Probabilistic tools

Central Limit Theorem
Let $(X_n)_{n\geq 1}$ be a real and independent sequence with same law such that $\mu = \mathbb{E}[X_1]$ and $\mathbb{V}[X_1] = \sigma^2$ are defined ($\mathbb{V}[X_1] \leq +\infty$). Noting $\bar{X}_n = \frac{1}{n}(X_1 + ... + X_n)$, we have:

$$\sqrt{n}\frac{(\bar{X}_n - \mu)}{\sigma} \sim_{n\to\infty} \mathcal{N}(0,1)$$

Spectral Theorem
Let M be a symmetric matrix with real coefficients. Then it exists U orthogonal and D diagonal with real coefficients such that $M = UDU^T$.

Inferential statistics

Likelihood method

This method consists on finding the parameter that maximizes the likelihood:
$L(x_1, ..., x_n; \theta) = f(X|\theta) = \prod_{i=1}^{n} f_\theta(x_i; \theta)$ which is the product of densities across all samples.

Intuitively, we want to find the θ that maximizes a certain event, that is, obtaining some data X (which is why we have $X|\theta$).
We often use the log in order to get rid of power coefficients appearing with the product.
likelihood equation: $\frac{d}{d\theta} ln(L(x_1, ..., x_n; \theta)) = 0$

Exploratory statistics

Mahalanobis distance
Mahalanobis distance is a good alternative to Euclidean distance. For any given point x in a set X, the squared Mahalanobis distance is:

$$D^2 = (x - \mu_X)^T \Sigma^{-1} (x - \mu_X)$$

Advantage : it takes into account the data standard deviation and correlation. The more the data is dispersed, the lower the distance is. Indeed, using the inverse matrix is like if we divided the distance from the mean $(x - \mu_X)$ by the standard deviation.
Note: Euclidean distance is when $\Sigma = Id$.

Predictive models

ROC curve

ROC curve is used essentially for binary classification.
ROC = Receiver Operating Curve

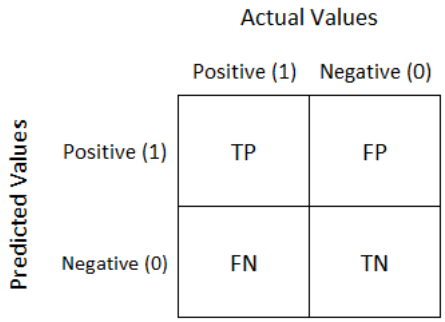
Use of the ROC curve

One model:
We use ROC curve to evaluate the performance of one classifying model that we can obtain when varying a threshold.

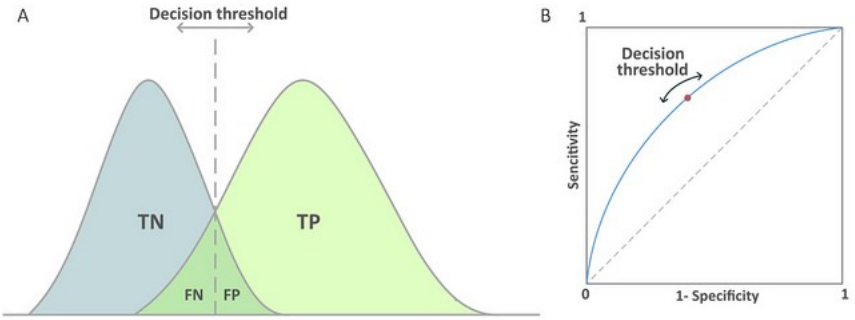
Several models:
We use ROC curve to compare several classifying models in evaluating the area under the curve (AUC) for a range of threshold.

Intuition

After running the prediction of a specific model, we draw the confusion matrix (actual vs predited) with a certain threshold.



We then modify the threshold and draw another confusion matrix.
The ROC graph summarizes all of the confusion matrices that each threshold produced.



On the left picture we see the ability of a model to give a clear distinction between the two classes. The curves are drawn from the predictions and the actual results (**how?**)

Implementation

1. Get probability predictions
2. Sort the probabilities (prediction)
3. Sort the validation (actual) according to previous sort
4. Loop on the sorted validation. At each iteration:
- increment TP or FP
- compute the TPR and FPR.
5. Plot (FPR, TPR)

See *https://docs.eyesopen.com/toolkits/cookbook/python/plotting/roc.html* for an implementation example, or data challenge Face_Recognition.