ELEC 548 Byron's Review of Classification

A What is Classification

Classification is a classic instance of **supervised learning**. The goal is to define a **classifier** which properly assigns a new data point – let's call it **x** – to the appropriate discrete class $C_k \in \{C_1, \dots, C_K\}$. While there are solutions to classification problems (i.e., nonparametric) which can be posed when the number of classes, K, is not known *a priori* or is otherwise unbounded, below we will assume that it not just countable, but also finite and typically specified as part of the problem statement. Notice that the discrete number of classes is what makes this a *classification* problem — the machine learning equivalent for continuous data would generally be termed *regression*.

The process of solving a a classification problem involves two stages: **Training** – optimizing the classifier using labeled training data, and **Testing** – evaluating the performance of the optimized classifier on labeled testing data. Thus, we always divide our labeled data into two: a test data set to evaluate performance and a separate training data set to ensure that the classifier we have learned generalizes. (Aside - if we didn't do this, what would be the best performing classifier? A lookup table!) In cases when there are hyperparameters to be optimized – for example in the model or statistical distribution of the data – we can split the labeled data into three groups: a training set, a **validation** data set to pick the best model, and a test set to evaluate final performance.

Classification is a machine learning problem with a wide variety of formulations and solutions. Below, we review classification using **Probabilistic Generative Models**. For more information as well as a description of other classification approaches, a good resource is *Pattern Recognition and Machine Learning* by Christopher Bishop.

B Classification Using Probabilistic Generative Models

In a classifier built using a probabilisitic generative model, there are two densities for each class $k \in \{1, \dots, K\}$:

— the class-conditional density, $\Pr(\mathbf{x} \mid C_k)$ and

— the class priors $\Pr(C_k)$

To **train** the classifier, we can use *maximum likelihood parameter estimation*. To use the classifier (i.e., on **test** data), we chose the class which maximizes the *a posteriori* probability. In other words, we

— use Bayes' rule to compute $\Pr(C_k \mid \mathbf{x})$

$$\begin{aligned}\Pr(C_k \mid \mathbf{x}) &= \frac{\Pr(\mathbf{x} \mid C_k)\Pr(C_k)}{\Pr(\mathbf{x})} \\ &= \frac{\Pr(\mathbf{x} \mid C_k)\Pr(C_k)}{\sum_{i=1}^{K}\Pr(\mathbf{x} \mid C_i)\Pr(C_i)}\end{aligned}$$

— assign **x** to class C_m where

$$m = \operatorname*{argmax}_k \Pr(C_k \mid \mathbf{x})$$

B.1 Maximum Likelihood Parameter Estimation

Once we write down the *training data likelihood*, maximum likelihood parameter estimation is not complicated. What is the data likelihood? Let us assume we are given training data: $\{\mathbf{x}_n, t_n\}, \quad n = 1, \dots, N$, where for each of the N training data, t_n is the label for data point x_n. Then, the *data likelihood* is just: