

Extended CIL Summary

FS 2013

Pascal Spörri
pascal@spoerri.io

July 16, 2013

This summary is based on the course slides of the Computational Intelligence Lab slides¹ from spring semester 2013.

¹<http://cil.inf.ethz.ch>

Contents

I. Dimensionality Reduction	4
1. Intrinsic Dimensionality	4
2. Principal Component Analysis	4
2.1. Statistics of Projected Data	5
2.2. Maximisation Problem	5
2.2.1. Second principal direction	6
2.3. Solution: Eigenvalue Decomposition	6
2.4. Error Formulation	6
2.5. Matrix viewpoint	7
2.6. Computation	8
3. Singular Value Decomposition	8
3.1. Introduction	8
3.2. Singular values	9
3.3. Closest Rank- k Matrix	10
3.4. Properties	10
3.5. Movie Example	10
II. Clustering	12
4. Introduction	12
4.1. Problem	12
4.1.1. The Cost Function of Vector Quantisation	12
5. K -Means	13
5.1. Overview	13
5.2. Algorithm	13
5.3. Stability	14
5.3.1. High-Level Stability test	14
5.3.2. Distance between Clusterings	15
5.3.3. Calculation of Stability	15
6. Clustering as Matrix Factorisation	16
7. Mixture Models	16
7.1. Introduction	16
7.2. Gaussian Mixture Model	17
7.2.1. Generative Viewpoint	17
7.2.2. Parameter Estimation	17

7.2.3. Probability of assigning a data point to a cluster	18
7.2.4. Expectation-Maximisation for Gaussian Mixture Models	18
7.2.5. Algorithm	19
7.2.6. Relation to K -means	19
7.2.7. Log Likelihood	19
7.2.8. Matrix Factorisation	20
8. Model Order Selection and Information Criteria	20
8.0.9. Negative Log-Likelihood	20
8.1. AIC and BIC	21
8.1.1. AIC - Akaike Information Criterion	21
8.1.2. BIC - Bayesian Information Criterion	21
Appendix	22
A. Matrix Definitions and Theorems	22
A.1. Norms	22
A.1.1. Vector norms	22
A.1.2. Matrix norms	22
A.2. Orthogonality	23
B. Occam's Razor	23
C. Probability	23
C.1. Distributions	24
C.1.1. Categorical distribution	24
C.1.2. Gaussian Distribution	25
C.1.3. Multivariate Gaussian Distribution	25
C.1.4. Multidimensional Moment Statistics	25
C.2. Latent Variables	26
C.3. Bayes' rule	27
D. Lagrange Multipliers	27

Part I.

Dimensionality Reduction

Select the *most interesting* dimensions.

1. Intrinsic Dimensionality

Pairwise Distances

Assume components of data $x = (x_1, \dots, x_D)^T \in \mathbb{R}^D$ are i.i.d. Gaussian distributed:

$$x_d \sim \mathcal{N}(0, 1) \implies x_d - y_d \sim \mathcal{N}(0, 2).$$

Using χ^2 -distribution:

$$\frac{1}{2}(x_d - y_d)^2 \sim \chi^2(1),$$

and extending to D dimensions:

$$\frac{1}{2} \sum_{d=1}^D (x_d - y_d)^2 \sim \chi^2(D) = \Gamma\left(\frac{D}{2}, 2\right)$$

$$\text{Recall: } \forall z, k, \theta > 0, \Gamma(z; k, \theta) = \frac{\theta^k}{\Gamma(k)} y^{k-1} e^{-\theta y}$$

Hence, the dimension-normalised squared distance is:

$$\frac{1}{D} \sum_{d=1}^D (x_d - y_d)^2 \sim \Gamma\left(\frac{D}{2}, \frac{4}{D}\right)$$

is Gamma distributed with mean 2 and variance $\frac{8}{D}$.

$\Gamma\left(\frac{D}{2}, \frac{4}{D}\right)$ tends towards normality with shrinking width for large D . Therefore, most points have *constant* pairwise distances in this limit.

2. Principal Component Analysis

Objectives of PCA:

1. Minimise error $\|x_n - \tilde{x}_n\|$ of point x_n and its approximation \tilde{x}_n .
2. Reveal "interesting" information: maximise *variance*.

Both objectives are show to be formally equivalent.

Consider a set of observations $\{x_n\}$, $n = 1, \dots, N$ and $x_n \in \mathbb{R}^D$.

Goal Project data onto $K < D$ dimensional space while maximising variance of the projected data.

For $K = 1$ Define direction of projection as u_1 . Set $\|u_1\|_2 = 1$ (only the direction of the projection is important).

2.1. Statistics of Projected Data

Original Data

Mean is given by the sample mean \bar{x} .

Covariance of the Data:

$$\Sigma = \frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})(x_n - \bar{x})^T$$

Projected Data

Mean is given by: $u_1^T \bar{x}$.

Variance is given by:

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N \{u_1^T x_n - u_1^T \bar{x}\}^2 &= \frac{1}{N} \sum_{n=1}^N \{u_1^T (x_n - \bar{x})\}^2 \\ &= \frac{1}{N} \sum_{n=1}^N u_1^T (x_n - \bar{x})(x_n - \bar{x})^T u_1 \\ &= u_1^T \Sigma u_1. \end{aligned}$$

2.2. Maximisation Problem

These statistics now can be fed into a maximisation problem:

$$\max_{u_1} u_1^T \Sigma u_1$$

such that $\|u_1\|_2 = 1$.

Writing the Lagrangian results in in:

$$\mathcal{L} := u_1^T \Sigma u_1 + \lambda_1 (1 - u_1^T u_1).$$

Setting $\frac{\delta}{\delta u_1} \mathcal{L} \stackrel{!}{=} 0$ results in:

$$\Sigma u_1 = \lambda_1 u_1$$

We observe that u_1 is an *eigenvector* of Σ and λ_1 it's associated *eigenvalue*. Furthermore λ_1 is also the variance of the projected data:

$$\lambda_1 = u_1^T \Sigma u_1$$

2.2.1. Second principal direction

The second principal direction can be obtained by maximising the variance $u_2^T \Sigma u_2$, subject to $\|u_2\|_2 = 1$ and $u_2^T u_1 = 0$:

$$\mathcal{L} = u_2^T \Sigma u_2 + \lambda_2 (1 - u_2^T u_2) + \nu (u_2^T u_1).$$

The maximum is found by setting $\frac{\delta \mathcal{L}}{\delta u_2} \stackrel{!}{=} 0$:

$$2\Sigma u_2 - 2\lambda_2 u_2 + \nu u_1 = 0.$$

Because of the orthogonality between u_2 and u_1 we observe that u_2 contains no component of u_1 and hence $\nu = 0$. We get:

$$\Sigma u_2 = \lambda_2 u_2.$$

We observe that u_2 is an eigenvector of Σ with the second largest eigenvalue of λ_2 .

2.3. Solution: Eigenvalue Decomposition

Hence we see that the eigenvalue decomposition of the covariance matrix

$$\Sigma = U \Lambda U^T$$

contains all relevant information.

For a projection space of size $K \leq D$ we choose the K eigenvectors $\{u_1, \dots, u_K\}$ with the largest associated eigenvalues $\{\lambda_1, \dots, \lambda_K\}$.

2.4. Error Formulation

We define an *orthonormal* basis $\{u_d\}$, $d = 1, \dots, D$ of \mathbb{R}^D . The scalar projection of x_n onto u_d (magnitude) is given by:

$$z_{n,d} = x_n^T u_d.$$

The associated projection onto u_d amounts to $z_{n,d} u_d$. Therefore, each data point can be represented in the basis by:

$$x_n = \sum_{d=1}^D z_{n,d} u_d = \sum_{d=1}^D (x_n^T u_d) u_d.$$

Restricted representation using $K < D$ basis vectors can be written as:

$$\tilde{x}_n = \sum_{d=1}^K a_{n,d} u_d + \sum_{d=K+1}^D b_d u_d,$$

where b_d does not depend on the data point x_n .

The approximation error can be represented by:

$$J(\{a_{n,d}\}, \{b_d\}) = \frac{1}{N} \sum_{n=1}^N \|x_n - \tilde{x}_n\|_2^2$$

Minimisation of J w.r.t. $a_{n,d} = x_n^T$

Minimisation of J w.r.t. $b_d = \bar{x}^T u_d$

The displacement can be obtained by resubstituting $a_{n,d}$ and b_d :

$$x_n - \tilde{x}_n = \sum_{d=K+1}^D \left\{ (x_n - \bar{x})^T u_d \right\} u_d.$$

We observe that the displacement vector is orthogonal to the principal space!
Resubstituting the displacement into the error criterion leads to:

$$J = \frac{1}{N} \sum_{n=1}^N \sum_{d=K+1}^D (x_n^T u_d - \bar{x}^T u_d)^2 = \sum_{d=K+1}^D u_d^T \Sigma u_d$$

2.5. Matrix viewpoint

The data can be represented as matrix:

$$X = [x_1, \dots, x_n, \dots, x_N]$$

The corresponding zero-centered data is:

$$\bar{X} = X - M,$$

where $M = \underbrace{[\bar{x}, \dots, \bar{x}]}_{N \text{ times}}$.

Compute the projection of \bar{X} on $U_k = [u_1, \dots, u_K]$ with:

$$\underbrace{\bar{Z}_K}_{K \times N} = \underbrace{U_K^T}_{K \times D} \cdot \underbrace{\bar{X}}_{D \times N}.$$

To approximate \bar{X} , we return to the original basis:

$$\tilde{\bar{X}} = U_K \cdot \bar{Z}_K.$$

For $K = D$ we obtain a perfect reconstruction.

2.6. Computation

First compute the *empirical mean*:

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n$$

Then *center the data* by subtracting the mean from each sample:

$$\bar{X} = X - M,$$

where $M = \underbrace{[\bar{x}, \dots, \bar{x}]}_{N \text{ times}}$. Now compute the *Covariance matrix*:

$$\Sigma = \frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})(x_n - \bar{x})^T = \frac{1}{N} \underbrace{\bar{X} \bar{X}^T}_{\text{Scatter Matrix } \mathbf{S}}.$$

Σ is *symmetric*.

Now the *Eigenvalue decomposition* can be computed:

$$\Sigma = U \Lambda U^T,$$

where $\Lambda = \text{diag}[\lambda_1, \dots, \lambda_D]$, such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_D$ with orthonormal eigenvectors.

Transformation the data can be transformed on to the new basis of K dimensions:

$$\tilde{\bar{Z}} = U_K^T \bar{X},$$

$\bar{Z} \in \mathbb{R}^{K \times N}$: We obtain a dimension reduction of the data.

Reconstruction Go back to the original basis by computing

$$\begin{aligned} \tilde{\tilde{X}} &= U_K \tilde{\bar{Z}} \\ \tilde{X} &= \tilde{\tilde{X}} + M \end{aligned}$$

3. Singular Value Decomposition

3.1. Introduction

The *Singular Value Decomposition* (SVD) is a widely used technique to decompose a matrix into several component matrices exposing many of the useful and interesting properties of the original matrix like rank, null-space, orthogonal basis of column and row space.

Every rectangular, real or complex matrix S has an SVD decomposition into a set of three matrix factors.

Let A be any real M by N matrix, $A \in \mathbb{R}^{M \times N}$, then A can be decomposed as $A = UDV^T$:

$$\begin{array}{ccccccc}
 \boxed{\mathbf{A}} & = & \boxed{\mathbf{U}} & \cdot & \boxed{\mathbf{D}} & \cdot & \boxed{\mathbf{V}^T} \\
 M \times N & & M \times M & & M \times N & & N \times N
 \end{array}$$

- U is an $M \times M$ orthogonal matrix, $U^T U = I$
- D is an $M \times N$ diagonal matrix
- V^T is an $N \times N$ orthogonal matrix, $V^T V = I$

3.2. Singular values

The elements of D are only non-zero on the diagonal and are called the *singular values*. By convention, the order of the singular vectors is determined by the *high-to-low* sorting of singular values, with the highest singular value in the upper left index of the D matrix. The first r columns of U are called *left singular vectors*, they form an orthogonal basis for the space spanned by the columns of the original matrix A . Similarly the first r rows of V^T are the *right singular vectors*, they form an orthonormal basis for the row space of A .

SVD provides an explicit representation of the range and null-space of a matrix A .

- The right side singular vectors corresponding to vanishing singular values of A , span the null space of A :

$$d_i = 0 \implies Av_i = 0 \implies v_i \in \text{Null}(A).$$

- The left singular vectors corresponding to the non-zero singular values of A span the range of A .

As a consequence, the rank of A equals the number of non-zero singular values (= the number of non-zero elements in D).

$$\text{Rank}(A) = \#d_i > 0.$$

3.3. Closest Rank- k Matrix

Let the SVD of $A \in \mathbb{R}^{M \times N}$ be given by $A = UDV^T$. If $k < r = \text{Rank}(A)$ and

$$A_k = \sum_{i=1}^k d_i u_i v_i^T.$$

Then

$$\min_{\text{Rank}(B)=k} \|A - B\|_2 = \|A - A_k\|_2.$$

This means that A_k is the closest $\text{Rank}(k)$ approximation to A in the Eculidean matrix norm sense hence:

$$\|A - A_k\|_2 = d_{k+1}.$$

3.4. Properties

The columns of U are the eigenvectors of AA^T . This claim can be verified using the SVD decomposition:

$$AA^T = UDV^TVDU^T = UD^2U^T.$$

Similarly the rows of V^T (or columns of V) are the eigenvectors of A^TA as:

$$A^TA = VDU^TUDV^T = VD^2V^T.$$

3.5. Movie Example

Let A be a list of users with their respective movie preferences. Then the SVD decomposition

$$A = UDV^T,$$

can be interpreted in the following way:

- **U**: Users-to-concept affinity matrix.
- **D**: Expression level of the different concepts in the data.
- **V**: Movies-to-concept similarity matrix.

	<i>Cremators</i>	<i>Evil spawn</i>	<i>Fatal justice</i>	<i>Clerks</i>	<i>American pie</i>
5	5	5	0	0	
4	4	4	0	0	
5	5	5	0	0	
3	3	3	0	0	
0	0	0	4	4	
0	0	0	5	5	
0	0	0	4	4	

 $=$

0.57	0
0.46	0
0.57	0
0.34	0
0	0.52
0	0.66
0	0.52

 \times

15	0
0	10.67

 \times

0.57	0.57	0.57	0	0
0	0	0	0.70	0.70

Part II.

Clustering

4. Introduction

A set of datapoints in a d -dimensional Euclidean space is given.

Aim The aim is to find a *meaningful partition* of the data; i.e. label each data point with a unique value $\{1, \dots, k\}$.

Objective The partition should group together similar data points, while the different groups/clusters should be as dissimilar as possible from each other.

This way we can uncover similarities between data points and give rise to data compression schemes.

4.1. Problem

Consider N data points in a D -dimensional space. Each data vector is denoted by x_n , $n = 1, \dots, N$. Our goal is to partition the data set into K clusters: Find vectors u_1, \dots, u_K that represent the centroid of each cluster.

A datapoint x_n belongs to cluster k if the Euclidean distance between x_n and u_k is smaller than the distance to any other centroid.

Mathematically, the clustering problem defines a mixed discrete continuous optimisation problem.

4.1.1. The Cost Function of Vector Quantisation

Objective Minimise the cost function

$$J(U, Z) = \|X - UZ\|_F^2 = \sum_{n=1}^N \sum_{k=1}^K z_{k,n} \|x_n - u_k\|_2^2$$

where

$$\begin{aligned} X &= [x_1, \dots, x_N] \in \mathbb{R}^{D \times N} \\ U &= [u_1, \dots, u_K] \in \mathbb{R}^{D \times K}, && \text{centroids} \\ Z &\in \{0, 1\}^{K \times N}, && \text{assignments} \end{aligned}$$

with $\sum_k z_{k,n} = 1 \forall n$ i.e., one element per columns set to 1.

Assignment notation:

Assignment Notation : Vector $\hat{z} \in \{1, \dots, K\}^N$ indicating for each data point to which cluster index it is assigned:

$$\hat{z} = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 2 \\ 2 \\ 1 \end{pmatrix}$$

Matrix Notation : The matrix $Z \in \{0, 1\}^{K \times N}$ with only one non-zero entry per column, assigns data points to clusters:

$$Z = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

5. K -Means

5.1. Overview

The algorithm alternates between two steps:

- *Assigning data points to clusters* Let $k^*(x_n)$ denote the cluster index with the minimal distance between a cluster centroid and the data point x_n :

$$k^*(x_n) = \arg \min_k \left\{ \|x_n - u_1\|_2^2, \dots, \|x_n - u_k\|_2^2, \dots, \|x_n - u_K\|_2^2 \right\}$$

- *Updating the cluster centroids* based on all the data points assigned to it. Compute the mean/centroid of a cluster that can be written as:

$$u_k = \frac{\sum_{n=1}^N z_{k,n} x_n}{\sum_{n=1}^N z_{k,n}} \quad \forall k, \quad k \in \{1, \dots, K\}$$

5.2. Algorithm

1. Initiate with a random choice of $u_1^{(0)}, \dots, u_K^{(0)}$ (or let $u_1^{(0)}, \dots, u_K^{(0)}$ equal data points from the set). Set $t = 1$.
2. **Cluster assignment.** Solve $\forall n$:

$$k^*(x_n) = \arg \min_k \left\{ \|x_n - u_1^{(t)}\|_2^2, \dots, \|x_n - u_K^{(t)}\|_2^2 \right\}.$$

Then, $z_{k^*(x_n), n}^{(t)} = 1$ and $z_{j,n}^{(t)} = 0 \quad \forall j \neq k, \quad j = 1, \dots, K$.

3. Centroid update

$$u_k^{(t)} = \frac{\sum_{n=1}^N z_{k,n}^{(t)} x_n}{\sum_{i=1}^N z_{k,n}^{(t)}} \quad \forall k, \quad k \in \{1, \dots, K\}$$

4. Increment t . Repeat step 2 until $\left\|u_k^{(t)} - u_k^{(t-1)}\right\|_2^2 < \varepsilon \quad \forall K \quad (0 < \varepsilon \ll 1)$ or until $t = t_{\text{finish}}$.

Aspects:

- The computational cost of each iteration is $\mathcal{O}(KN)$.
- Convergence is guaranteed
- Optimises a *non-convex* objective. Hence only a local minimum can be guaranteed.
- Can be used to compress data: store only the centroids and the assignments of data point to clusters.

Problems:

- Non-convex objective, local minima and sensitive to initialisations.
- Not appropriate for non-Euclidean data \mapsto need to use other distances.
- The optimal number of clusters K is unknown: One has to find a balance between total compression ($K = 1$) and no loss of information ($K = N$).

5.3. Stability

5.3.1. High-Level Stability test

The following is a high-level stability test for a given set of data points and a given number of clusters:

1. Generate perturbed versions of the set for example by adding noise or drawing sub-samples.
2. Apply the clustering algorithm on all versions.
3. Compute pair-wise distances between all clusterings (using some distance measure).
4. Compute the *instability* as the mean distance between all clusterings.

Repeat this for different numbers of clusters and choose the one that minimises the instability.

5.3.2. Distance between Clusterings

For two clusterings C and C' that are defined on the same data points we compute the distance between clusterings d in the following procedure:

1. Compute the distances between the two clusterings by counting points on which the two clusterings agree or disagree.
2. Repeat over all permutations of the cluster labels (since the same cluster might be sometimes labeled 1 and sometimes 2 etc...).
3. Choose the permutation with minimal distance and the corresponding distance is d .

In other words,

$$d = \min_{\pi} \|Z - \pi(Z')\|_0$$

where $\pi(Z')$ is one of the possible row permutations of Z' and $\|Z\|_0$ denotes the cardinality of Z . If two clusterings are defined on different data sets but many points overlap, we use only these for comparison, otherwise, a mapping from one domain to the other is required.

5.3.3. Calculation of Stability

The rate of inconsistent data items r is computed as follows”

1. Cluster data sets X, X' to infer assignments Z, Z' .
2. Train a classifier φ on (X, Z) to transfer the clustering results Z on X to X' .
3. Apply φ on X' and compare the optimally permuted output with Z' :

$$r := \frac{1}{N} \min_{\pi \in \mathbb{S}_K} \left\{ \sum_{i=1}^N \mathbb{I}_{\{\pi(\varphi(x'_i)) \neq z'_i\}} \right\}.$$

The indicator function $\mathbb{I}_{\{p\}}$ is 1 if predicate p is true, and 0 otherwise.

Minimisation of $\pi \in \mathbb{S}_K$ compensates for the permutation of the cluster numbers.

The higher the number of clusters, the more difficult it is to have a small rate r of inconsistent cluster assignments. Given K clusters of equal size, a random assignment yields

$$r_{rand} = \frac{K-1}{K}.$$

To be able to compare hypotheses with different K , relate r to r_{rand} . The *stability* is this defined as:

$$stab := 1 - \frac{r}{r_{rand}}.$$

- $stab = 1$: No inconsistent assignments
- $stab = 0$: Not better than a random assignment

6. Clustering as Matrix Factorisation

SVD is a class matrix factorisation technique according to which every matrix matrix can be decomposed into $X = UDV^T$. With $U \in \mathbb{R}^{D \times D}$, $D \in \mathbb{R}^{D \times N}$ and $V \in \mathbb{R}^{N \times N}$. By setting UD in the decomposition as U and renaming V^T to Z , we can write

$$X = UZ.$$

Approximating X using the K largest singular values we get a factorisation involving matrices of the same dimensionality as K -Means.

7. Mixture Models

Soft Clustering The term Soft Clustering is ambiguous since it can refer to the algorithm or to the model:

Algorithmic: Soft K -Means: Instead of assigning a point to exactly one cluster. Consider assigning a probability that a data point belongs to a certain cluster.

Model relaxation: The model can be relaxed by replacing the "hard" constraint given by

$$z_{k,n} \in \{0, 1\}, \quad \sum_{k=1}^K z_{k,n} = 1,$$

and replace it by the *soft constraint*:

$$z_{k,n} \in [0, 1], \quad \sum_{k=1}^K z_{k,n} = 1.$$

This relaxed model **cannot** be written in Matrix factorisation form.

7.1. Introduction

The mixture of K probability densities is defined as

$$p(x) = \sum_{k=1}^K \pi_K p(x|\theta_k).$$

Each probability distribution $p(x|\theta_k)$ is a *component* of the mixture and has its own parameters θ_k . Almost any continuous density can be approximated by using a sufficient number of component distributions. For a Gaussian component distribution the parameters θ_k are given by the mean μ_k and the covariance Σ_k .

Mixture models are constructed from:

- Component distributions of the form $p(x|\theta_k)$.

- Mixing coefficients π_k that give the probability of each component.

In order for $p(x)$ to be a proper distribution, we have to ensure that

$$\sum_{k=1}^K \pi_k = 1 \quad \text{and} \quad \pi_k \geq 0, \quad 1 \leq k \leq K.$$

Therefore, the parameters π_k , $1 \leq k \leq K$ define a *categorical* distribution which represents the probability of each component distribution.

7.2. Gaussian Mixture Model

The Gaussian Mixture Model (GMM) uses Gaussians as the component distributions. We thus assume the following distribution for our data:

$$p(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x | \mu_k, \Sigma_k).$$

Given data points $\{x_1, \dots, x_N\}$, we then aim at learning parameters μ_k, Σ_k , such that we approximate the data as closely as possible. This is equivalent to finding the parameters that *maximise the likelihood* of the data.

7.2.1. Generative Viewpoint

Given the model parameters Σ, μ, π , sample the data x_n as follows:

1. Sample a cluster index k according to the probabilities π_k .
2. Sample a data point x_n from the distribution $p(x_n | \mu_k, \Sigma_k)$.

7.2.2. Parameter Estimation

We assume that the data points x_n are independent and identically distributed (i.i.d.). The probability or likelihood of the observed data X , given the parameters can thus be written as:

$$p(X | \pi, \mu, \Sigma) = \prod_{n=1}^N p(x_n) = \prod_{n=1}^N \sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k).$$

We aim at finding the parameters that maximise the likelihood of the data:

$$(\hat{\pi}, \hat{\mu}, \hat{\Sigma}) \in \arg \max_{\pi, \mu, \Sigma} p(X | \pi, \mu, \Sigma).$$

To simplify the expression one takes the logarithm, such that the product becomes a sum:

$$(\hat{\pi}, \hat{\mu}, \hat{\Sigma}) \in \arg \max_{\pi, \mu, \Sigma} \sum_{n=1}^N \log \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right\}.$$

Due to the presence of the summation over k inside the logarithm, the maximum likelihood solution for the parameters no longer has a closed-form analytic solution. In order to solve the equation we make use of an algorithmic approach: *Expectation Maximisation*.

7.2.3. Probability of assigning a data point to a cluster

We use the Bayes' rule to get:

$$\begin{aligned}\gamma(z_k) &:= p(z_k = 1|x) = \frac{p(z_k = 1)p(x|z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(x|z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)}.\end{aligned}$$

7.2.4. Expectation-Maximisation for Gaussian Mixture Models

A powerful method for finding maximum likelihood solutions for models with latent variables. Setting the derivatives of $\log p(X|\pi, \mu, \Sigma)$ with respect to the means μ_k to zero, we obtain:

$$0 = \sum_{n=1}^N \underbrace{\frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}}_{\gamma(z_{k,n})} \Sigma_k^{-1} (x_n - \mu_k).$$

Assume that Σ_k is not singular. Multiplying by Σ_k we obtain:

$$\begin{aligned}\sigma_k &= \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{k,n}) x_n, \\ N_k &= \sum_{n=1}^N \gamma(z_{k,n}).\end{aligned}$$

The mean μ_k is obtained by taking a weighted mean of all the points in the data set. Maximising $\log p(X|\pi, \mu, \Sigma)$ with respect to the mixing coefficients π_k and taking account of the constraint which requires the mixing coefficients to sum to one, can be achieved using the following Lagrangian:

$$\log p(X|\pi, \mu, \Sigma) + \lambda \left(\sum_{k=1}^K \pi_k - 1 \right),$$

which gives

$$\begin{aligned}0 &= \sum_{n=1}^N \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} + \lambda, \\ &\implies \pi_k = \frac{N_k}{N}.\end{aligned}$$

Therefore, π_k is given by the average responsibility of the k -th component.

7.2.5. Algorithm

Given a Gaussian mixture model, the goal is to maximise the likelihood function with respect to the parameters.

1. Initialise the means μ_k , and mixing coefficients π_k . Set the Σ_k to the given covariances.
2. **E-step.** Evaluate the responsibilities using the current parameter values:

$$\gamma(z_{k,n}) = \frac{\pi_k \mathcal{N}(x | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x | \mu_j, \Sigma_j)}.$$

3. **M-step.** Re-estimate the parameters using the current responsibilities

$$\begin{aligned} \mu_k^{new} &= \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{k,n}) x_n, \\ \Sigma_k^{new} &= \frac{1}{N_k} \sum_{i=1}^N \gamma(z_{k,n}) \cdot (x_i - \mu_k^{new})(x_i - \mu_k^{new})^T \quad 1 \leq k \leq K, \\ \pi_k^{new} &= \frac{N_k}{N} \quad \text{where } N_k = \sum_{n=1}^N \gamma(z_{k,n}). \end{aligned}$$

4. Evaluate the log likelihood and check for convergence of either the parameters or the log likelihood.

7.2.6. Relation to K -means

The EM algorithm makes a soft assignment based on posterior probabilities². Whereas the K -means algorithm performs a hard assignment of data points to clusters.

The K -means algorithm does not estimate the covariances of the clusters but only the cluster means. The EM algorithm can be reduced to K -means.

The EM algorithm takes more iterations to converge than K -means.

The K -means algorithm can be used to find a suitable initialisation for a Gaussian mixture model.

There will generally be multiple local maxima of the log likelihood function, and EM is not guaranteed to find the largest of these maxima.

7.2.7. Log Likelihood

$$\mathbb{E}_Z[\log p(X, Z | \mu, \Sigma, \pi)] \rightarrow -\frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K z_{k,n} \|x_n - \mu_k\|_2^2 + \text{const.}$$

²The posterior probability is the probability of the parameters θ given the evidence $X : p(\theta | X)$ (Wikipedia)

7.2.8. Matrix Factorisation

Assumption: $z_{k,n} \in [0, 1]$ and $\sum_{k=1}^K z_{k,n} = 1 \forall n$.

$$\begin{aligned} \|X - UZ\|_F^2 &= \sum_{n=1}^N \left\| x_n \sum_{k=1}^K z_{k,n} u_k \right\|_2^2 \\ &= \sum_{n=1}^N \left(\|x_n\|^2 - 2 \sum_{k=1}^K x_n u_k z_{k,n} + \left(\sum_{k=1}^K z_{k,n} u_k \right)^2 \right) \\ &= \sum_{n=1}^N \sum_{k=1}^K z_{k,n} \|x_n - u_k\|_2^2 - \sum_{n=1}^N \sum_{k=1}^K z_{k,n} \left(u_k - \sum_{k'=1}^K z_{k',n} u_{k'} \right)^2 \end{aligned}$$

For identity covariance matrices GMM gives an upper bound!

8. Model Order Selection and Information Criteria

The selection of the number of clusters is a trade-off between two conflicting goals:

Data Fit: We want to predict the data well, e.g., maximises the likelihood. The likelihood usually increases by increasing the clusters.

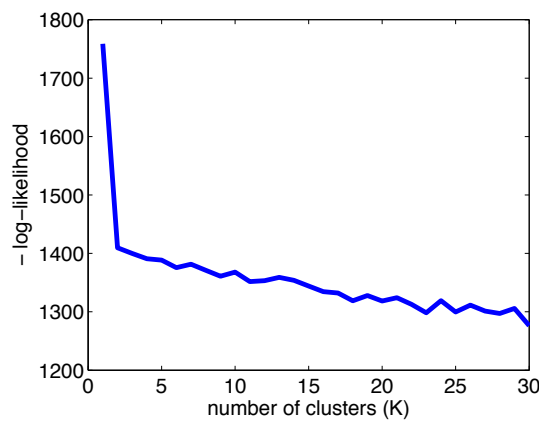
Complexity: Choose a model that abstracts a lot of data. This is often measured by the number of free parameters.

8.0.9. Negative Log-Likelihood

The smaller the negative log-likelihood, the better the fit.

$$-\log p(X|\pi, \mu, \Sigma) = -\sum_{n=1}^N \log \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right\}$$

In practice increasing K does not always decrease the negative log-likelihood due to local minima.



8.1. AIC and BIC

Achieve balance between data fit (measured by likelihood $p(X|\cdot)$) and complexity. Complexity can be measured by the number of free parameters $\kappa(\cdot)$.

8.1.1. AIC - Akaike Information Criterion

$$AIC(U, Z|x_1, \dots, x_N) = -\log(p(X|\cdot)) + \kappa(U, Z)$$

8.1.2. BIC - Bayesian Information Criterion

$$BIC(U, Z|x_1, \dots, x_N) = -\log(p(X|\cdot)) + \frac{1}{2}\kappa(U, Z) \log N$$

Generally speaking, the BIC criterion penalises complexity more than the AIC criterion. A single AIC (BIC) result is meaningless. One has to repeat the analysis for different K s and compare the differences: the most suitable number of clusters corresponds to the smallest AIC (BIC) value.

