Advanced course in machine learning
582744
Lecture 5

Arto Klami

Outline

Unsupervised learning

No distinction between inputs and outputs; we have some data $\mathbf{x}_n$ and want to understand the process generating them

Often solved with latent-variable models: Some unknown quantity for each data point becomes the new representation for that data

- Clustering: Data points represented by cluster index, similar points grouped together
- Dimensionality reduction: Data points represented by lower-dimensional vectors (linear or non-linear mapping)
- Missing value imputation: Estimate data elements that are unknown or censored

Expectation maximization for mixture models

1. *Observed data log-likelihood*:
   $l(\theta) = \log p(x_n|\theta) = \log \sum_k p(x_n, z_n = k|\theta)$
2. *Complete data log-likelihood*: $l_c(\theta) = \log p(x_n, z_n|\theta)$
3. *Expected complete data log-likelihood*: $Q(\theta, \hat{\theta}) = \mathbb{E}_z[l_c(\theta)|\hat{\theta}]$

For one data point the complete data log-likelihood is

$$p(\mathbf{x}_n, z_n) = \prod_k [p(\mathbf{x}_n, z_n = k)]^{\mathbb{I}(z_n=k)},$$

which means the expected one becomes

$$\begin{aligned} Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) &= \sum_n \sum_k \mathbb{E}[\mathbb{I}(z_n = k)] \log(\pi_k p_k(\mathbf{x}_n|z_n, \boldsymbol{\theta}_k)) \\ &= \sum_n \sum_k p(z_n = k|\mathbf{x}_n, \hat{\boldsymbol{\theta}}) \log(\pi_k p_k(\mathbf{x}_n|z_n, \boldsymbol{\theta}_k)), \end{aligned}$$

which is still a function of $\boldsymbol{\theta}$

Expectation maximization for mixture models

The EM algorithm:

1. Compute the expectated values for the latent variables (here z_n)
2. Maximize the expected complete data log-likelihood over the parameters of the model (here $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$)

Principal component analysis

The classical tool for dimensionality reduction, familiar already because of the first exercise (and the Intro course)

Classical (equivalent) definitions: Find orthonormal linear projection of k components that

- Minimizes the reconstruction error
- Maximizes the variance of the projections

The solution is found by computing the eigen-value decomposition of $\mathbf{X}\mathbf{X}^T$ for centered data, retaining the eigenvectors corresponding to the k largest eigenvalues as the lower-dimensional representation

...or equivalently as the right singular vectors of the SVD of $\mathbf{X}$

PCA: Reconstruction

The projections are given by $\mathbf{z}_n = \mathbf{W}^T \mathbf{x}_n$

...and the reconstructions by $\hat{\mathbf{x}}_n = \mathbf{W}\mathbf{z}_n = \mathbf{W}\mathbf{W}^T x_n$

Intuition:
$z_n = \mathbf{w}^T \mathbf{x}_n$ tells how far the point is along $\mathbf{w}$
$z_n \mathbf{w}$ is the actual representation of the point in that subspace

PCA: Inference

$L(\mathbf{w}) = \max \text{var}(\mathbf{w}^T \mathbf{x}) = \mathbf{w}^T \mathbf{\Sigma} \mathbf{w}$
subject to $\mathbf{w}^T \mathbf{w} = 1$

How can we solve such constrained optimization problems?

Lagrange multipliers

- Given a loss $L(\boldsymbol{\theta})$ and a constraint $f(\boldsymbol{\theta}) = C$, we can use *Lagrange functions* to recognize optima
- Intuition:
  - Along the curve of $f(\boldsymbol{\theta}) = C$ we still need to find the optimum of the loss, which is recognized as the gradient being perpendicular to the constraint: $\nabla L(\boldsymbol{\theta}) = \lambda \nabla f(\boldsymbol{\theta})$
  - If it were not, we could take a small step along the gradient projected on to the constraint set
- Augment the loss as $L' = L(\boldsymbol{\theta}) - \lambda(f(\boldsymbol{\theta}) - C)$ and find $L' = 0$
- ...which is simply another way of writing $\nabla L(\boldsymbol{\theta}) = \lambda \nabla f(\boldsymbol{\theta})$

