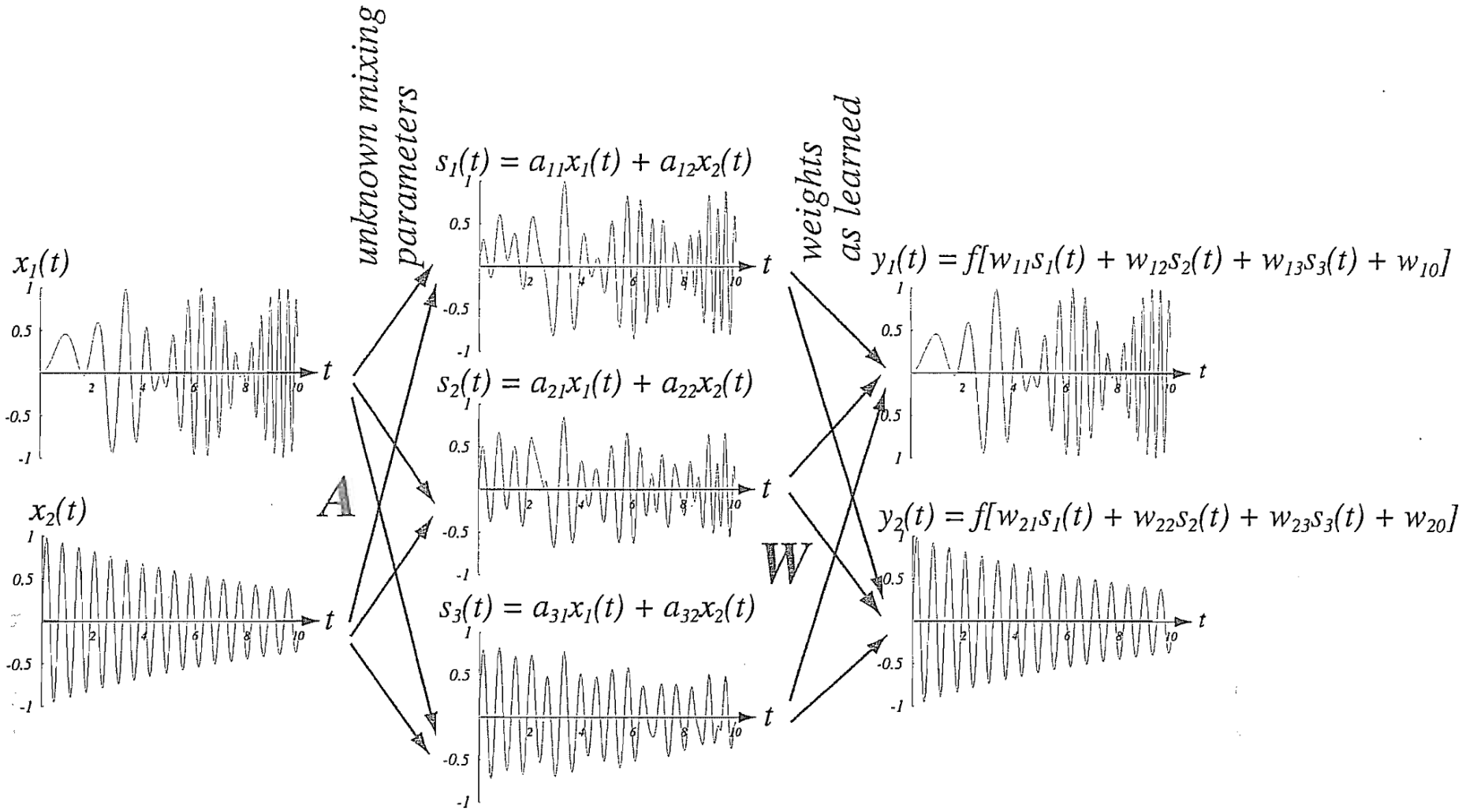
Ch3-part4: Independent Component Analysis (ICA)

- Method for finding underlying components from multi-dimensional data.
- Focus is on Independent and Non-Gaussian components in ICA as compared to uncorrelated and Gaussian components in FA and PCA
- Multiple sensors receiving signals which are mixture of original signals
- Estimate original source signals from mixture of received signals

- Can be viewed as Blind-Source Separation as mixing parameters are not known
- Observe *n* random variables $x_1, x_2, ..., x_n$ which are linear combinations of *n* random variables $s_1, s_2, ..., s_n$ which are mutually independent
- In Matrix Notation, **X = AS**
- If we can estimate **A**, then we can compute **S** by inverting **A**: $\mathbf{y} = \mathbf{W}\mathbf{x}$ *where* $\mathbf{W} = \tilde{\mathbf{A}}^{-1}$
- Assume source signals are statistically independent $x_i = a_{i1}s_1 + a_{i2}s_2 + ... + a_{in}s_n$

- Estimate the mixing parameters and source signals
- Find a linear transformation of observed signals such that the resulting signals are as independent as possible
- Components are assumed independent
- Components must have non-Gaussian densities
- Energies of independent components can not be estimated
- Sign Ambiguity in independent components

Independent Component Analysis (ICA)



d-sources *k*-sensed signals *d*-independent components

Gaussian and Non-Gaussian components

- If some components are Gaussian and some are non-Gaussian.
  - Can estimate all non-Gaussian components
  - Linear combination of Gaussian components can be estimated.
  - If only one Gaussian component, model can be estimated

Why Non-Gaussian Components

- Uncorrelated Gaussian r.v. are independent
- Orthogonal mixing matrix can’t be estimated from Gaussian r.v.
- For Gaussian r.v. estimate of model is up to an orthogonal transformation
- ICA can be considered as non-Gaussian factor analysis

ICA vs. PCA

- PCA
  - Find smaller set of components with reduced correlation. Based on finding *uncorrelated* components
  - Needs only second order statistics
- ICA
  - Based on finding *independent* components
  - Needs higher order statistics

Whitening as Preprocessing for ICA

- Elements are uncorrelated and have unit variances
- Decorrelation followed by scaling
- Any orthogonal transformation of whitened r.v. will be white
- So whitening gives components up to orthogonal transformation.
- Useful as preprocessing step for ICA.
- Search is restricted to orthogonal mixing matrices
- Parameters reduced from n^2 to $n(n-1)/2$

ICA Techniques

- Maximum Likelihood Estimation

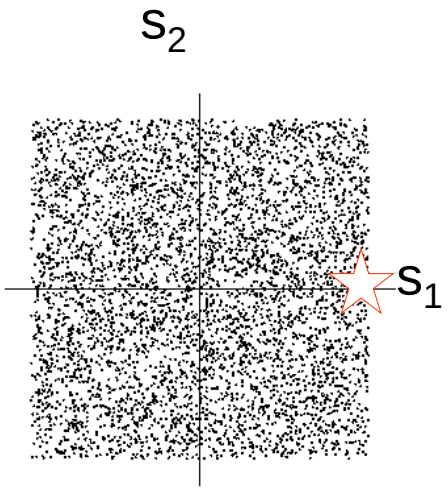
$$L = \sum_{t=1}^{T} \sum_{i=1}^{n} \log\left(f_i(\mathbf{w}_i^T \mathbf{x}(t)\right) + T \log|\mathbf{W}|$$

- Minimization of Mutual Information (Kullback-Leibler Divergence and maximum entropy)

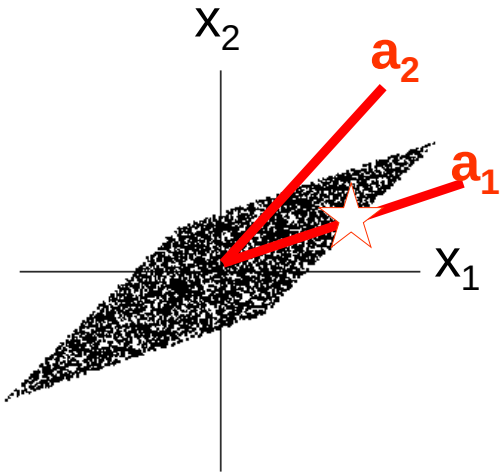
$$I(y_1, y_2, \ldots, y_n) = \sum_{i=1}^{n} H(y_i) - H(\mathbf{y})$$

- Maximization of non-Gaussianity (kurtosis and negentropy)
- Negentropy $J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y})$

Illustration of ICA with 2 signals

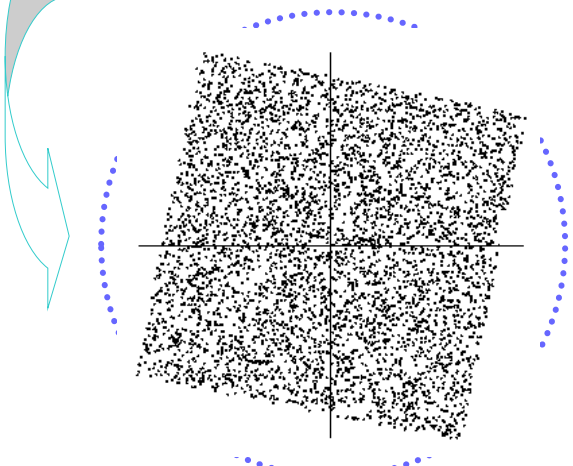


Original s

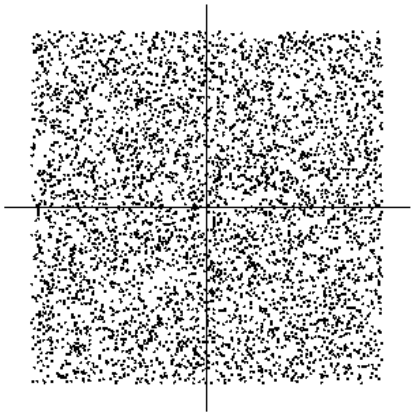


Mixed signals

$$x_1(t) = a_{11}\, s_1(t) + a_{12} s_2(t)$$
$$x_2(t) = a_{21}\, s_1(t) + a_{22} s_2(t)$$
$$\forall t = 1:T$$



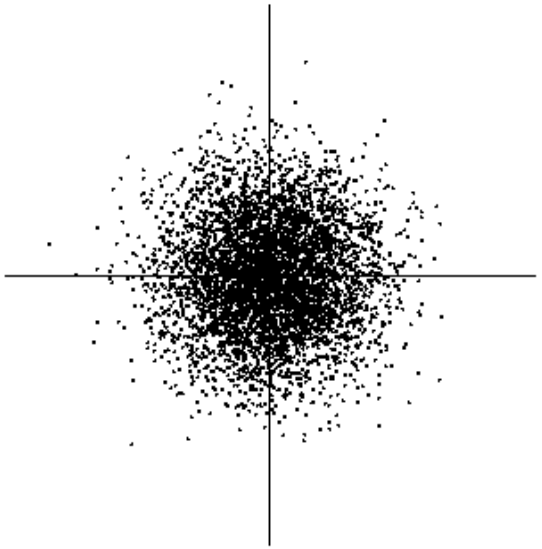
Step1:
Sphering



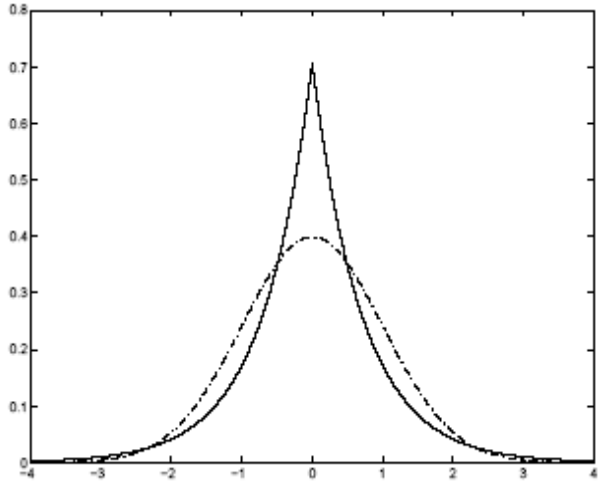
Step2:
Rotatation

Excluded case

There is one case when rotation doesn't matter. This case cannot be solved by basic ICA.