Example Mixture of Gaussians with fixed covariance. Number of free parameters:

$$\kappa(U, Z) = K \cdot D + (K - 1).$$

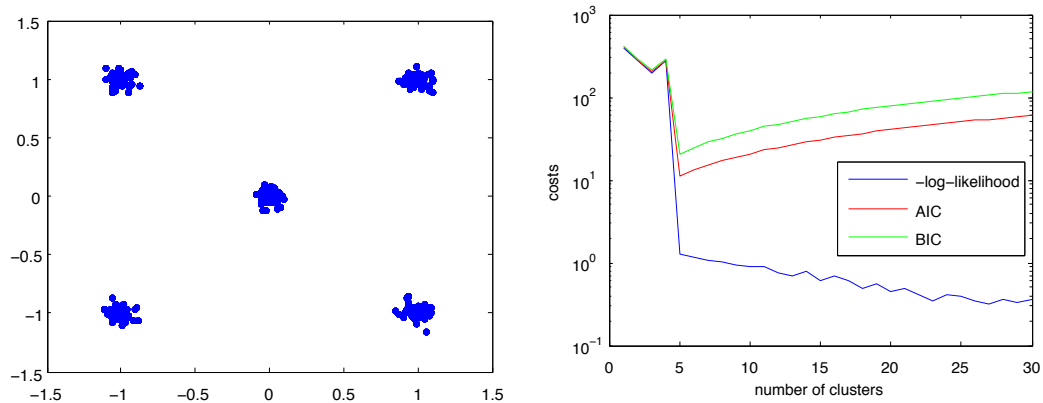


Figure 1: Comparison of AIC, BIC and negative Log-Likelihood on a synthetic dataset with 5 clusters.

Appendix

A. Matrix Definitions and Theorems

A.1. Norms

A *norm* is a function $\|\cdot\| : V \mapsto \mathbb{R}$ quantifying the size of a vector. It must satisfy

- *Positive scalability*:

$$\|a \cdot x\| = |a| \cdot \|x\|.$$

- *Triangle inequality*

$$\|x + y\| \leq \|x\| + \|y\| \quad \forall x, y \in V.$$

- *Separability*:

$$\|x\| = 0 \implies x = 0.$$

A.1.1. Vector norms

p-norms The most commonly used matrix norms are *p*-norms.

$$\|x\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$$

for $p \in [1, \infty]$, where $|x_i|$ denotes the absolute value of coordinate x_i .

A special case of the *p* norm is the *Euclidean norm*:

$$\|x\|_2 := \sqrt{\sum_{i=1}^n x_i^2}.$$

0-norm technically not really a norm is defined by:

$$\|x\|_0 := \text{number of nonzero coordinates in } x.$$

A.1.2. Matrix norms

p-norm for matrices:

$$\|X\|_p := \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}.$$

A special case is the Euclidean or *spectral norm*:

$$\|X\|_2 = \sigma_{\max}(X),$$

the largest singular value of X .

Frobenius norm is defined as:

$$\|X\|_F := \sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{ij}^2} = \sum_{i=1}^{\min(m,n)} \sigma_i^2,$$

where σ_i are the singular values of X .

A.2. Orthogonality

Orthogonal vectors Two vectors in an inner product are orthogonal if their inner product is zero.

Orthonormal vectors Orthogonal vectors that have unit length 1

Orthogonal matrix An orthogonal matrix is a square matrix with real entries whose columns and rows are orthogonal unit vectors (i.e. orthonormal vectors). For orthogonal matrices it also holds that

$$A^T A = I \implies A^T = A^{-1} \text{ since,}$$

$$(A^T A)_{i,j} = a_i^T a_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

B. Occam's Razor

Entities must not be multiplied beyond necessity.

It states that among competing hypotheses, the hypothesis with the fewest assumptions should be selected. It is often understood as 'the simplest explanation is usually the correct one', although this is potentially misleading. The application of the principle often shifts the burden of proof in a discussion.[a] The razor states that one should proceed to simpler theories until simplicity can be traded for greater explanatory power. The simplest available theory need not be most accurate. Philosophers also point out that the exact meaning of simplest may be nuanced.³

C. Probability

We denote Ω the sample space and by A an event, which is a subset of Ω .

Probability distribution A function p that assigns a real number $p(A)$ to each event A is a *probability distribution* if it satisfies the following three axioms:

- $p(A) \geq 0$ for every A .
- $p(\Omega) = 1$.

³http://en.wikipedia.org/wiki/Occam's_razor

- If A_1, A_2, \dots are disjoint then

$$p\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} p(A_i)$$

Random Variable A random variable is a mapping:

$$\begin{aligned} X &: \Omega \rightarrow \mathcal{K} \\ \omega &\mapsto X(\omega) \end{aligned}$$

that assigns an element $X(\omega) \in \mathcal{K}$ to each outcome ω .

Notation and types of random variables (sample spaces):

Notation :

X Random variable
 x a value taken by the r.v. X .

Discrete random variables :

- Finite: e.g. $X \in \mathcal{B} \equiv \{0, 1\}$ or $X \in \mathcal{S}_n$ (set of permutations)
- Countably Infinite: e.g. $X \in \mathbb{N}, \mathbb{Z}$ etc...

Continuous random variables : e.g. $X \in \mathbb{R}, [a, b]$ etc...

C.1. Distributions

C.1.1. Categorical distribution

Multinomial distribution.

Example The sample space of throwing two dice is $\Omega = [1, \dots, 6] \times [1, \dots, 6]$. We consider the two random variables $X_1 + X_2$ given by summing up the numbers x_1, x_2 of the two dice. The random variable can thus take values in $[2, \dots, 12]$. This leads to the following probabilities:

Random variable	2	3	4	5	6	7	8	9	10	11	12
Probability	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$

The random variable $X_1 + X_2$ is an example of the *categorical distribution*: A discrete distribution over K events. Each event has the probability π_k of occurring.

