability $P(y = 1|\mathbf{x}; \mathbf{w})$ is parametrized by the weight vector $\mathbf{w}$. We have only $N = 2$ labeled data points with features $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$ and labels $y^{(1)} = 1, y^{(2)} = -1$ at our disposal in order to find a good choice for $\mathbf{w}$. Let $\mathbf{w}_{\mathrm{ML}}$ be a vector which satisfies

$$P(y{=}1|\mathbf{x}^{(1)}; \mathbf{w}_{\mathrm{ML}})P(y{=}{-}1|\mathbf{x}^{(2)}; \mathbf{w}_{\mathrm{ML}}) = \max_{\mathbf{w}\in\mathbb{R}^2} P(y{=}1|\mathbf{x}^{(1)}; \mathbf{w})P(y{=}{-}1|\mathbf{x}^{(2)}; \mathbf{w}).$$

Show that the vector $\mathbf{w}_{\mathrm{ML}}$ solves the empirical risk minimization problem using logistic loss $L((\mathbf{x}, y); \mathbf{w}) = \ln\big(1 + \exp\big(-y(\mathbf{w}^T\mathbf{x})\big)\big)$, i.e., $\mathbf{w}_{\mathrm{ML}}$ is a solution to

$$\min_{\mathbf{w}\in\mathbb{R}^2} (1/N) \sum_{i=1}^{N} L((\mathbf{x}^{(i)}, y^{(i)}); \mathbf{w}).$$

Answer. First, using $P(y{=}{-}1|\mathbf{x}^{(2)}; \mathbf{w}) = 1 - P(y{=}1|\mathbf{x}^{(2)}; \mathbf{w})$, we have

$$\begin{aligned}
\mathbf{w}_{\mathrm{ML}} &= \operatorname*{argmax}_{\mathbf{w}\in\mathbb{R}^2} P(y{=}1|\mathbf{x}^{(1)}; \mathbf{w})P(y{=}{-}1|\mathbf{x}^{(2)}; \mathbf{w}) \\
&= \operatorname*{argmax}_{\mathbf{w}\in\mathbb{R}^2} h^{(\mathbf{w})}(\mathbf{x}^{(1)})(1 - h^{(\mathbf{w})}(\mathbf{x}^{(2)})). \qquad (1)
\end{aligned}$$

Then, using the fact that $\log(x)$ is monotonically increasing (and therefore preserving ordering) for all $x > 0$, we obtain further

$$\begin{aligned}
\mathbf{w}_{\mathrm{ML}} &= \operatorname*{argmax}_{\mathbf{w}\in\mathbb{R}^2} h^{(\mathbf{w})}(\mathbf{x}^{(1)})(1 - h^{(\mathbf{w})}(\mathbf{x}^{(2)})) \\
&= \operatorname*{argmax}_{\mathbf{w}\in\mathbb{R}^2} \log(h^{(\mathbf{w})}(\mathbf{x}^{(1)})(1 - h^{(\mathbf{w})}(\mathbf{x}^{(2)}))) \\
&= \operatorname*{argmax}_{\mathbf{w}\in\mathbb{R}^2} \log h^{(\mathbf{w})}(\mathbf{x}^{(1)}) + \log(1 - h^{(\mathbf{w})}(\mathbf{x}^{(2)})). \qquad (2)
\end{aligned}$$

By definition of the loss function $L((\mathbf{x}, y); \mathbf{w})$, we have the identities

$$L((\mathbf{x}, 1); \mathbf{w}) = -\log h^{(\mathbf{w})}(\mathbf{x}), \text{ and } L((\mathbf{x}, -1); \mathbf{w}) = -\log(1 - h^{(\mathbf{w})}(\mathbf{x})). \qquad (3)$$

