Start: 10.6.2024
Return: 24.6.2024
https://classroom.github.com

Assignment 5
Introduction to Probabilistic Machine Learning

Part 1: Bayesian Classification in Julia (3 Points)

- Relevant source files: `bayesian_linear_classification.jl`
- Supporting library files from previous assignments: `gaussian.jl, example.jl`
- Test files: `test_classification.jl`

In this task, we'd like to implement the Bayesian linear classification, as introduced in the lecture, as shown in Unit 8, slides 11 and 12.

For the first part of the task, please complete the function `learn_weight_distribution`. The function should take the following arguments:

- `phi`: The matrix of input values x, already transformed into the feature space, with the number of rows i being the number of samples.
- `y`: The vector containing the target values, i.e. the respective target label -1.0 or +1.0.
- `epsilon`: The value determening the stopping condition based on the values of a and b as described in on slide 11, Unit 8.
- `beta`: The noise parameter, as used in the likelihood definition in the lecture.
- `tau`: The parameter, used to initialize the covariance matrix, as described on slide 11, Unit 8.

The function should return the posterior distribution of the weights, using the same GaussianND struct that we used in the last exercise.

For the second part of the task, please complete the function `prediction`. The function should take the following arguments:

- `phi`: The matrix of input values x, already transformed into the feature space, with the number of rows i being the number of samples.
- `posterior`: The posterior distribution of the weights, using the same GaussianND struct as before.
- `beta`: The noise parameter, as used in the likelihood definition in the lecture.

The function should return a vector of floats, corresponding to the probabilities of the each sample having the label +1.0. You may want to implement this function for a single given feature vector ϕ_i first, using multiple dispatch, as indicated by the second function stub.

For your convenience during debugging, there are also functions and tests for intermediete results used in the algorithm on slide 11, Unit 8. Using these is optional.

Finally, if you want to test your code on a real life example, run `example.jl` to train and test a classifier on two classes from the MNIST data set and print out its accuracy on the test set.

Note: Accuracy is defined as the ratio of true classifications vs. all classifications, i.e.

$$Accuracy = \frac{\text{correct classifications}}{\text{all classifications}}$$

Accuracy is commonly used as a statistical measure for the performance of classification models, however it can be misleading when used on data sets that are not balanced, i.e., where the number of samples per class varies a lot. Fortunately, we were lazy enough to choose a data set where this is not the case.

Part 2: Fast Bayesian Linear Regression (1 Point)

We want to show the assertion of Unit 7, slide 16. We consider the following two densities of Normal Distributions, i.e., $N(w_i; \mu_i, \sigma_i^2)$ and $N\left(w_i; \phi_i^{-1} \cdot (y - \boldsymbol{\mu}^T \boldsymbol{\phi} + \mu_i \phi_i), \phi_i^{-2} \cdot (\beta^2 + \sum_j \phi_j^2 \sigma_j^2 - \phi_i^2 \sigma_i^2)\right)$ with $\sigma_i^2 > 0$.

Show the following identity:

$$
\begin{aligned}
&N(w_i; \mu_i, \sigma_i^2) \cdot N\left(w_i; \phi_i^{-1} \cdot \left(y - \boldsymbol{\mu}^T \boldsymbol{\phi} + \mu_i \phi_i\right), \phi_i^{-2} \cdot \left(\beta^2 + \textstyle\sum_j \phi_j^2 \sigma_j^2 - \phi_i^2 \sigma_i^2\right)\right) \\
&= N\left(w_i; \mu_i + \tfrac{y - \boldsymbol{\mu}^T \boldsymbol{\phi}}{\phi_i} \cdot \tfrac{\phi_i^2 \sigma_i^2}{\beta^2 + \sum_j \phi_j^2 \sigma_j^2}, \sigma_i^2 \cdot \left(1 - \tfrac{\sigma_i^2 \phi_i^2}{\beta^2 + \sum_j \phi_j^2 \sigma_j^2}\right)\right)
\end{aligned}
$$

Hint: Recall that $N(x; \mu_1, \sigma_1^2) \cdot N(x; \mu_2, \sigma_2^2) = N\left(x; \frac{\mu_1 \cdot \sigma_2^2 + \mu_2 \cdot \sigma_1^2}{\sigma_1^2 + \sigma_2^2}, \frac{\sigma_1^2 \cdot \sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)$ and use the constant $a := \sum_j \phi_j^2 \sigma_j^2$.

Part 3: Sherman-Morrison Formula (1 Point)

Consider the Sherman-Morrison Formula, see Unit 8, slide 9:

$$
\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T\right)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}
$$

Verify that the inverse is correct, i.e., that the following identity holds:

$$
\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T\right)\left(\boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}\right) = \boldsymbol{I}
$$

Hint: For a vector $\boldsymbol{x}$ and matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ (of suitable format) we have the relations (i) $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{x} = (\boldsymbol{A} + \boldsymbol{B})\boldsymbol{x}$ and (ii) $\boldsymbol{x}^T\boldsymbol{A} + \boldsymbol{x}^T\boldsymbol{B} = \boldsymbol{x}^T(\boldsymbol{A} + \boldsymbol{B})$.