…when both densities are Gaussian

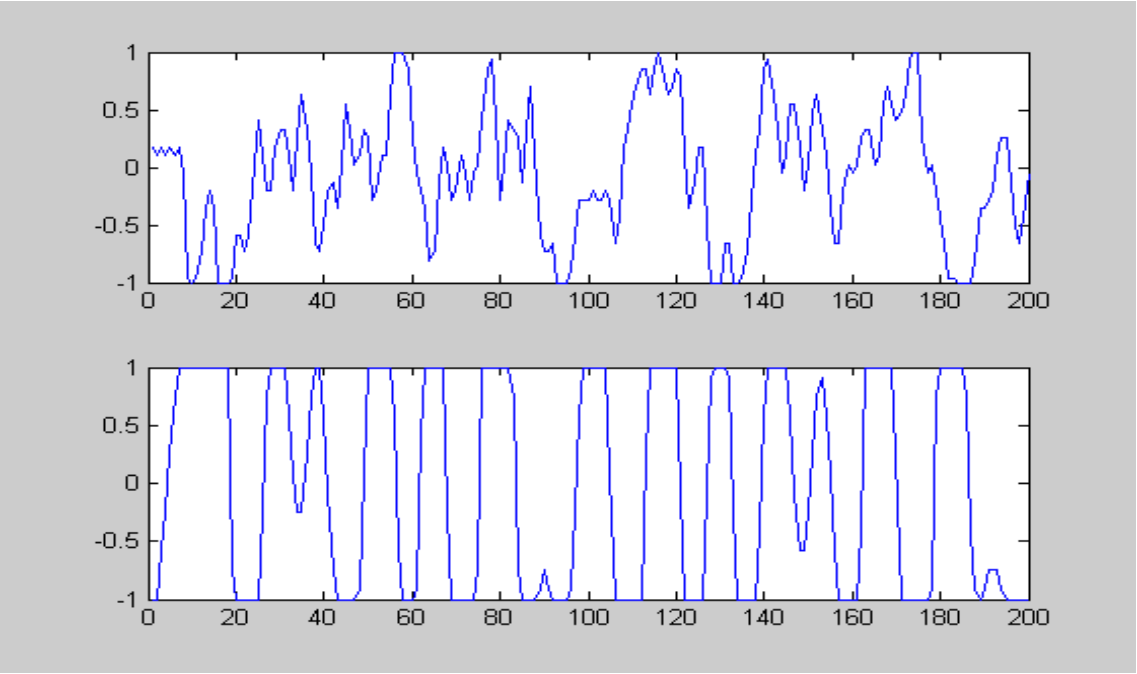


Example of non-Gaussian density (-) vs.Gaussian (-.)

Seek non-Gaussian sources for two reasons:
* identifiability
* interestingness: Gaussians are not interesting since the superposition of independent sources tends to be Gaussian

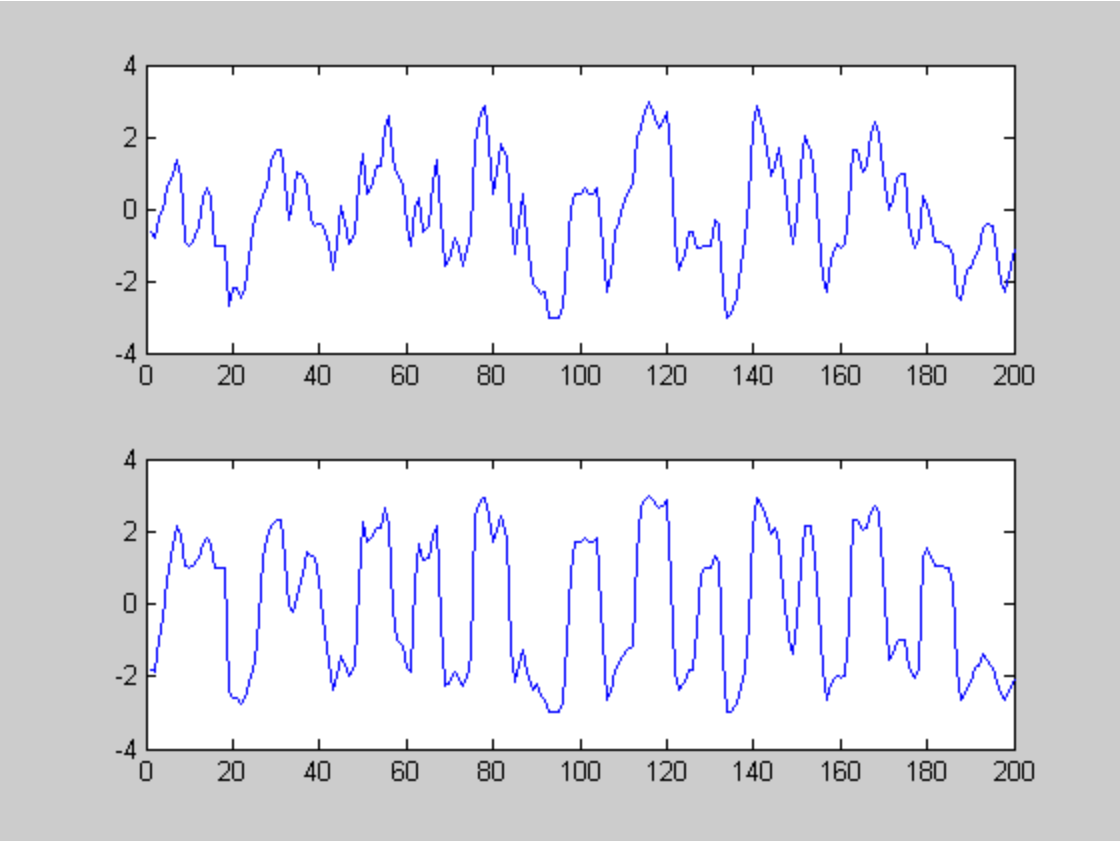
Simple Simulation

- Separation of 2 components
- Figure 1: Two independent non Gaussian wave samples



Simple Simulation

- Figure 2: Mixed signals



Simple Simulation

- Recovered signals vs original signals

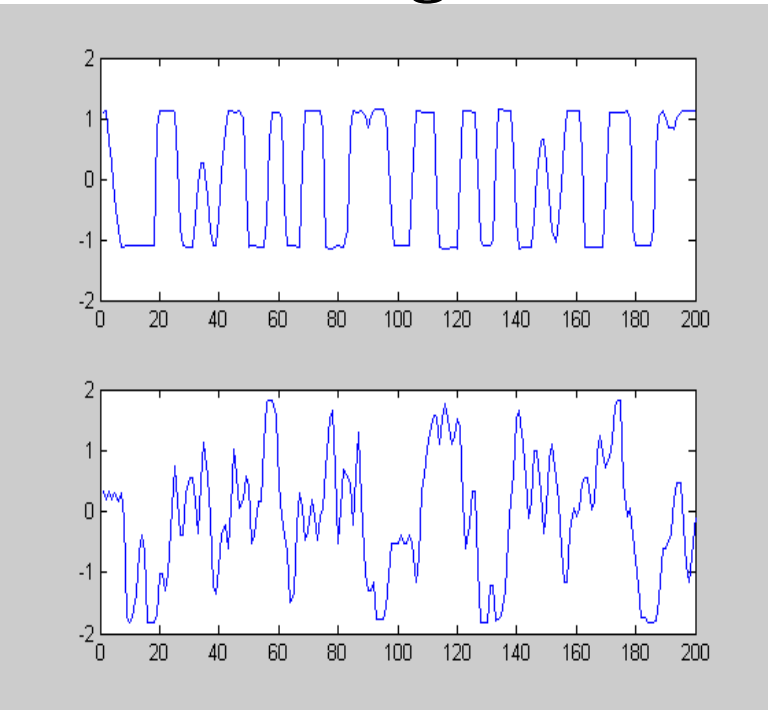


Figure 3: Recovered signals

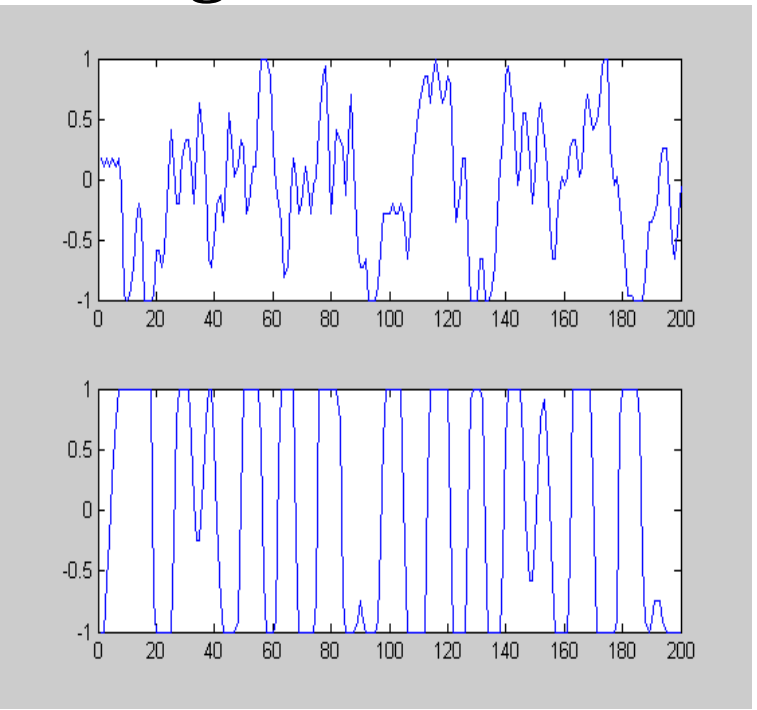


Figure 4: Original signals

Gaussian Simulation

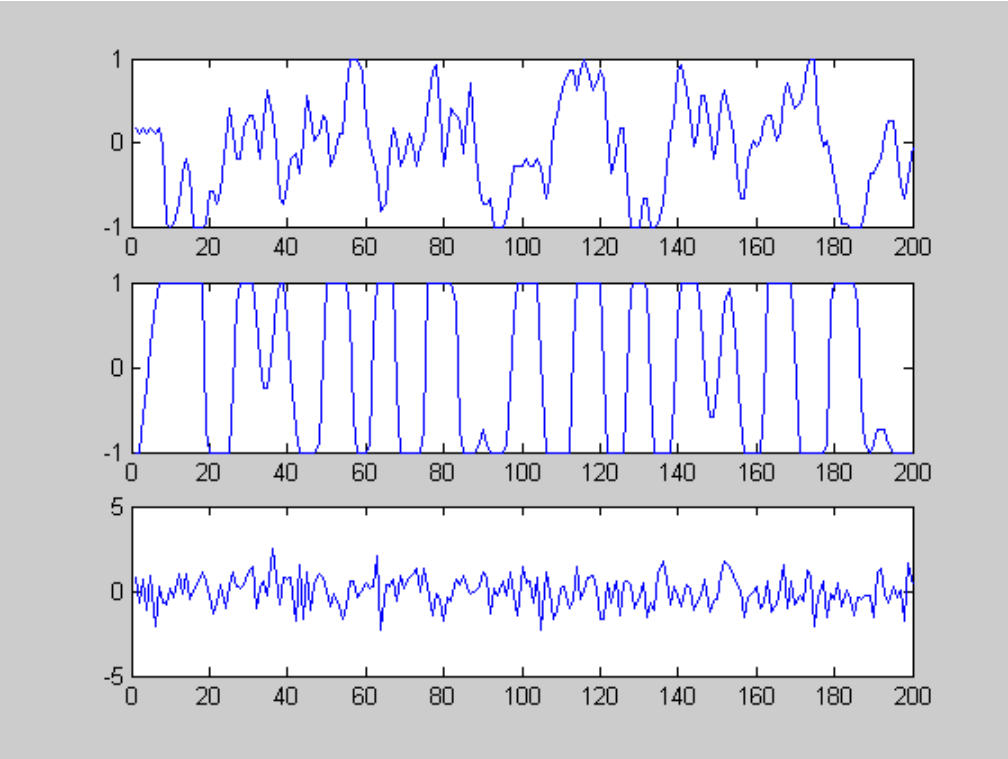


Figure 5: 2 wave samples and noise signal

Gaussian Simulation

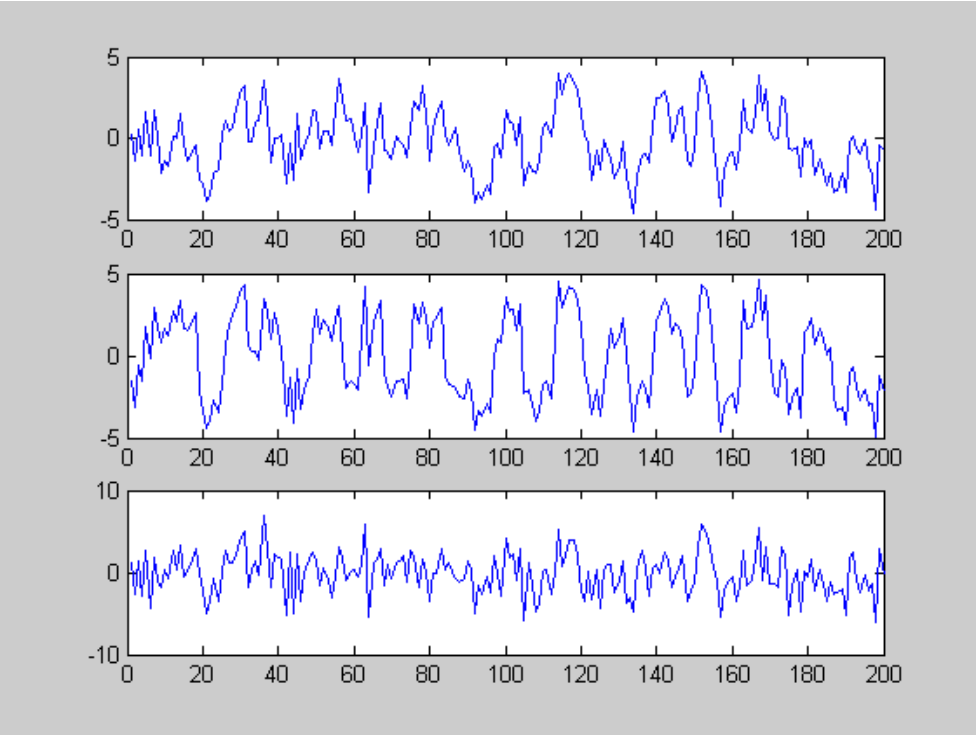


Figure 6: 3 mixed signals

Gaussian Simulation

- Comparison of recovered signals vs original signals

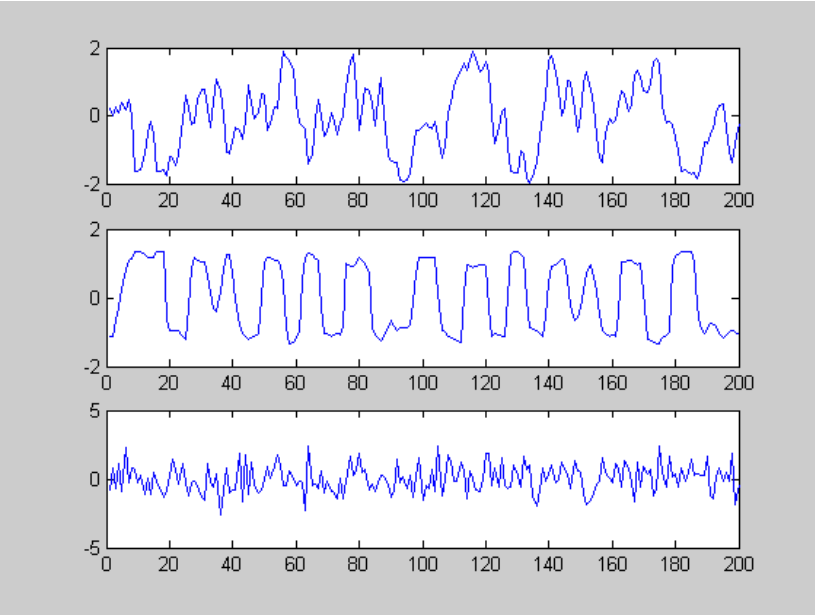


Figure 7: Recovered signals

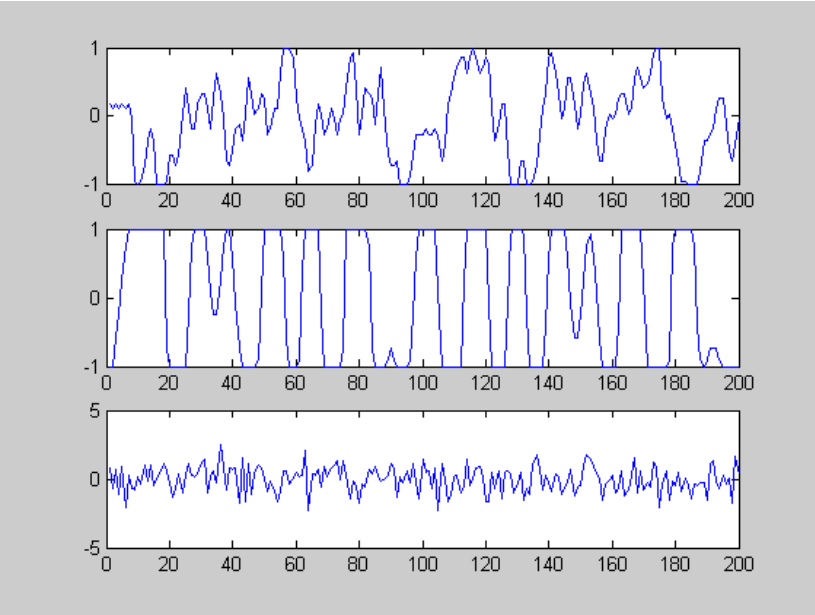


Figure 8: Original signal

Gaussian Simulation 2:

- Tried with 2 Gaussian components

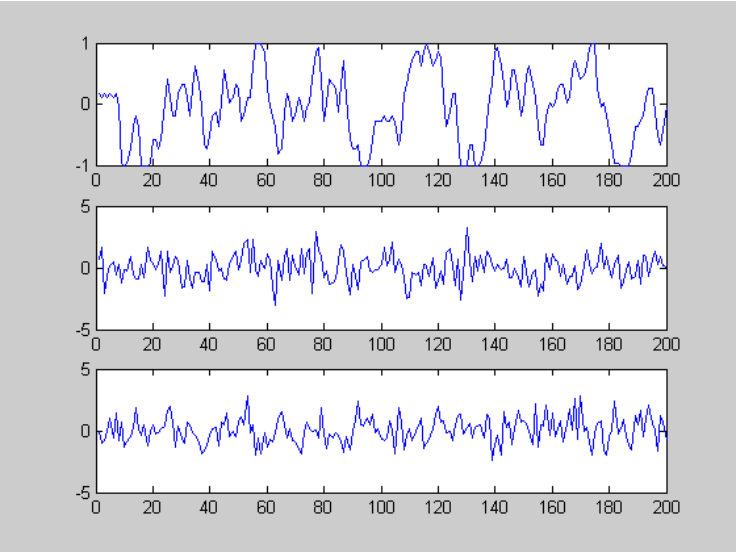


Figure 10: Original signals

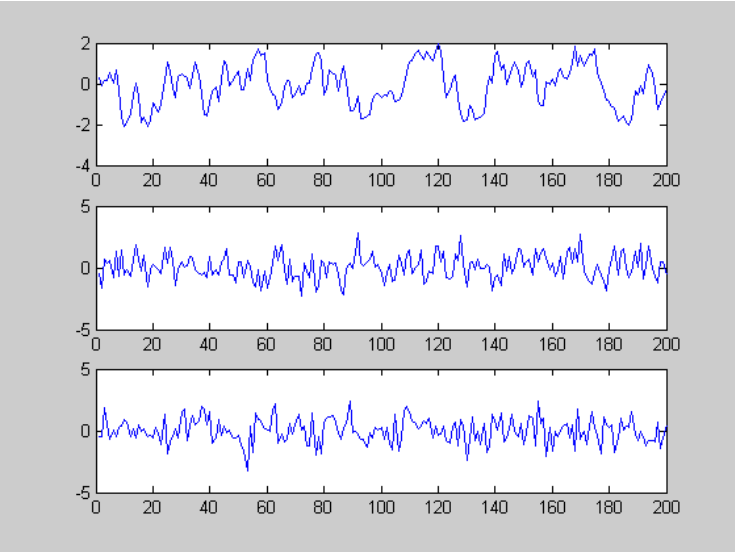


Figure 11: Recovered signals

– Components were not estimated properly due to more than one Gaussian component

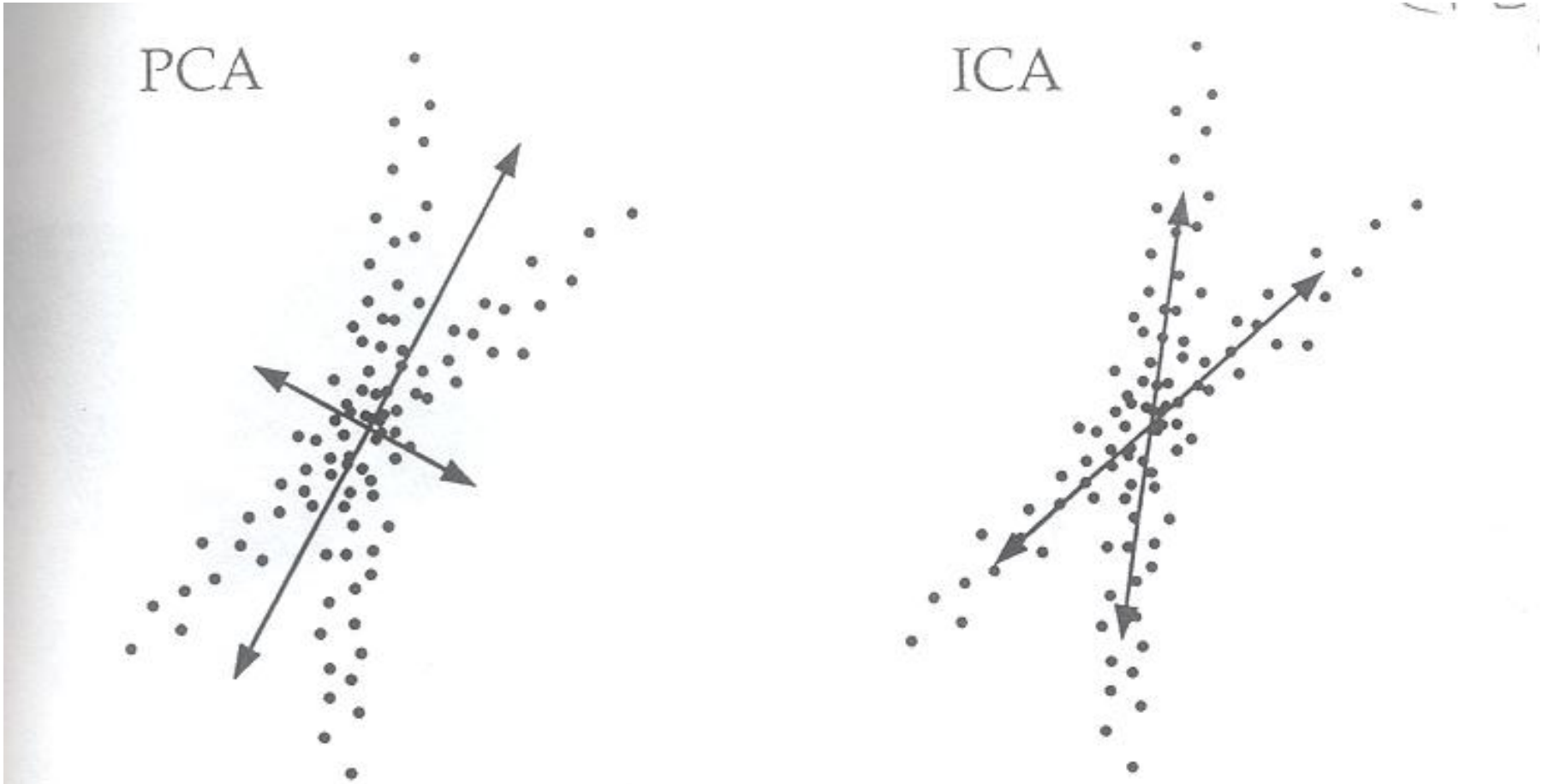
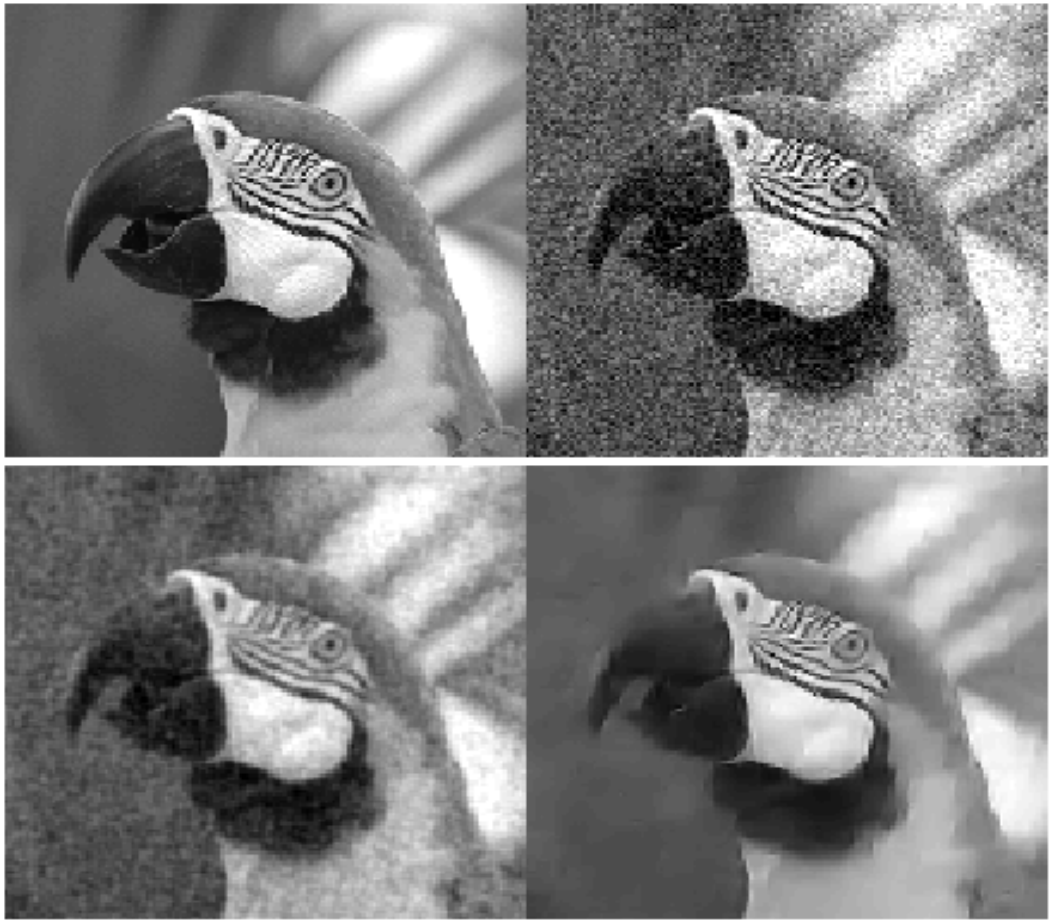


Figure 2.3. Example 2-D data distribution and the corresponding principal component and independent component axes. The data points could be, for example, grayvalues at pixel 1 and pixel 2. Figure inspired by Lewicki & Sejnowski (2000).

Image denoising

Original image



Noisy image

Wiener filtering

ICA filtering

Dimensionality Reduction (Other Methods)

- Kernel PCA
- Locally Linear Embedding (LLE)
- Laplacian Eigenmaps (LEM)
- Multidimensional Scaling (MDS)
- Isomap
- Semidefinite Embedding (SDE)
- Unified Framework
- …

Expectation-Maximization (EM)

- We saw in Ch **2** how we could classify a test point even when it has missing features. We can now extend our application of maximum likelihood techniques to permit the *learning* of parameters governing a distribution from training points, some of which have missing features.
- If we had uncorrupted data, we could use maximum likelihood, i.e., find $\hat{\boldsymbol{\theta}}$ that maximized the log-likelihood $l(\boldsymbol{\theta})$.
- We can also extend maximum likelihood techniques to allow learning of parameters when some training patterns have missing features.

- The basic idea in the expectation maximization or EM algorithm, is to iteratively estimate the likelihood given the data that is present.
- There are two main applications of the EM algorithm:
  - Learning when the data is incomplete or has missing values.
  - Optimizing a likelihood function that is analytically intractable but can be simplified by assuming the existence of values for additional but missing (or hidden) parameters.
- The second problem is more common in pattern recognition applications.

- Some creativity is required to recognize where the EM algorithm can be used!
- Typically used for estimating the parameters of Mixtures of Gaussians.

- The EM algorithm is ideal (i.e., it produces ML estimates) for problems with unobserved (missing) data.

$$\text{Actual data: } \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \text{Observed data: } \mathbf{y} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\text{Complete pdf: } p(\mathbf{x}/\theta), \quad \text{Incomplete pdf: } p(\mathbf{y}/\theta)$$

If the complete *pdf* was available $p(\mathbf{y} \mid \theta) = \int \ldots \int p(\mathbf{x} \mid \theta) d\mathbf{x}_{\text{missing}}$

Consider a full sample $D = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ of points taken from a single distribution. Suppose, though, that here some features are missing; thus any sample point can be written as $\mathbf{x}_k = \{\mathbf{x}_{kg}, \mathbf{x}_{kb}\}$, i.e., comprising the "good" features and the missing, or "bad" ones. $D = D_g \cup D_b$

Next we form the function

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^i) = E_{D_b}\left[\ln p(\mathbf{x}_g, \mathbf{x}_b; \boldsymbol{\theta}) \mid D_g; \boldsymbol{\theta}^i\right]$$

where the use of the semicolon denotes, that $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^i)$ is a function of $\boldsymbol{\theta}$ with $\boldsymbol{\theta}^i$ assumed fixed.

The parameter vector $\boldsymbol{\theta}^i$ is the current (best) estimate for the full distribution; $\boldsymbol{\theta}$ is a candidate vector for an improved estimate.

Given such a candidate $\boldsymbol{\theta}$, the right hand side of Eq. calculates the likelihood of the data, including the unknown feature D_b *marginalized* with respect to the current best distribution, which is described by $\boldsymbol{\theta}^i$.

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^i) = E_{D_b}\left[\ln p(\mathbf{x}_g,\mathbf{x}_b;\boldsymbol{\theta}) \mid D_g;\boldsymbol{\theta}^i\right]$$

$$= \int \ln p(\mathbf{x}_g,\mathbf{x}_b;\boldsymbol{\theta})\, p(\mathbf{x}_b \mid \boldsymbol{\theta}^i) d\mathbf{x}_b$$

Different candidate $\boldsymbol{\theta}_s$ will of course lead to different such likelihoods. Our algorithm will select the best such candidate $\boldsymbol{\theta}$ and call it $\boldsymbol{\theta}^{i+1}$ — the one corresponding to the greatest $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^i)$.

The EM estimate is only guaranteed to never get worse. Usually, it will find a peak in the likelihood $p(\mathbf{y} | \boldsymbol{\theta})$, but if the likelihood function $p(\mathbf{y} | \boldsymbol{\theta})$ has multiple peaks, EM will not necessarily find the global maximum of the likelihood. In practice, it is common to start EM from multiple random initial guesses, and choose the one with the largest likelihood as the final guess for $\boldsymbol{\theta}$.

Algorithm 1 (Expectation-Maximization)

1 **begin initialize** $\boldsymbol{\theta}^0, T, i = 0$
2 **do** $i \leftarrow i + 1$
3 **E step** : compute $Q(\boldsymbol{\theta};\ \boldsymbol{\theta}^i)$
5 **M step** : $\boldsymbol{\theta}^{i+1} \leftarrow \arg\max_{\theta} Q(\boldsymbol{\theta};\ \boldsymbol{\theta}^i)$
6 **until** $Q(\boldsymbol{\theta}^{i+1};\ \boldsymbol{\theta}^i) - Q(\boldsymbol{\theta}^i;\ \boldsymbol{\theta}^{i-1}) \leq T$
7 **return** $\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^{i+1}$
8 **end**

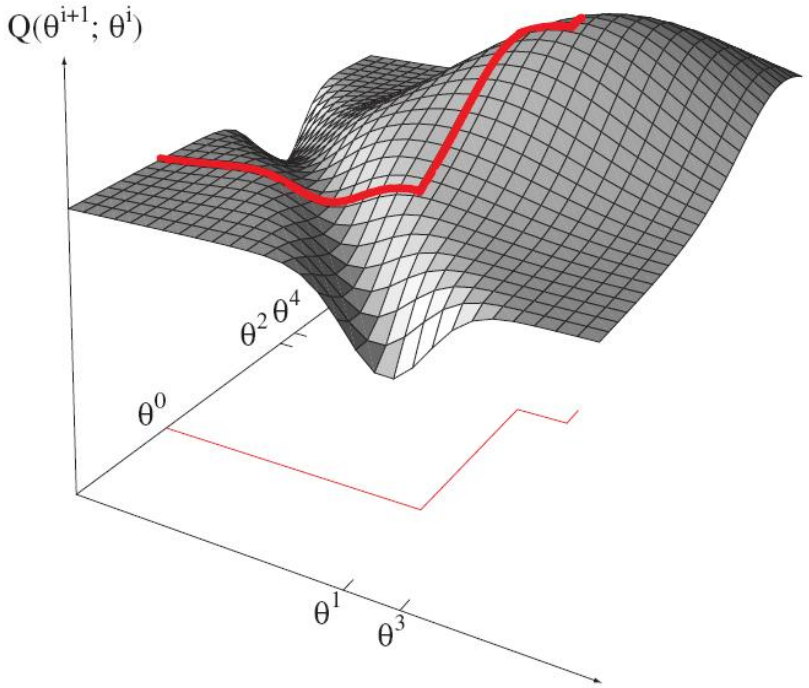


Figure 3.5:

Figure 3.5: The search for the best model via the EM algorithm starts with some initial value of the model parameters, θ^0. Then, via the **M step** the optimal θ^1 is found. Next, θ^1 is held constant and the value θ^2 found which optimizes $Q(\cdot, \cdot)$. This process iterates until no value of θ can be found that will increase $Q(\cdot, \cdot)$. Note in particular that this is different from a gradient search. For example here θ^1 is the global optimum (given fixed θ^0), and would not necessarily have been found via gradient search.