Combining (2) and (3) yields

$$\begin{aligned}
\mathbf{w}_{\mathrm{ML}} &= \operatorname*{argmax}_{\mathbf{w}\in\mathbb{R}^2} -L((\mathbf{x}^{(1)}, y^{(1)}); \mathbf{w}) - L((\mathbf{x}^{(2}, y^{(2)}); \mathbf{w}) \\
&= \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^2} L((\mathbf{x}^{(1)}, y^{(1)}); \mathbf{w}) + L((\mathbf{x}^{(2}, y^{(2)}); \mathbf{w}) \\
&= \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^2} (1/N) \sum_{i=1}^{N} L((\mathbf{x}^{(i)}, y^{(i)}); \mathbf{w}). \qquad (4)
\end{aligned}$$

Problem 2: Logistic Regression - II

Consider a binary classification problem where the goal is classify or label a webcam snapshot into "winter" ($y = -1$) or "summer" ($y = 1$) based on the feature vector $\mathbf{x} = (x_{\rm g}, 1)^T \in \mathbb{R}^2$ with the image greenness $x_{\rm g}$. A particular classification method is logistic regression, where we classify a datapoint as $\hat{y} = 1$ if $h^{(\mathbf{w})}(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) > 1/2$ and $\hat{y} = -1$ otherwise. Here, we used the sigmoid function $\sigma(z) = 1/(1 + \exp(-z))$.

Given some labeled snapshots $\mathbb{X} = \big\{(x^{i)}, y^{(i)})\big\}_{i=1}^N$, we choose the weight vector $\mathbf{w}$ by empirical risk minimization using logistic loss $L((\mathbf{x}, y); \mathbf{w}) = \ln\big(1+\exp\big(-y(\mathbf{w}^T\mathbf{x})\big)\big)$, i.e.,

$$\mathbf{w}_{\rm opt} = \arg\min_{\mathbf{w}\in\mathbb{R}^2} \underbrace{(1/N)\sum_{i=1}^N L((\mathbf{x}^{(i)}, y^{(i)}); \mathbf{w})}_{=f(\mathbf{w})}. \tag{5}$$

Since there is no simple closed-form expression for $\mathbf{w}_{\rm opt}$, we have to use some optimization method for (approximately) finding $\mathbf{w}_{\rm opt}$. One extremely useful such method is gradient descent which starts with some initial guess $\mathbf{w}^{(0)}$ and iterates

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha\nabla f(\mathbf{w}^{(k)}), \tag{6}$$

for $k = 0, 1, \ldots$. For a suitably chosen step-size $\alpha > 0$ one can show that $\lim_{k\to\infty} \mathbf{w}^{(k)} = \mathbf{w}_{\rm opt}$. Can you find a simple closed-form expression for the gradient $\nabla f(\mathbf{w}^{(k)})$ in terms of the current iterate $\mathbf{w}^{(k)}$ and the data points $\mathbb{X} = \big\{(x^{(i)}, y^{(i)})\big\}_{i=1}^N$.

Answer. The gradient is obtained as

$$\nabla f(\mathbf{w}^{(k)}) = (1/N)\sum_{i=1}^N \frac{-y^{(i)}\mathbf{x}^{(i)}}{1+\exp\big(y^{(i)}(\big(\mathbf{w}^{(k)}\big)^T\mathbf{x}^{(i)})\big)} \tag{7}$$

