MULTIVARIATE STATISTICS, Lesson 12.
Classification and Clustering

- **Discriminant Analysis** for given groups (Statistical Learning, Artificial Intelligence)

  We want to separate given (by the expert) groups of objects based on multivariate observations, e.g., diagnostic groups of patients based on clinical measurements. If the separation is good, we can use the algorithm for diagnostic purposes (carefully) without the expert, just making predictions by calculations based on the clinical measurements.

  The algorithm minimizes the average Bayesian loss function (of missclassifications). Here we discuss the multivariate Gaussian case. The observed random vector is $\mathbf{X} \sim \mathcal{N}_p(\mathbf{m}_j, \mathbf{C}_j)$ in the j-th group, $j = 1, \ldots, k$. Based on its observed value $\mathbf{x}$ we classify the object into the group j, for which index j the *quadratic discriminant score*

  $$S'_j(\mathbf{x}) = -\frac{1}{2} \ln |\mathbf{C}_j| - \frac{1}{2}(\mathbf{x} - \mathbf{m}_j)^T \mathbf{C}_j^{-1} (\mathbf{x} - \mathbf{m}_j) + \ln \pi_j$$

  is maximum, where π_j is the relative importance (e.g., relative frequency) of the group j. If $\mathbf{C}_1 = \cdots = \mathbf{C}_k = \mathbf{C}$ then it suffices to maximize the *linear informant*

  $$S''_j(\mathbf{x}) = \mathbf{m}_j^T \mathbf{C}^{-1} \mathbf{x} - \frac{1}{2} \mathbf{m}_j^T \mathbf{C}^{-1} \mathbf{m}_j + \ln \pi_j.$$

  In case of $k = 2$, the classification depends on the sign of

  $$S''_1(\mathbf{x}) - S''_2(\mathbf{x}) = L(\mathbf{x}) - c,$$

  where $L(\mathbf{x})$ is a linear function of the measurements (the coefficients show the importance of the measurements in the classification). Then simple or jackknife cross-classification indicates the goodness of classification.

- **Cluster Analysis** (for finding groups of data)

  In case of metric data: Total variance=Within groups variance+Between groups variance. Therefore, minimizing $W/B = W/(T - W)$ is equivalent to minimizing

  $$W = \sum_{j=1}^k W_j = \sum_{j=1}^k \sum_{\mathbf{x}_i \in \mathcal{C}_j} \|\mathbf{x}_i - \mathbf{c}_j\|^2 = \sum_{j=1}^k \frac{1}{|\mathcal{C}_j|} \sum_{i<i',\, \mathbf{x}_i, \mathbf{x}_{i'} \in \mathcal{C}_j} \|\mathbf{x}_i - \mathbf{x}_{i'}\|^2$$

  over the k-partitions $\mathcal{C}_1, \ldots, \mathcal{C}_k$ of the n points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, where $\mathbf{c}_j = \frac{1}{|\mathcal{C}_j|} \sum_{\mathbf{x}_i \in \mathcal{C}_j} \mathbf{x}_i$ is the baricenter of cluster j. **Algorithms**:

  - *k-means* (MacQueen) method: For given k, starting with initial clustering $\mathcal{C}_1^{(0)}, \ldots, \mathcal{C}_k^{(0)}$ of objects, the following two steps are alternated until convergence. $m := 1$.
    1. Find the clusters' baricenters: $\mathbf{c}_1^{(m-1)}, \ldots, \mathbf{c}_k^{(m-1)}$.
    2. Replace $\mathbf{x}_i$ into the cluster j, if $j = \operatorname{argmin}_\ell \|\mathbf{x}_i - \mathbf{c}_\ell^{(m-1)}\|$, $\quad i = 1, \ldots, n$. Then $m := m + 1$, and go to step 1 with the so obtained new clusters $\mathcal{C}_1^{(m)}, \ldots, \mathcal{C}_k^{(m)}$.

    As the objective function in each step decreases, the algorithm converges to a local minimum of it. In case of *well-separated* clusters, the global minimum is found.
  - *Hierarchical* methods (agglomerative/divisive): We use the $n \times n$ symmetric similarity/dissimilarity matrix of objects. In the agglomerative method we start with n clusters, and in each step the clusters with similarity above (dissimilarity below) a given threshold are merged, until 1 cluster is present. In each step we find new cluster centers and similarities/dissimilarities between them. Based on the so obtained *dendrogram*, the user decides, how many clusters he/she needs.