PCA: Inference

$L(\mathbf{w}) = \max \text{var}(\mathbf{w}^T\mathbf{x}) = \mathbf{w}^T\mathbf{\Sigma}\mathbf{w}$ subject to $\mathbf{w}^T\mathbf{w} = 1$

Using Lagrange multipliers for the constraint we get $(\mathbf{w}^T\mathbf{\Sigma}\mathbf{w}) - \lambda(\mathbf{w}^T\mathbf{w} - 1)$, which has the gradient $\nabla L = 2\mathbf{\Sigma}\mathbf{w} - 2\lambda\mathbf{w}$

The gradient is zero for any pair $(\mathbf{w}, \lambda)$ that are some eigenvector and -value of $\mathbf{\Sigma}$, and the maximum is obtained with the largest one

...and the variance retained is simply $\mathbf{w}_1^T\mathbf{\Sigma}\mathbf{w}_1 = \lambda_1$

How about the next direction? Add the constraints $\mathbf{w}_2^T\mathbf{w}_1 = 0$ and $\mathbf{w}_2^T\mathbf{w}_2 = 1$ and go through the same derivation

PCA: Whitening

PCA can be used to perform a data processing step called *whitening*

- Solve PCA and retain all components
- Normalize the components with the standard derivation $\sqrt{\lambda}$

The resulting data has unit variance in all directions and the dimensions are uncorrelated

Principal component analysis

It is also possible to write a probabilistic formulation for PCA as a generative latent variable model, and infer the parameters using EM algorithm

Why bother?

- Missing data
- EM actually faster for some cases than direct SVD, and here actually is guaranteed to converge to the global optimum
- Extensions as part of a hierarchical model, or by changing the likelihood
- The probabilistic evaluation tools and model selection

Continuous latent variables

Mixture models were based on discrete latent variables indicating a cluster membership

The same basic tools are applicable also for continuous latent variables

Multivariate normal distribution

Remember the first exercises:
If $\mathbf{x} \sim N(0, \mathbf{I})$ then $\mathbf{y} = \mathbf{W}\mathbf{x} \sim N(0, \mathbf{W}\mathbf{W}^T)$

We can think of this as low-rank parameterization for a covariance matrix: $\mathbf{\Sigma} = \mathbf{W}\mathbf{W}^T$

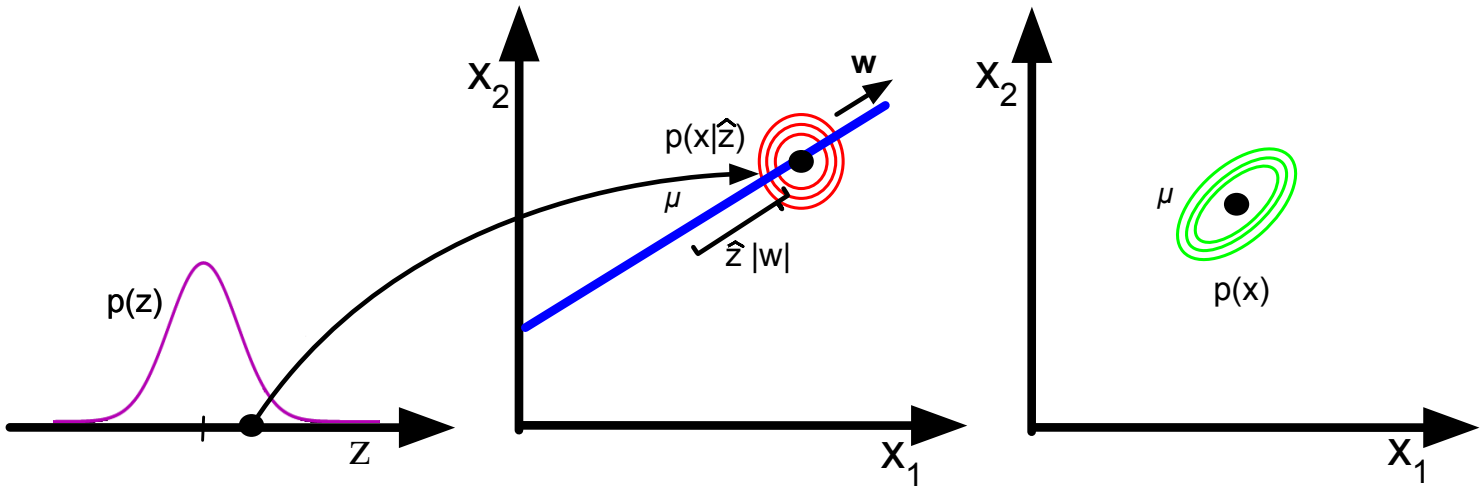
More generally

$$\mathbf{x} \sim N(\boldsymbol{\mu}, \mathbf{\Sigma}) \rightarrow \mathbf{y} \sim N(\mathbf{W}\boldsymbol{\mu}, \mathbf{W}\mathbf{\Sigma}\mathbf{W}^T)$$

and

$$\mathbf{z} = \mathbf{y} + N(\mathbf{a}, \mathbf{S}) \sim N(\mathbf{W}\boldsymbol{\mu} + \mathbf{a}, \mathbf{W}\mathbf{\Sigma}\mathbf{W}^T + \mathbf{S})$$

Probabilistic interpretation for PCA



Factor analysis

Factor analysis is defined as

$$\mathbf{z}_n \sim N(0, \mathbf{I})$$
$$\mathbf{x}_n \sim N(\mathbf{W}\mathbf{z}_n, \mathbf{\Psi})$$

where $\mathbf{\Psi}$ is diagonal

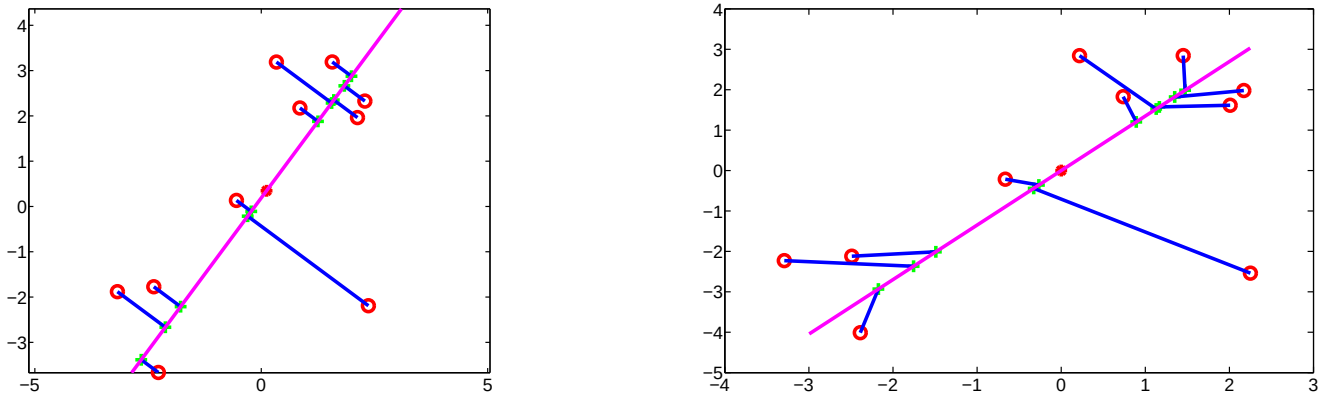
Used for learning interpretable factors of the data

Unidentifiability: The maximum likelihood solution is identical for arbitrary rotations of $\mathbf{W}$, but can be fixed by

- Requiring orthonormality
- Making $\mathbf{W}$ lower-triangular
- Various sparsity assumptions

Probabilistic PCA is a special case with $\mathbf{\Psi} = \sigma^2\mathbf{I}$, and classical PCA is obtained when $\sigma^2 \to 0$

Regularizing effect of PPCA



Projections of probabilistic PCA (right) pulled towards zero because of the prior $\mathbf{z}_n \sim N(0, \mathbf{I})$

EM for factor analysis

For inference we can use the same EM algorithm presented last lecture

The expected data log-likelihood is

$$I(\theta, \theta^t) = \mathbb{E}_{p(\mathbf{z}_n|\mathbf{x}_n, \hat{\mathbf{W}}, \hat{\Psi})}[\sum_n \log p(\mathbf{x}_n, \mathbf{z}_n|\mathbf{W}, \Psi)],$$

where the conditional posterior of $\mathbf{z}_n$ given the previous values of $\hat{\Psi}$ and $\hat{\mathbf{W}}$ is $N(\mathbf{m}, \mathbf{S})$ with

$$\mathbf{S}_n = (\mathbf{I} + \mathbf{W}^T \Psi^{-1} \mathbf{W})^{-1}$$
$$\mathbf{m}_n = \mathbf{S}_n(\mathbf{W}^T \Psi^{-1} \mathbf{x}_n)$$

EM for factor analysis

The epectations of that normal distribution are

$$\mathbb{E}[\mathbf{z}_n] = \mathbf{m}_n$$
$$\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] = \mathbf{m}_n \mathbf{m}_n^T + \mathbf{S}_n$$

...and after quite some algebraic manipulation (Section 4, Section 12.1.5) we get the following updates for the actual parameters:

- $\mathbf{W}^{t+1} = \mathbb{E}[\mathbf{x}_n \mathbf{z}_n^T] \mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T]^{-1}$
- $\mathbf{\Psi} = \frac{1}{N} \text{diag}(\sum_n (\mathbf{x}_n - \mathbf{W}^{t+1} \mathbf{m}) \mathbf{x}_n^T)$

Principal component analysis: Missing data

Standard PCA algorithms fail is some elements of $\mathbf{X}$ are unknown

The probabilistic formulation allows computing the likelihood only over the observed entries

The updates on the previous slide still apply, but now we only sum over the observed values: The latent variables are pulled towards zero and the variance increases

A more justified approach would be to interpret the missing values as latent variables, postulate a probability density $q(x_{nd})$ over them and use EM algorithm for inferring it

The same idea works for mixture models too. See Section 11.6.

Principal component analysis: Number of factors

The number of factors:
Classical solution is to inspect the eigenspectrum and keep components that capture, for example, 90% or 95% of the variance

Probabilistic formulation:

- Cross-validated likelihood
- Automatic-relevance determination: Suitable prior on $\mathbf{W}$ automatically prunes out excess components

Canonical correlation analysis

A related dimensionality reduction method is called CCA:
Given two random variables X and Y, find linear projections $\mathbf{u}$ and $\mathbf{v}$ that maximize the correlation

$$\max \operatorname{cor}(\mathbf{u}^T\mathbf{x}, \mathbf{v}^T\mathbf{y}) = \frac{\mathbf{u}^T\boldsymbol{\Sigma}_{XY}\mathbf{v}}{\sqrt{\mathbf{u}^T\boldsymbol{\Sigma}_X\mathbf{u}}\sqrt{\mathbf{v}^T\boldsymbol{\Sigma}_Y\mathbf{v}}}$$

such that the projections for different components are uncorrelated

The solution obtained via generalized eigenvalue problem, or by whitening of each variable followed by PCA for the concatenation of the whitened variables

Canonical correlation analysis

Probabilistic formulation:

$$\mathbf{z}_n \sim N(0, I)$$
$$\mathbf{x}_n \sim N(\mathbf{W}_x^T \mathbf{z}_n, \mathbf{\Sigma}_x)$$
$$\mathbf{y}_n \sim N(\mathbf{W}_y^T \mathbf{z}_n, \mathbf{\Sigma}_y)$$

...which is equivalent to "factor analysis" of $[\mathbf{x}_n; \mathbf{y}_n]$ with block-diagonal covariance matrix

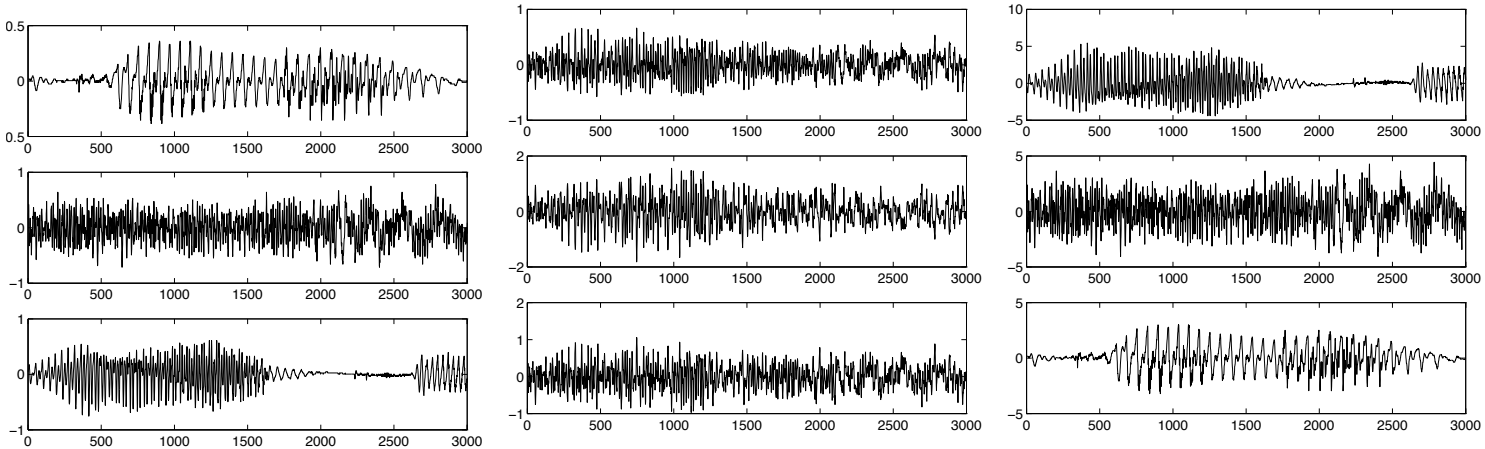
We can further represent the covariance matrices $\mathbf{\Sigma}_x$ with yet another linear model; the resulting model is called *inter-battery factor analysis*

Gaussians are gaussians

The PCA/FA family of models is computationally very simple since linear transformations of normal distributions are still normal distributions

This also means PCA does not reveal very interesting properties of the data

The coctail party problem



A collection of signals $\mathbf{s}$ are mixed with a linear mixing matrix, and we observe only the mixed signals $\mathbf{x} = \mathbf{As}$

Independent component analysis

Intuitive idea: The original signals are statistically independent because they were generated by independent processes

To solve the problem we hence need to learn $\mathbf{s}$ such that the dimensions are independent

Remember the generative model for PCA: $p(\mathbf{z}_n) \sim N(0, \mathbf{I})$
It assumes the latent sources are independent, but also that they are normal distributions

Independent component analysis

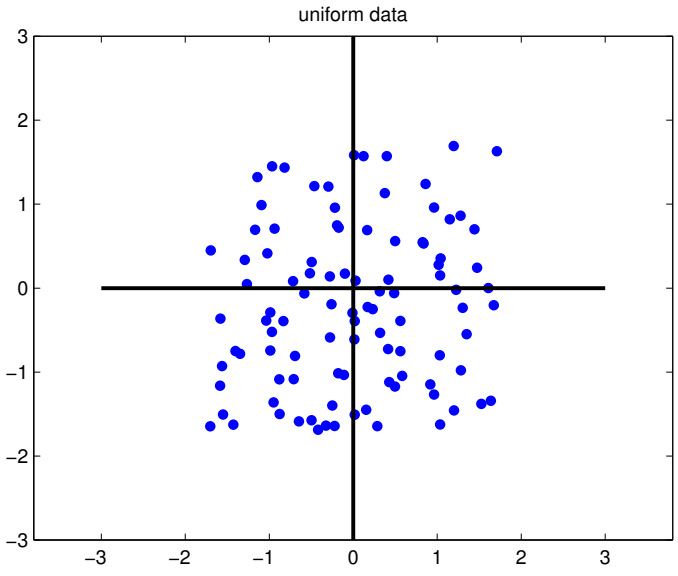
We still need to assume the independence, but is normal distribution a good assumption for the latent source?

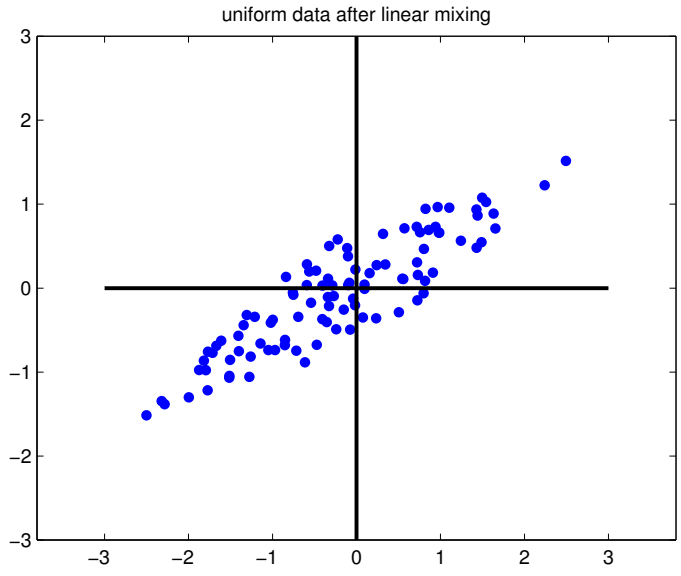
In fact, it is the worst possible one: For normal distribution correlation and mutual information are equivalent and hence we cannot do anything more than find uncorrelated dimensions

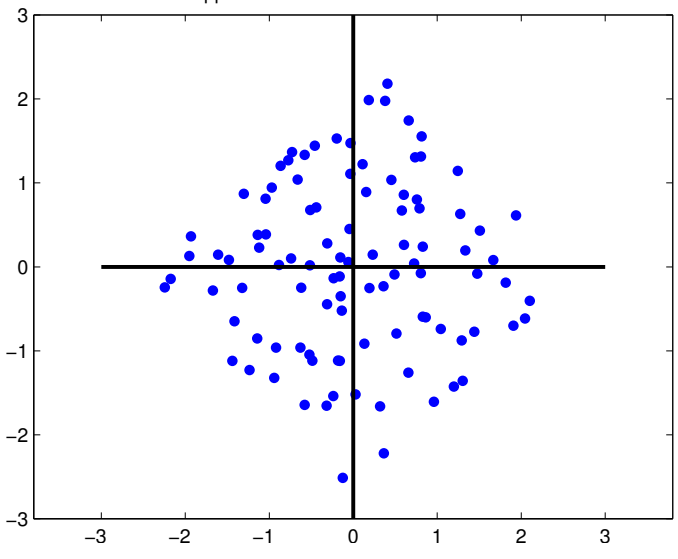
Central limit theorem: Sums of independent random variables approach a normal distribution

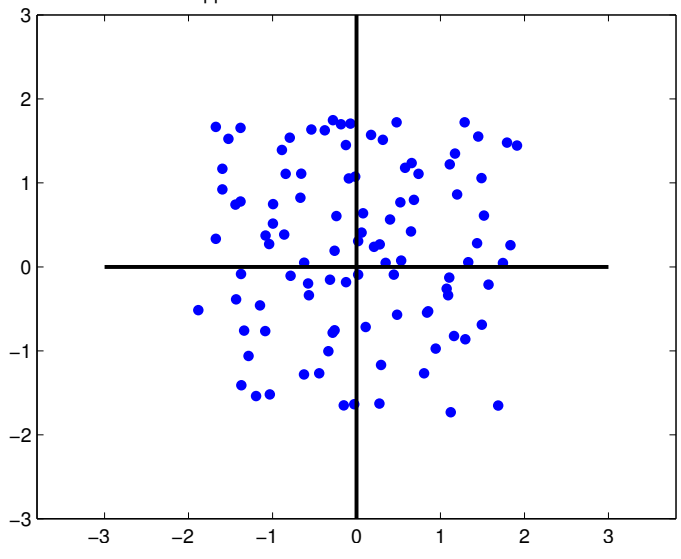
Hence, the original sources can be recovered by assuming they are not normal distributions

Whitening is not enough









Maximizing nongaussianity

Normal distribution has zero higher moments

Kurtosis $\text{kurt}(y) = \mathbb{E}(y^4) - 3(\mathbb{E}(y^2))^2$ is zero for normal distribution, so maximizing its absolute value makes the distribution non-Gaussian

Gradient of the absolute value for $y = \mathbf{w}^T\mathbf{x}$ given by

$$\frac{\partial|\text{kurt}(\mathbf{w}^T\mathbf{x})|}{\partial\mathbf{w}} = 4\text{sign}(\text{kurt}(\mathbf{w}^T\mathbf{x}))[\mathbb{E}(\mathbf{x}(\mathbf{w}^T\mathbf{x})^3) - 3\mathbf{w}\|\mathbf{w}\|^2]$$

Positive for super-Gaussian distributions (spikes, heavy tails) and negative for sub-Gaussian (compact)

FastICA

FastICA is approximative Newton's method for maximizing the absolute value of kurtosis, implemented with a simple fixed-point rule

$$\mathbf{w} \leftarrow \mathbb{E}(\mathbf{x}(\mathbf{w}^T\mathbf{x})^3 - 3\mathbf{w})$$

followed by normalizing $\mathbf{w}$ to the unit-sphere

Further components by deflation:
Replace the normalization stage with $\mathbf{w}_p \leftarrow \mathbf{w}_p - \sum_{j=1}^{p-1}(\mathbf{w}_p^T\mathbf{w}_j)\mathbf{w}_j$

ICA

The whole algorithm:

1. Whiten the data using PCA; this makes the mixing matrix orthogonal
2. Run FastICA on the whitened data to find one ICA component
3. While solving for further components, remember to make them orthogonal also with respect to the earlier ones

ICA can be interepreted as one solution to the rotation ambiquity of factor analysis

Independent component analysis

Generative formulation straightforward to write, but one needs to assume some specific family of non-Gaussian signals

- Super-Gaussian: Laplace distribution
  $\log p(z) = -\sqrt{(2)}|z| - \log(\sqrt{2})$ or the logistic distribution
  $\log p(z) = -2 \log \cosh \frac{\pi}{2\sqrt{2}} z - \log \frac{4\sqrt{3}}{\pi}$
- Sub-Gaussian: Uniform distribution
- Skewed: Normal distribution is symmetric, so any non-symmetric distribtion is non-Gaussian; Gamma distribution is one example
- Mixture of univariate Gaussians

In practice: Just pick one of these and try it out; even if the likelihood is incorrect we will typically find some independent sources

ICA properties

- We cannot identify the scales of the components: Multiplying $\mathbf{s}$ with a and $\mathbf{A}$ with $\frac{1}{a}$ does not change kurtosis or other measures of non-Gaussianity
- There is no natural ordering for the components
- Gaussian components cannot be separated