Problem 3: Bayes' Classifier - I

Consider a binary classification problem where the goal is classify or label a webcam snapshot into "winter" ($y = -1$) or "summer" ($y = 1$) based on the feature vector $\mathbf{x} = (x_{\mathrm{g}}, x_{\mathrm{r}})^T \in \mathbb{R}^2$ with the image greenness x_{g} and redness x_{r}. We might interpret the feature vector and label as (realizations) of random variables, whose statistics is specified by a joint distribution $p(\mathbf{x}, y)$. This joint distribution factors as $p(\mathbf{x}, y) = p(\mathbf{x}|y)p(y)$ with the conditional distribution $p(\mathbf{x}|y)$ of the feature vector given the true label y and the prior distribution $p(y)$ of the label values. The prior probability $p(y = 1)$ is the fraction of overall summer snapshots. Assume that we know the distributions $p(\mathbf{x}|y)$ and $p(y)$ and we want to construct a classifier $h(\mathbf{x})$, which classifies a snapshot with feature vector $\mathbf{x}$ as $\hat{y} = h(\mathbf{x}) \in \{-1, 1\}$. Which classifier map $h(\cdot) : \mathbf{x} \mapsto \hat{y} = h(\mathbf{x})$, mapping the feature vector $\mathbf{x}$ to a predicted label $\hat{y}$, yields the smallest error probability (which is $p(y \neq h(\mathbf{x}))$) ?

Answer. By definition of conditional probabilities and marginalization,

$$\begin{aligned} p(y \neq h(\mathbf{x})) &= \int_{\mathbf{z}} p(y \neq h(\mathbf{x})|\mathbf{x} = \mathbf{z})p(\mathbf{z})d\mathbf{z} \\ &= \int_{\mathbf{z}} p(y \neq h(\mathbf{z})|\mathbf{z})p(\mathbf{z})d\mathbf{z}, \end{aligned} \tag{8}$$

where $p(\mathbf{z})$ denotes the pdf value of the random vector $\mathbf{x}$ for $\mathbf{x} = \mathbf{z}$. We can minimize the integral in a point-wise fashion by, separately for each possible realization $\mathbf{z}$ of $\mathbf{x}$, choosing $h(\mathbf{z})$ to make $p(y \neq h(\mathbf{z})|\mathbf{z} = \mathbf{x})$ as small as possible or, equivalently, make $p(y = h(\mathbf{z})|\mathbf{z} = \mathbf{x})$ as large as possible. Thus, the optimal choice for $h(\mathbf{x})$ is given by

$$h(\mathbf{x}) = \underset{y' \in \{-1,1\}}{\operatorname{argmax}}\, p(y = y'|\mathbf{x}), \tag{9}$$

which is called the "maximum a-posteriori" (MAP) classifier since it is based on maximizing the posterior probability $p(\hat{y}|\mathbf{x})$ of the predicted label $\hat{y} = h(\mathbf{x})$ given (conditioned on) the feature vector $\mathbf{x}$.

Moreover, by Bayes' Theorem, we have

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}. \tag{10}$$

Combining (9) with (10), the optimal choice for $h(\mathbf{x})$ is given by

$$h(\mathbf{x}) = \underset{y' \in \{-1,1\}}{\operatorname{argmax}}\, p(\mathbf{x}|y')p(y'). \tag{11}$$

Problem 4: Bayes' Classifier - II

Reconsider the binary classification problem of Problem 3, where the goal is classify or label a webcam snapshot into "winter" ($y = -1$) or "summer" ($y = 1$) based on the feature vector $\mathbf{x} = (x_{\rm g}, x_{\rm r})^T \in \mathbb{R}^2$ with the image greenness $x_{\rm g}$ and redness $x_{\rm r}$. While in Problem 3 we assumed perfect knowledge of the joint distribution $p(\mathbf{x}, y)$ of features $\mathbf{x}$ and label y (which are modelled as random variables), now we consider only knowledge of the prior probability $P(y = 1)$, which we denote P_1. A useful "guess" for the distribution of the features $\mathbf{x}$, given the label y, is via a Gaussian distribution. Thus, we assume

$$p(\mathbf{x}|y = 1; \mathbf{m}_s, \mathbf{C}_s) = \frac{1}{\sqrt{\det\{2\pi\mathbf{C}_s\}}} \exp(-(1/2)(\mathbf{x}-\mathbf{m}_s)^T\mathbf{C}_s^{-1}(\mathbf{x}-\mathbf{m}_s))$$

and, similarly,

$$p(\mathbf{x}|y = -1; \mathbf{m}_w, \mathbf{C}_w) = \frac{1}{\sqrt{\det\{2\pi\mathbf{C}_w\}}} \exp(-(1/2)(\mathbf{x}-\mathbf{m}_w)^T\mathbf{C}_w^{-1}(\mathbf{x}-\mathbf{m}_w)).$$