This so-called Expectation-Maximization or EM algorithm is most useful when the optimization of $Q(\cdot, \cdot)$ is simpler than that of $l(\cdot)$.

Most importantly, the algorithm guarantees that the log-likelihood of the good data (with the bad data marginalized) will increase monotonically.

Example 2: Expectation-Maximization for a 2D normal model

$$D = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\} = \left\{\begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}*\\4\end{bmatrix}\right\}$$

where * represents the unknown value of the first feature of point $\mathbf{x}_4$. D_b consists of the single feature x_{41}, and the good data D_g all the rest.

We assume our model is a Gaussian with diagonal covariance and arbitrary mean, and thus can be described by the parameter vector

$$\boldsymbol{\theta} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \sigma_1^2 \\ \sigma_2^2 \end{pmatrix}.$$

We take our initial guess to be a Gaussian centered on the origin having $\boldsymbol{\Sigma} = \mathbf{I}$, that is:

$$\boldsymbol{\theta}^0 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

In finding our first improved estimate, $\boldsymbol{\theta}^1$, we must calculate $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^0)$

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}') = E_{x_{41}}\left[\ln p(\mathbf{x}_g, \mathbf{x}_b;\boldsymbol{\theta}) \,|\, D_g;\boldsymbol{\theta}^0\right]$$

$$= \int_{-\infty}^{\infty}\left[\sum_{k=1}^{3}\ln\left(p(\mathbf{x}_k\,|\boldsymbol{\theta}\right) + \ln\left(p(\mathbf{x}_4\,|\,\boldsymbol{\theta}\right)\right]p(x_{41}\,|\boldsymbol{\theta}^0; x_{42}=4)dx_{41}$$

$$= \left(\sum_{k=1}^{3}\ln\left(p(\mathbf{x}_k\,|\boldsymbol{\theta}\right)\right) + \int_{-\infty}^{\infty}\left[\ln p\left(\begin{pmatrix}x_{41}\\4\end{pmatrix}\middle|\boldsymbol{\theta}\right)\frac{p\left(\begin{pmatrix}x_{41}\\4\end{pmatrix}\middle|\boldsymbol{\theta}^0\right)}{\underbrace{\int_{-\infty}^{\infty}p\left(\begin{pmatrix}x'_{41}\\4\end{pmatrix}\middle|\boldsymbol{\theta}^0\right)dx'_{41}}_{K}}\right]dx_{41}$$

where x_{41} is the unknown first feature of point $\mathbf{x}_4$, and K is a constant that can be brought out of the integral.

$$
\begin{aligned}
Q(\boldsymbol{\theta};\, \boldsymbol{\theta}^0) &= \sum_{k=1}^{3}[\ln\, p(\mathbf{x}_k|\boldsymbol{\theta})] + \frac{1}{K}\int_{-\infty}^{\infty}\ln\, p\left(\begin{pmatrix} x_{41} \\ 4 \end{pmatrix}\middle|\boldsymbol{\theta}\right)\frac{1}{2\pi\left|\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right|}\exp\left[-\frac{1}{2}(x_{41}^2 + 4^2)\right] dx_{41} \\
&= \sum_{k=1}^{3}[\ln\, p(\mathbf{x}_k|\boldsymbol{\theta})] - \frac{1+\mu_1^2}{2\sigma_1^2} - \frac{(4-\mu_2)^2}{2\sigma_2^2} - \ln\,(2\pi\sigma_1\sigma_2).
\end{aligned}
$$

This completes the expectation or **E step**. Through a straightforward calculation, we find the values of $\boldsymbol{\theta}$ (that is, μ_1, μ_2, σ_1 and σ_2 that maximize $Q(\cdot\,;\,\cdot)$, to get the next estimate:

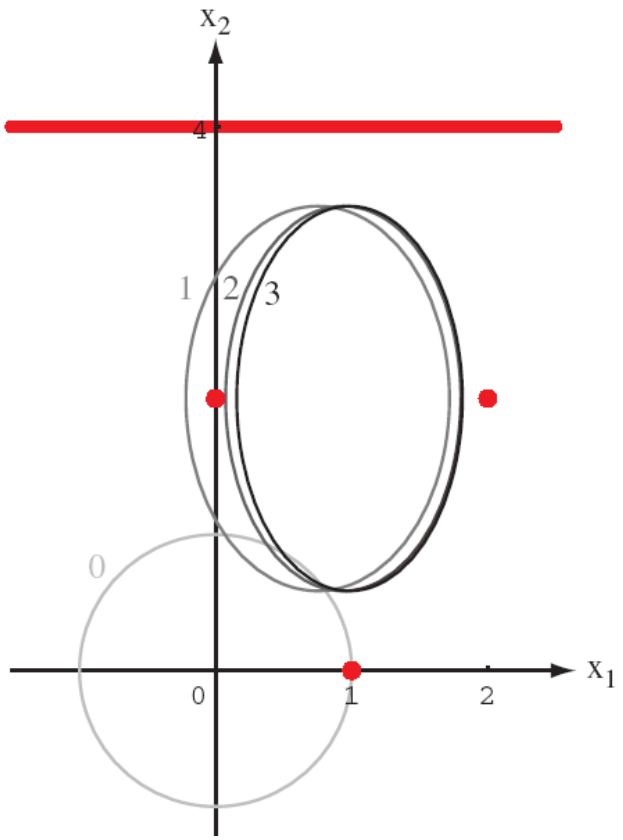
$$
\boldsymbol{\theta}^1 = \begin{pmatrix} 0.75 \\ 2.0 \\ 0.938 \\ 2.0 \end{pmatrix}.
$$

This new mean and the 1/*e* ellipse of the new covariance matrix are shown in the figure. Subsequent iterations are conceptually the same, but require a bit more extensive calculation

The mean will remain at $\mu_2 = 2$. After three iterations the algorithm converges at the solution

$$\boldsymbol{\mu} = \begin{pmatrix} 1.0 \\ 2.0 \end{pmatrix}, \text{ and } \boldsymbol{\Sigma} = \begin{pmatrix} 0.667 & 0 \\ 0 & 2.0 \end{pmatrix}.$$

We must be careful and note that the EM algorithm leads to the greatest loglikelihood of the *good* data, with the bad data marginalized. There may be particular values of the bad data that give a different solution and an even greater log-likelihood.



For instance, in this Example if the missing feature had value $x_{41} = 1$, so that $\mathbf{x}_4 = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$

we would have a solution

$$\boldsymbol{\theta} = \begin{pmatrix} 1.0 \\ 2.0 \\ 0.5 \\ 2.0 \end{pmatrix}$$

and a log-likelihood for the *full* data (good plus bad) that is greater than for the good alone.

Note too that if no data is missing, the calculation of $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^i)$ is simple since no integrals are involved.