Exercise Sheet 2 - Naive Bayes
Prof. Dr. Paul Swoboda

NOTE: For this exercise sheet, you must submit a jupyter notebook and a PDF.

Exercise 1: LogSumExp Trick (1.5 Points)

Computations with probabilities of widely varying orders of magnitude can lead to numerical instability issues with floating-point arithmetics. Such instability issues can be mildened by performing computations in log-space. Define

$$p_* = \prod_{i=1}^{n} p_i$$
$$p_+ = \sum_{i=1}^{n} p_i$$

where $p_1, \ldots, p_n \in [0, 1]$ are probabilities.

a) Suppose we want to compute p_* or p_+ given the p_i's. Why can such an approach lead to numerical instabilities with floating-point arithmetics? Try to construct concrete examples.

b) An alternative is to operate in log-space. Show how to compute $\ln p_*$ given the $\ln p_i$'s. (You are not allowed to compute $\exp(\ln p_i)$, of course.)

c) To compute $\ln p_+$ given the $\ln p_i$'s, we may use

$$\ln p_+ = \ln \sum_{i=1}^{n} \exp(\ln p_i). \tag{0.1}$$

Such an approach obviously does not help to address numerical stability issues.

i) The LogSumExp trick sets $c = \max_i \ln p_i$ and uses $\exp(\ln p_i - c)$ instead of $\exp(\ln p_i)$. Rewrite Eq. (0.1) using this trick (and make sure that it stays correct).

ii) Why does this approach reduce numerical stability issues?

Exercise 2: Conjugate Priors for Gaussians (1.5 Points)

Let us assume that our data distribution is Gaussian, i.e.

$$p(x|\theta) \sim \mathcal{N}(x|\mu, \sigma^2) \tag{0.2}$$

where our parameters θ are $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_+$. Also assume that our prior distribution on the data distribution mean is given by

$$p(\mu|\mu_0, \sigma_0^2) \sim \mathcal{N}(\mu|\mu_0, \sigma_0^2) \tag{0.3}$$

Let us assume that our observed data is $\mathcal{D} = (x^1, \dots, x^n)$. Show that the posterior is given by

$$p(\mu|\mathcal{D}, \mu_0, \sigma_0^2) \sim \mathcal{N}\left(\frac{1}{n}\left(\frac{\sigma_0^2}{\frac{\sigma^2}{n} + \sigma_0^2}\bar{x} + \frac{\sigma^2}{\frac{\sigma^2}{n} + \sigma_0^2}\mu_0\right), \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}\right)^{-1}\right) \tag{0.4}$$

where ~~$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x^i$~~ $\bar{x} = \sum_{i=1}^{n} x^i$.

Exercise 3: Naive Bayes (2 Points)

We will use a preprocessed variant of the MNIST digits dataset in this assignment. The task is to classify hand-written digits. There is one class for each digit (i.e., classes $0, 1, 2, ..., 9$). The features represent a scanned image (28×28 pixels, values in $\{0, 1, \dots, 255\}$). The dataset contains both training data ($\approx$ 6000 images per class) and test data ($\approx$ 1000 images per class). See `http://yann.lecun.com/exdb/mnist/` for more information.

Utilize the provided Jupyter notebook to complete the implementation of a Naive Bayes classifier. We will implement Bayesian Naive Bayes with symmetric Dirichlet priors to enhance performance. In Naive Bayes, the prior and posterior probabilities are determined by counting the relevant samples from the training set. To perform Naive Bayes classification, these counts are adjusted with a pseudo-counts α. When the class probability distribution estimated by MLE (i.e. without a prior) is given by

$$P(C) = \frac{N_C}{N}$$

with N_C being the number of samples in class C and N being the total number of samples. For a Dirichlet prior this terms transforms into

$$P(C) = \frac{N_C + \alpha - 1}{N + N_{classes} * (\alpha - 1)}$$

$N_{classes}$ is the number of classes. The class-conditional posteriors have to be determined analogously.