How would you choose (fit) the parameters $\mathbf{m}_s, \mathbf{m}_w \in \mathbb{R}^2$ and $\mathbf{C}_s, \mathbf{C}_w \in \mathbb{R}^{2\times 2}$ for (to) a given labeled dataset $\mathbb{X} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$.

Answer. Let us denote $[N] = \{1, \ldots, N\}$, $N_s = \{i \in [N]|y^{(i)} = 1\}$, and $N_w = \{i \in [N]|y^{(i)} = -1\}$. We estimate the parameters $\mathbf{m}_s, \mathbf{C}_s, \mathbf{m}_w, \mathbf{C}_w$ using the maximum likelihood principle, i.e.,

$$\begin{aligned}(\widehat{\mathbf{m}}_s, \widehat{\mathbf{C}}_s, \widehat{\mathbf{m}}_w, \widehat{\mathbf{C}}_w) &= \underset{\mathbf{m}_s,\mathbf{C}_s,\mathbf{m}_w,\mathbf{C}_w}{\operatorname{argmax}} p(\mathbb{X}; \mathbf{m}_s, \mathbf{C}_s, \mathbf{m}_w, \mathbf{C}_w) = \underset{\mathbf{m}_s,\mathbf{C}_s,\mathbf{m}_w,\mathbf{C}_w}{\operatorname{argmax}} \log p(\mathbb{X}; \mathbf{m}_s, \mathbf{C}_s, \mathbf{m}_w, \mathbf{C}_w) \\ &= \underset{\mathbf{m}_s,\mathbf{C}_s,\mathbf{m}_w,\mathbf{C}_w}{\operatorname{argmax}} \Bigg\{ \underbrace{-\frac{N_s}{2}\log(\det(\mathbf{C}_s)) - \frac{1}{2}\sum_{i\in N_s}(\mathbf{x}^{(i)}-\mathbf{m}_s)^T\mathbf{C}_s^{-1}(\mathbf{x}^{(i)}-\mathbf{m}_s)}_{l(\mathbf{m}_s,\mathbf{C}_s)} \\ &\qquad \underbrace{-\frac{N_w}{2}\log(\det(\mathbf{C}_w)) - \frac{1}{2}\sum_{i\in N_w}(\mathbf{x}^{(i)}-\mathbf{m}_w)^T\mathbf{C}_w^{-1}(\mathbf{x}^{(i)}-\mathbf{m}_w)}_{l(\mathbf{m}_w,\mathbf{C}_w)} \Bigg\}. \quad (12)\end{aligned}$$

This maximum likelihood problem decomposes into two very similar sub-problems $(\widehat{\mathbf{m}}_s, \widehat{\mathbf{C}}_s) = \operatorname{argmax}_{\mathbf{m}_s,\mathbf{C}_s} l(\mathbf{m}_s, \mathbf{C}_s)$ and $(\widehat{\mathbf{m}}_w, \widehat{\mathbf{C}}_w) = \operatorname{argmax}_{\mathbf{m}_w,\mathbf{C}_w} l(\mathbf{m}_w, \mathbf{C}_w)$. By setting the partial derivative of $l(\mathbf{m}_s, \mathbf{C}_s)$ w.r.t. $\mathbf{m}_s$ to zero, we obtain $\widehat{\mathbf{m}}_s = \frac{1}{N_s}\sum_{i\in N_s}\mathbf{x}^{(i)}$. Rewriting the log-likelihood $l(\mathbf{m}_s, \mathbf{C}_s)$, using $\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\mathbf{BA})$, yields

$$\begin{aligned}l(\mathbf{m}_s, \mathbf{C}_s) &= -\frac{N_s}{2}\log(\det(\mathbf{C}_s)) - (1/2)\sum_{i\in N_s}(\mathbf{x}^{(i)}-\mathbf{m}_s)^T\mathbf{C}_s^{-1}(\mathbf{x}^{(i)}-\mathbf{m}_s)) \\ &= -\frac{N_s}{2}\log(\det(\mathbf{C}_s)) - (1/2)\sum_{i\in N_s}\mathrm{Tr}((\mathbf{x}^{(\mathrm{i})}-\mathbf{m}_{\mathrm{s}})^{\mathrm{T}}\mathbf{C}_{\mathrm{s}}^{-1}(\mathbf{x}^{(\mathrm{i})}-\mathbf{m}_{\mathrm{s}})) \\ &= \frac{N_s}{2}\log(\det(\mathbf{C}_s^{-1})) - (1/2)\sum_{i\in N_s}\mathrm{Tr}(\mathbf{C}_{\mathrm{s}}^{-1}(\mathbf{x}^{(\mathrm{i})}-\mathbf{m}_{\mathrm{s}})(\mathbf{x}^{(\mathrm{i})}-\mathbf{m}_{\mathrm{s}})^{\mathrm{T}}). \quad (13)\end{aligned}$$

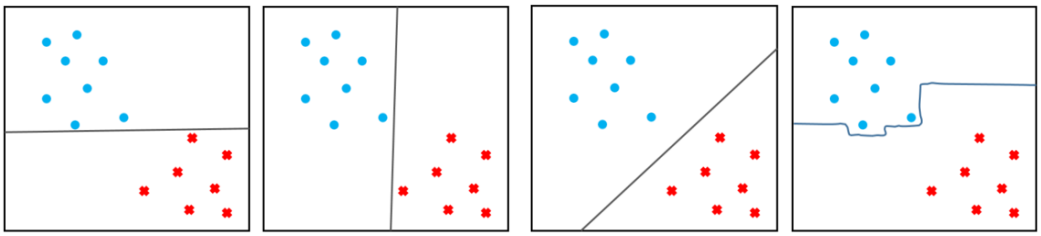
Computing the partial derivative $l(\mathbf{m}_s, \mathbf{C}_s)$ w.r.t $\mathbf{A} = \mathbf{C}_s^{-1}$ and equating to zero further yield

$$\widehat{\mathbf{C}}_s = \widehat{\mathbf{A}}^{-1} = (1/N_s)\sum_{i\in N_s}(\mathbf{x}^{(i)}-\widehat{\mathbf{m}}_s)(\mathbf{x}^{(i)}-\widehat{\mathbf{m}}_s)^T. \quad (14)$$

Similarly, we obtain $\widehat{\mathbf{m}}_w = \frac{1}{N_w}\sum_{i\in N_w}\mathbf{x}^{(i)}$ and $\widehat{\mathbf{C}}_w = (1/N_w)\sum_{i\in N_w}(\mathbf{x}^{(i)}-\widehat{\mathbf{m}}_w)(\mathbf{x}^{(i)}-\widehat{\mathbf{m}}_w)^T$.

Problem 5: Support Vector Classifier

Consider data points with features $\mathbf{x}^{(i)} \in \mathbb{R}^2$ and labels $y^{(i)} \in \{-1, 1\}$. In the figures below, the data points with $y^{(i)} = 1$ are depicted as red crosses and the data points with $y^{(i)} = -1$ are depicted as blue filled circles. Which of the four figures depicts a decision boundary which could have been generated by a SVC. Justify your selection.



(a) (b) (c) (d)

Answer. The classification method called SVC amounts to finding linear decision boundary, which is a hyperplane (excluding (d)), such that the distance of data points to the boundary (the "margin") is as large as possible. Among (a), (b) and (c), the boundary in (c) has the largest margin. Therefore, (c) could be a possible answer.