Definition The categorical distribution is a discrete probability distribution whose sample space is the set of 1-of- K encoded random vectors x of dimensions K having the property:

$$\sum_{k=1}^K z_k = 1, \quad z_k \in \{0, 1\}.$$

The probability mass function is defined as

$$p(z|\pi) = \prod_{k=1}^K \pi_k^{z_k},$$

where π_k represents the probability of seeing element k ($\pi_k \geq 0$, $\sum_{k=1}^K \pi_k = 1$).

C.1.2. Gaussian Distribution

The probability density function of the Gaussian distribution is given by:

$$p(x|\mu, \sigma) = \mathcal{N}(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

$\mu =$ mean of distribution
 $\sigma^2 =$ variance of distribution.

Probability for $X \in [a, b]$ is given by an integral:

$$P(a < X < b) = \int_a^b p(x)dx = \frac{1}{\sqrt{2\pi}\sigma} \int_a^b \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} dx.$$

C.1.3. Multivariate Gaussian Distribution

Generalises the univariate Gaussian distribution to higher dimensions. The distribution samples the space $\mathcal{X} \subseteq \mathbb{R}^D$. A random vector $X = (X_1, \dots, X_D)^T$ has a multivariate normal distribution if every linear combination of its components (i.e., $Y = a_1 X_1 + \dots + a_D X_D$) has a univariate normal distribution.

Definition :

$$p(x|\mu, \Sigma) = \mathcal{N}(x|\mu, \Sigma) := \frac{1}{(\sqrt{2\pi})^D |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

C.1.4. Multidimensional Moment Statistics

Expectation Vector of component expectations:

$$\mathbb{E}[X] = \begin{pmatrix} \mathbb{E}[X_1] \\ \vdots \\ \mathbb{E}[X_D] \end{pmatrix}$$

Variance Generalised covariance:

$$\begin{aligned} \text{Cov}[X, Y] &:= \int_{\mathcal{X}} \int_{\mathcal{Y}} p(x, y)(x - \mu_X)(y - \mu_Y) dx dy \\ &= \mathbb{E}_{X, Y}[(x - \mu_X)(y - \mu_Y)] \end{aligned}$$

Covariance Matrix For random variables X_1, \dots, X_D we record covariances in the covariance matrix Σ :

$$\Sigma_{i,j} := \text{Cov}[X_i, Y_j] \quad i, j \in \{1, \dots, D\}.$$

Σ generalises the notion of variance to sets of random variables for multiple dimensions.

C.2. Latent Variables

Define a K -dimensional binary random variable z having a 1-of- K representation in which a particular element z_k is equal to 1 and all other elements are equal to 0, i.e.:

$$z_k \in \{0, 1\}, \quad \sum_k z_k = 1.$$

The marginal distribution over z is specified in terms of the mixing coefficients π_k , i.e.

$$p(z_k = 1) = \pi_k.$$

Because z uses a 1-of- K representation, we can write this distribution in the form:

$$p(z) = \prod_{k=1}^K \pi_k^{z_k}.$$

Similarly, the conditional distribution of x given a particular value for z is a gaussian:

$$p(x|z_k = 1) = \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k}.$$

The conditional distribution can also be written in the form

$$p(x|z) = \prod_{k=1}^K \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k}.$$

The marginal distribution of x is obtained by summing the joint distribution over all possible states of z to yield:

$$p(x) = \sum_z p(z)p(x|z) = \sum_{k=1}^K \pi_k \mathcal{N}(x|\mu_k, \Sigma_k).$$

For the full data log-likelihood we get:

$$\log p(X|\pi, \mu, \Sigma) = \sum_{n=1}^N \log \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}.$$

C.3. Bayes' rule

The conditional probability of A given B (posterior) is given by:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}.$$

We call $p(A)$ prior, $p(B|A)$ likelihood and $p(B)$ evidence.

D. Lagrange Multipliers

This part was taken from Wikipedia⁴.

In mathematical optimisation the method of Lagrange multipliers is a strategy for finding the local maxima and minima of a function subject to *equality constraints*.

Consider the two-dimensional problem:

$$\begin{aligned} &\text{maximise } f(x, y) \\ &\text{subject to } g(x, y) = c, \end{aligned}$$

with both f and g to having *continuous first partial derivatives*.

We can visualise contours of f given by

$$f(x, y) = d,$$

for various values of d , and the contour of g given by

$$g(x, y) = c.$$

Suppose we walk along the contour line $g = c$. In general the contour lines of f and g may be distinct, so following the contour line for $g = c$ one could intersect with or cross the contour lines of f . Only when the contour line for $g = c$ meets contour lines of f *tangentially*, do we not increase or decrease the value of f (in terms of maximising $f(x, y)$ subject to $g(x, y) = c$).

The contour lines of f and g touch when the *tangent vectors* of the contour lines are parallel. Thus we want points (x, y) where $g(x, y) = c$ and

$$\nabla_{x,y} f = -\lambda \nabla_{x,y} g.$$

To solve this system we introduce an auxiliary function:

$$\Lambda(x, y, \lambda) = f(x, y) + \lambda \cdot (g(x, y) - c),$$

and solve

$$\nabla_{x,y,\lambda} \Lambda(x, y, \lambda) = 0.$$

⁴http://en.wikipedia.org/wiki/Lagrangian_multiplier