# Lab 2 - Classical Models

{bengio,mkeller}@idiap.ch
http://www.idiap.ch/~{bengio,mkeller}

November 25, 2005

1. **Getting familiarized with python: function to process classification databases**

   Download:

   - `data.py` a set of functions to process classification databases,
     [
     Reminder: Open them with a smart enough editor (*eg* `C:\Program␣Files\Notepad2\Notepad2.exe`),
     Explore them using `ipython`. Try for example:

     ```
     > cd toyour\download\path
     > run -i data.py
     > ?train_test()
     ```

     ]
   - `http://www.idiap.ch/ bengio/lectures/dbases/index.html` a choice of databases.
   - Choose a database, try on it `data.py` functions and compute the examples' mean and variance.

2. **Bayes Classifier**

   Show that for a bayes classifier:

$$\frac{P(X|Y=0)}{P(X|Y=1)} > \frac{P(Y=1)}{P(Y=0)} \Leftrightarrow \hat{y} = 0$$

.................................................................

$$\hat{y} = \text{argmax}_{i\in\{0,1\}} P(Y=i|X) = 0,$$

$$\Leftrightarrow P(Y=0|X) > P(Y=1|X)$$

$$\Leftrightarrow \frac{P(X|Y=0)P(Y=0)}{P(X)} > \frac{P(X|Y=1)P(Y=1)}{P(X)}$$

$$\Leftrightarrow \frac{P(X|Y=0)}{P(X|Y=1)} > \frac{P(Y=1)}{P(Y=0)}$$

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

In practice instead of estimating $P(Y=i)$, the ratio $\frac{P(Y=1)}{P(Y=0)}$ is replaced by a threshold $\theta$, which is tuned to minimize the error.

- Generate a training set of $N_0$ points from a gaussian distribution $\mathcal{N}(-0.5, 1)$ (using `random.gauss(mean,std)`, don't forget `import random`) and $N_1$ points from a $\mathcal{N}(0.5, 1)$.
- Estimate with gaussian models $\hat{p}_i(x)$ the densities $P(X|Y=i)$ by maximizing the likelihoods and compare the estimated values to the actual ones.
- On a validation set tune $\theta$ and compare its value to $\frac{N_1}{N_0}$.
  (Hint: For a validation set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, sort the ratios $\frac{\hat{p}_0(x_1)}{\hat{p}_1(x_1)}, \ldots, \frac{\hat{p}_0(x_n)}{\hat{p}_1(x_n)}$ in increasing order, and compute the error for $\theta$ equal to each ratio. Choose the $\theta$ corresponding to the smallest error.)

3. **Implement a $K$ Nearest Neighbors classification function.**

   Hint: Use `bbox.py` as a model.

   Edit `decision.py` to observe the modification of the decision function over a 2 dimensions database with respect to the hyperparameter $K$ variation. And compare with the decision function of `bbox.py` (Parzen Window).

4. **Curse of dimensionality**

   Let place ourselves in a 1 Nearest Neighbor framework. We have an $m$ dimensional training set $D_{train}$ from which the labels of a test set $D_{test}$ are estimated. We want to show empirically that for non-structured data:

$$\frac{d_{max}}{d_{min}} \longrightarrow 1 \text{ when } m \to \infty$$

   (Which makes NN meaningless in high dimension.)

   - Generate using `random.uniform(0,1)`, $D_{train}$ and $D_{test}$ for several values of $m$.
   - Compute the maximum and minimum euclidian distances of each of the test examples to the training examples.
   - Plot the average of $\frac{d_{max}}{d_{min}}$ against $m$.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

See `dim_curse.py`.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .