Probabilistic Principal Component Analysis and Independent Component Analysis

Presented by Eric Heim

Outline

- Review of PCA
- Probable Principal Component Analysis (pPCA)
  - Need and advantages of pPCA compared to PCA
  - Relation to Factor Analysis
  - Definition of pPCA
  - pPCA and Dimensionality Reduction
  - An EM algorithm for pPCA
- Independent Component Analysis (ICA)
  - Definitions of ICA
  - Applications of ICA
  - Multi-Unit Objective (Contrast) Functions
  - One-Unit Objective (Contrast) Functions
  - Algorithms for ICA

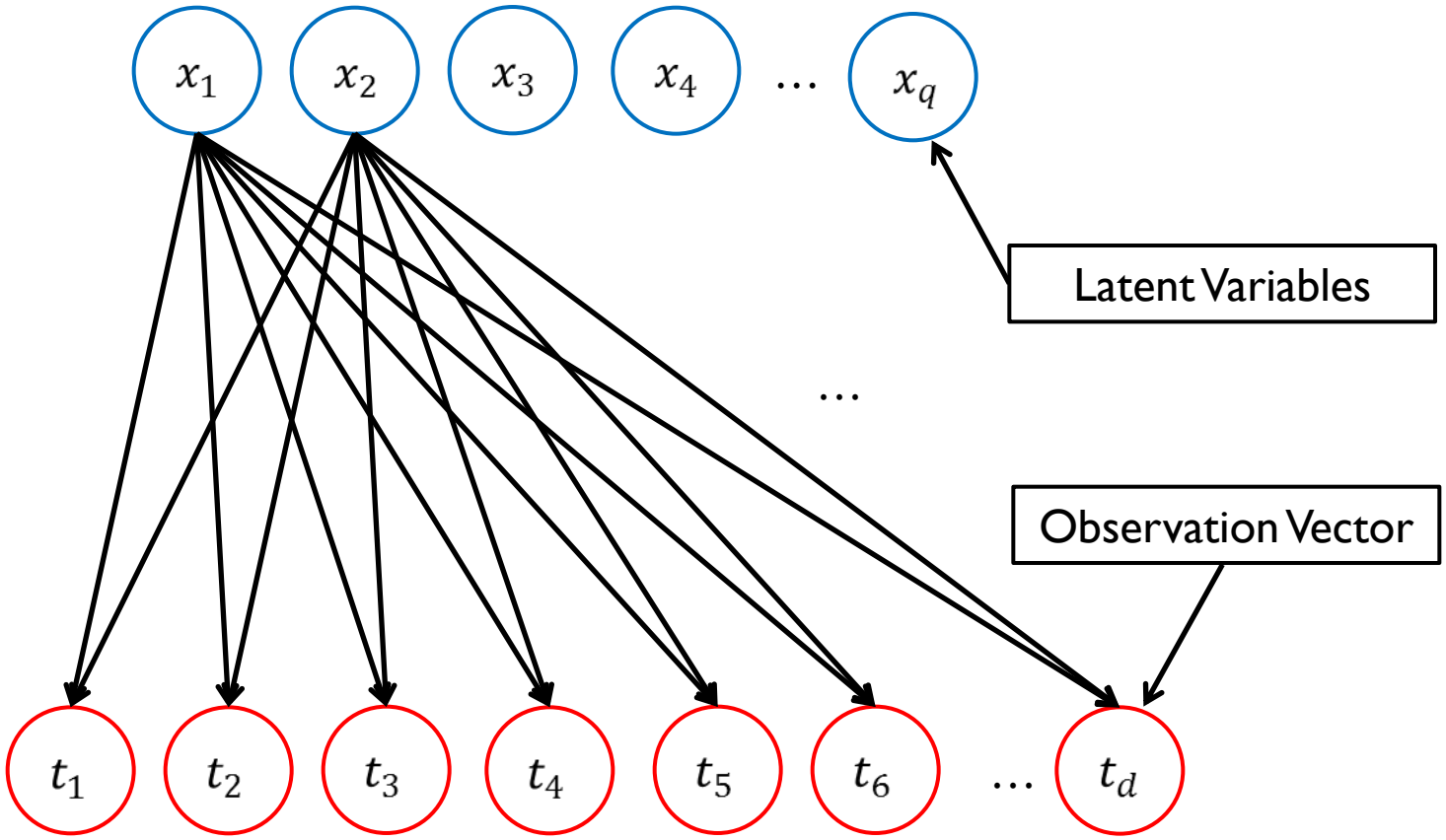
Principal Component Analysis

- Used to transform observed data matrix $\mathbf{X}$ (N x d) into $\mathbf{Y}$ (N x q) (find the q principal components)
  - Fairly simple solution:
    1. Centralize the $\mathbf{X}$
    2. Calculate the covariance matrix $\mathbf{C}$ of $\mathbf{X}$
    3. Calculate the eigenvectors of the $\mathbf{C}$
    4. Select the dimensions that correspond to the q highest eigenvalues
  - Big win for linear algebra.
    - However, because it is a simple linear algebra transformation, PCA does not produce a probabilistic model for the observed data.
      - A probabilistic model can be very useful

Advantages of a probabilistic PCA model

- Enables comparison with other probabilistic techniques
- Facilitates statistical testing
- Permits the application of Bayesian methods
- Extends the scope of PCA
  - Multiple PCA models can be combined as a probabilistic mixture
  - PCA projections can be obtained when some data values are missing
- Can be utilized as a constrained Gaussian density model
  - Classification
  - Novelty detection

Graphical Representation of pPCA



Factor Analysis and PCA

- One way to view PCA probabilistically is to relate it to *latent variable models*.
  - **Goal**: relate a d-dimensional observation vector $\mathbf{t}$ to a corresponding q-dimensional vector of latent variables $\mathbf{x}$
  - One common model *factor analysis* where the relationship is linear:

$$\mathbf{t} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\varepsilon}$$

    - $\mathbf{W}$ is a d x q matrix that relates the variables in t to the latent ones
    - $\boldsymbol{\mu}$ permits non-zero mean
    - $\boldsymbol{\varepsilon}$ is a noise parameter
    - Conventionally, $\mathbf{x} \sim N(\mathbf{0}, \mathbf{I})$
    - If you specify the noise to also be Gaussian $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \boldsymbol{\Psi})$ then $\mathbf{t}$ follows a Gaussian $\mathbf{t} \sim N(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^{\mathrm{T}} + \boldsymbol{\Psi})$
      - Thus, the model's parameters may be found using maximum likelihood

Factor Analysis and PCA

- $$\mathbf{t} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\varepsilon}$$
  $$\mathbf{t} \sim N(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^{\mathrm{T}} + \boldsymbol{\Psi})$$
- If we constrain $\boldsymbol{\Psi}$ to be a diagonal matrix whose elements are usually estimated from the data, the observed variables $\mathbf{t}$ are independent of each other given the latent variables $\mathbf{x}$.
  - $\mathbf{x}$ represents correlations between observation variables
  - ε_i represents variability unique to a particular t_i
    - Differs from PCA in that PCA treats covariance and variance identically!
- Unfortunately, columns of $\mathbf{W}$ will generally not correspond to the principal subspace of the observed data.
  - However, a link can be made if $\boldsymbol{\Psi} = \sigma^2\mathbf{I}$ (isotropic error model)
  - As it turns out, if you estimate $\mathbf{W}$ and σ^2 using maximum likelihood, the isotropic error model corresponds to PCA (probabilistic PCA)

Probabilistic Principal Component Analysis

- Using the isotropic Gaussian noise model $N(\mathbf{0}, \sigma^2\mathbf{I})$ and the original factor analysis model ($\mathbf{t} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\varepsilon}$) we can obtain:

$$\mathbf{t}|\mathbf{x} \sim N(\mathbf{W}\mathbf{x} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$$

- The marginal distribution over the latent variables is conventionally defined by $\mathbf{x} \sim N(\mathbf{0}, \mathbf{I})$, and thus the marginal distribution over $\mathbf{t}$ can be obtained by integrating out the latent variable:

$$\mathbf{t} \sim N(\boldsymbol{\mu}, \mathbf{C}) \text{ where } \mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$$

- The log-likelihood is then defined by:

$$L = -\frac{N}{2}\{d\ \ln(2\pi) + \ln|\mathbf{C}| + \mathrm{tr}(\mathbf{C}^{-1}\mathbf{S})\}, \text{where}$$

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{t}_n - \boldsymbol{\mu})(\mathbf{t}_n - \boldsymbol{\mu})^{\mathrm{T}}$$

Probabilistic Principal Component Analysis

- It can be shown ([1] Appendix A) that the likelihood is maximized when:

$$\mathbf{W}_{\mathrm{ML}} = \mathbf{U}_q(\sqrt[2]{\mathbf{\Lambda}_q - \sigma^2\mathbf{I}})\,\mathbf{R}$$

  - $\mathbf{U}_q$ is a d x q matrix where the q column vectors are the principal eigenvectors of $\mathbf{S}$.
  - $\mathbf{\Lambda}_q$ is a q x q diagonal matrix with corresponding eigenvalues along the diagonal.
  - $\mathbf{R}$ is an arbitrary q x q orthogonal rotation matrix
- For $\mathbf{W} = \mathbf{W}_{\mathrm{ML}}$ the maximum likelihood estimate for σ^2 is:

$$\sigma^2{}_{\mathrm{ML}} = \frac{1}{d-q}\sum_{j=q+1}^{d}\lambda_j$$

- To find the most likely model given $\mathbf{S}$, estimate $\sigma^2{}_{\mathrm{ML}}$ and then $\mathbf{W}_{\mathrm{ML}}$ with $\mathbf{R} = \mathbf{I}$, or you can employ the EM algorithm (discussed later) where $\mathbf{R}$ at convergence can be arbitrary.

Dimensionality Reduction in pPCA

- So, how do we use this to reduce the dimensionality of data?
- Consider the dimensionality reduction process in terms of the distribution of latent variables, conditioned on the observation:

$$\mathbf{x}|\mathbf{t} \sim N\left(\mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{t} - \boldsymbol{\mu}), \sigma^2\mathbf{M}^{-1}\right), \text{where}$$
$$\mathbf{M} = \mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2\mathbf{I}$$

- This can be summarized by its mean:

$$\langle \mathbf{x}_n | \mathbf{t}_n \rangle = \mathbf{M}^{-1}\mathbf{W}_{\mathrm{ML}}{}^{\mathrm{T}}(\mathbf{t}_n - \boldsymbol{\mu})$$

- Intuitively, the optimal reconstruction of $\mathbf{t}_n$ should be $\mathbf{W}_{\mathrm{ML}}\langle \mathbf{x}_n | \mathbf{t}_n \rangle + \boldsymbol{\mu}$. However, it is not. For $\sigma^2 > 0$ it is not an orthogonal projection of $\mathbf{t}_n$.
- If we consider the limit as $\sigma^2 \to 0$, the projection $\mathbf{W}_{\mathrm{ML}}\langle \mathbf{x}_n | \mathbf{t}_n \rangle$ does become orthogonal and is equivalent to conventional PCA, but then the density model is singular and thus undefined.

Dimensionality Reduction in pPCA

- So, what do we do?
- Fortunately, there is no need to take this limit, since the optimal least-squares linear reconstruction of the data from the posterior mean vectors $\langle \mathbf{x}_n | \mathbf{t}_n \rangle$ may be obtained from (see [2] Appendix B for derivation)

$$\hat{\mathbf{t}}_n = \mathbf{W}_{\mathrm{ML}} ({\mathbf{W}_{\mathrm{ML}}}^{\mathrm{T}} \mathbf{W}_{\mathrm{ML}})^{-1} \mathbf{M} \langle \mathbf{x}_n | \mathbf{t}_n \rangle + \boldsymbol{\mu}$$

EM for pPCA

- There are a couple EM algorithms for pPCA:
- [1]
  - E-Step

$$\langle \mathbf{x}_n \rangle = \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{t}_n - \boldsymbol{\mu})$$
$$\langle \mathbf{x}_n \mathbf{x}_n{}^{\mathrm{T}} \rangle = \sigma^2 \mathbf{M}^{-1} \langle \mathbf{x}_n \rangle \langle \mathbf{x}_n \rangle^{\mathrm{T}}, \text{ where}$$
$$\mathbf{M} = \mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2 \mathbf{I}$$

  - M-Step

$$\widetilde{\mathbf{W}} = \left\{ \sum_{n=1}^{N} (\mathbf{t}_n - \boldsymbol{\mu}) \langle \mathbf{x}_n \rangle^{\mathrm{T}} \right\} \left(\sum_{n=1}^{N} \langle \mathbf{x}_n \mathbf{x}_n{}^{\mathrm{T}} \rangle \right)^{\mathrm{T}}$$

$$\tilde{\sigma}^2 = \frac{1}{Nd} \sum_{n=1}^{N} \left\{ ||\mathbf{t}_n - \boldsymbol{\mu}||^2 - 2\langle \mathbf{x}_n \rangle^{\mathrm{T}} \widetilde{\mathbf{W}}^{\mathrm{T}} (\mathbf{t}_n - \boldsymbol{\mu}) + \mathrm{tr}(\langle \mathbf{x}_n \mathbf{x}_n{}^{\mathrm{T}} \rangle \widetilde{\mathbf{W}}^{\mathrm{T}} \widetilde{\mathbf{W}}) \right\}$$

  - Combine the two by substituting $\langle \mathbf{x}_n \rangle$ and $\langle \mathbf{x}_n \mathbf{x}_n{}^{\mathrm{T}} \rangle$ in the M-Step:

$$\widetilde{\mathbf{W}} = \mathbf{S}\mathbf{W}(\sigma^2 \mathbf{I} + \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{S}\mathbf{W})^{-1}$$
$$\tilde{\sigma}^2 = \frac{1}{d} \mathrm{tr}(\mathbf{S} - \mathbf{S}\mathbf{W}\mathbf{M}^{-1}\widetilde{\mathbf{W}}^{\mathrm{T}}), \text{ where}$$
$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{t}_n - \boldsymbol{\mu})(\mathbf{t}_n - \boldsymbol{\mu})^{\mathrm{T}}$$

Switching Gears…

- Now onto ICA

Independent Component Analysis

- So far our linear transform models defined a principal that tells us which transforms are optimal.
- The goal of Independent Component Analysis (ICA) is to find which components are as statistically independent from each other as possible.
  - Define $y_1, y_2, \dots, y_m$ as some random variables with joint density $f(y_1, y_2, \dots, y_m)$
    - Assume zero mean
    - These variables are mutually independent if

$$f(y_1, y_2, \dots, y_m) = f_1(y_1) f_2(y_2) \dots f_3(y_3)$$

Definitions of ICA

1. General Definition of ICA:

ICA of the observed m-dimensional random vector $\mathbf{x}$ consists of finding a linear transform $\mathbf{s} = \mathbf{W}\mathbf{x}$ so that the components s_i are as independent as possible, in the sense of maximizing some function $F(s_1, \dots, s_m)$ that measures independence.

2. Noisy ICA model:

ICA of a random vector $\mathbf{x}$ consists of estimating the following generative model for the data:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n}$$

where the latent variables (components) s_i in vector $\mathbf{s} = (s_1, \dots, s_n)^{\mathrm{T}}$ are assumed independent. The matrix $\mathbf{A}$ is a constant m x n "mixing" matrix, and $\mathbf{n}$ is a m-dimensional random noise vector.

3. Noise-free ICA model:

ICA of a random vector $\mathbf{x}$ consists of estimating the following generative model for the data:

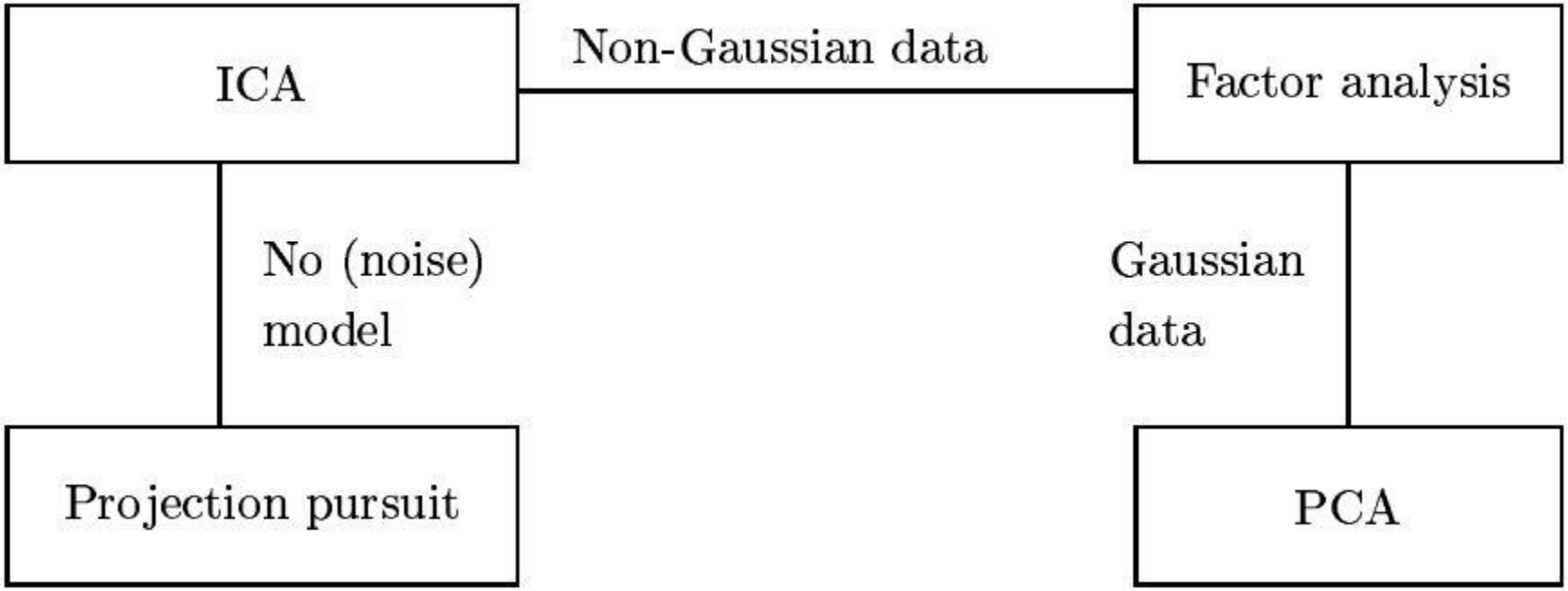
$$\mathbf{x} = \mathbf{A}\mathbf{s}$$

where the matrix $\mathbf{A}$ and $\mathbf{s}$ are the same as Noisy ICA model

Identifiability of an ICA model

- $$\mathbf{x} = \mathbf{A}\mathbf{s}$$
- An ICA model is identifiable if :
  1. All the independent components s_i, with the possible exception of one component, must be non-Gaussian
     - This is necessary because for Gaussian random variables, uncorrelatedness implies independence, and thus, any decorrelating representation would give independence.
  2. $m \geq n$
     - Not completely necessary, but for our purposes it is.
  3. $\mathbf{A}$ is full column rank
     - Some rank restriction is required

Relations to Other Methods



Applications of ICA

- Blind source separation
  - Trying to differentiate the different sources from a single discrete time-stepped signal
- Feature Extraction
  - Shown to be very effective in natural image data
- Blind deconvolution
  - $\mathbf{s}$ and $\mathbf{x}$ are different observations of signals, beginning at different points in time.
- Anything where projection pursuit and factor analysis are used (pPCA!).

Multi-Unit Objective (Contrast) Functions

- $$\mathbf{x} = \mathbf{A}\mathbf{s},\ \mathbf{W} = \mathbf{A}^{-1}$$
- Log-Likelihood

$$L = \sum_{r=1}^{R} \sum_{i=1}^{m} \log f_i\left(\mathbf{w}_i{}^{\mathrm{T}}\mathbf{x}(r)\right) + R\ln|\det \mathbf{W}|$$

  - f_i is the density function of s_i (assumed to be known)
  - $\mathbf{x}(r)$ is the rth realization of $\mathbf{x}$
- Output Entropy (Information Flow)

$$L_2 = H\left(g_1\left(\mathbf{w}_1{}^{\mathrm{T}}\mathbf{x}\right), \ldots, g_m\left(\mathbf{w}_m{}^{\mathrm{T}}\mathbf{x}\right)\right)$$

  - Neural network viewpoint
    - $\mathbf{x}$ viewed as the input to the neural network
    - g_i are some non-linear, scalar function viewed as the outputs of the neural network
    - $\mathbf{w}_i$ are the weight vectors of the neurons

Multi-Unit Objective (Contrast) Functions

- $$\mathbf{x} = \mathbf{A}\mathbf{s},\ \mathbf{W} = \mathbf{A}^{-1}$$
- Mutual Information
  - Standard Definition

$$I(y_1, y_2, \dots, y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y})$$

    - $y_1, y_2, \dots, y_m$ are random variables
    - H is differential entropy
    - Good start because it represents the natural dependence between random variables
    - So, finding a transformation that minimizes the mutual information between components in s is a natural way of estimating the ICA model.
    - Define the transform as $\mathbf{s} = \mathbf{W}\mathbf{x}$

$$I(s_1, s_2, \dots, s_m) = \sum_{i=1}^{m} H(s_i) - H(\mathbf{x}) - \log|\det \mathbf{W}|$$

Multi-Unit Objective (Contrast) Functions

- $$\mathbf{x} = \mathbf{A}\mathbf{s},\ \mathbf{W} = \mathbf{A}^{-1}$$
- Kullback-Liebler divergence
  - One could measure independence by the KL divergence between the real density $f(\mathbf{s})$ and the factored marginal densities $\tilde{f}\ (\mathbf{s}) = f_1(s_1)f_2(s_2)\ldots f_m(s_m)$

$$\delta(f, \tilde{f}) = \int f(\mathbf{s})\log\frac{f(\mathbf{s})}{\tilde{f}(\mathbf{s})}\,d\mathbf{s}$$

  - Again, the problem here is that you have to estimate the densities. Some authors have proposed cumulant-based approximations. See [4] for explanation of how this is done.
- Other multi-unit objection functions
  - Non-linear cross-correlations
  - Non-linear PCA criteria
  - Higher-order cumulant tensors
  - Weighted covariance matrix

One-Unit Objective (Contrast) Functions

- Different than multi-unit in that the function optimizes estimation of a single independent component (one vector $\mathbf{w}$ in $\mathbf{W}$).
- This can be iterated to find several independent components.

One-Unit Objective (Contrast) Functions

- Why should we use one-unit objective functions?
  - They are directly connected to projection pursuit. They can be seen as measures of non-Gaussianity.
  - Most applications do not need to estimate all of the independent components. For example, in projection pursuit the most interesting independent components are found first.
  - Complexity is reduced
  - Prior knowledge of the number of independent components is not needed, as the components can be estimated one-by-one
  - Connected to neural networks, and thus has computationally simple solutions.
  - After estimating one independent component, one can use simple decorrelation to find different independent components (independent components are by definition decorrelated). So, making decorelation a constraint with respect to the independent components already found, one can find the other independent components.

One-Unit Objective (Contrast) Functions

- General Contrast Functions
  - "Statistically appealing properties"
  - Require no prior knowledge of the densities of the independent components.
  - Allow simple algorithmic implementation
  - Simple to analyze
  - View independence as a measure of non-normality

$$J_G = |E_y\{G(y)\} - E_v\{G(v)\}|^p$$

    - G is "practically any function"
    - y is a random variable
    - v is a standardized Gaussian random variable
    - J_G is a measure of non-normality of y
  - Two proposed G functions:

$$G_1(u) = \log\cosh a_1 u, G_2(u) = \exp(-a_2 u^2/2)$$

Algorithms for ICA

- After choosing an objective function, one needs to find a way to optimize it.
- There are a number of algorithms for this.
  - Most of the algorithms require that the data is sphered (if not required, it usually helps)
    - Sphering means the observed variable $\mathbf{x}$ is transformed to the variable $\mathbf{v}$ by

$$\mathbf{v} = \mathbf{Q}\mathbf{x}, \text{ such that}$$
$$E\{\mathbf{v}\mathbf{v}^{\mathrm{T}}\} = \mathbf{I}$$

    - Sphering can be done with simple PCA
      - We will assume the data is sphered for the following algorithms
      - Note: that [4] assumes $m = n$, this is partially justified by the sphering assumption.

Algorithms for ICA

- Jutten-Hérault Algorithm
  - Based on cancelling the non-linear cross-correlations
  - Non-linear cross correlations are of the form

$$E\{g_1(y_i)g_2(y_j))\}$$

    - g_1 and g_2 are some suitably-chosen odd non-linearities
    - If y_i and y_j are independent, the above cross-correlation is zero.
  - The non-diagonal terms of the matrix $\mathbf{W}$ are updated according to:

$$\Delta\mathbf{W}_{ij} \propto g_1(y_i)g_2(y_j), \text{for } i \neq j$$

  - $\mathbf{y}$ is updated every iteration by:

$$\mathbf{y} = (\mathbf{I} + \mathbf{W})^{-1}\mathbf{x}$$

  - The diagonal terms $\mathbf{W}_{ii}$ are set to zero.
  - $\mathbf{y}$ gives estimates of the independent components after convergence.
    - Notice, no explicit objective function.
    - Unfortunately, this only happens under severe restrictions.

Algorithms for ICA

- Maximum likelihood or network entropy (infomax) estimation
  - Class of algorithms usually based on gradient ascent of the objective function.
  - One example:

$$\Delta\mathbf{W} \propto [\mathbf{W}^{\mathrm{T}}]^{-1} - 2\tanh(\mathbf{W}\mathbf{x})\mathbf{x}^{\mathrm{T}}$$

    - tanh is applied separately on every component of the vector $\mathbf{W}\mathbf{x}$
    - tanh is used because it is the derivative of the log-density of the logistic distribution.
    - This converges very slowly, but faster if data is sphered.
      - However, this can be sped up by multiplying the right hand side by $\mathbf{W}\mathbf{W}^{\mathrm{T}}$ (natural gradient method):

$$\Delta\mathbf{W} \propto (\mathbf{I} - 2\tanh(\mathbf{y})\mathbf{y}^{\mathrm{T}})\mathbf{W}, \text{where}$$
$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

    - Later, a Newton method for maximizing was introduced, increasing speedup further.

Algorithms for ICA

- FastICA
  - Gradient methods are often slow.
  - An alternative approach is to use a batch (block) algorithm based on fixed-point iteration.
  - For sphered data, the one-unit FastICA algorithm has the following form:

$$\mathbf{w}(k) = E\{\mathbf{x}g\left(\mathbf{w}(k-1)^{\mathrm{T}}\mathbf{x}\right\} - E\left\{g'\left(\mathbf{w}(k-1)^{\mathrm{T}}\mathbf{x}\right)\right\}\mathbf{w}(k-1)$$

    - The weight vector $\mathbf{w}$ is also normalized to unit norm after every iteration.
    - g is the derivative of the "practically any" G function from the general one-unit contrast function.
    - The expectations are estimated in practice, using sample averages

Conclusion

- pPCA
  - Views principal component analysis probabilistically
  - Has many advantages over simple PCA:
    - Permits the application of Bayesian methods
    - Can combine multiple PCA models
    - Allows for missing data values
    - Facilitates statistical testing
    - Can be utilized as a constrained Gaussian density model
- ICA
  - Transformation of the data into components that are "independent as possible"
  - Applications in:
    - Projection pursuit
    - Factor analysis
    - Blind source separation
    - Feature extraction,
    - Blind deconvolution

Thanks

- Questions?

References

[1] C. Bishop and M. Tipping, "Probabilistic principal component analysis," *J. R. Statist. Soc. B*, vol. 21, no. 3, pp. 611–622, 1999.

[2] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.

[3] Roweis, S,. (1998) EM Algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems*. v.10, p.626, 1998.

[4] A. Hyvarinen *Survey on Independent Component Analysis*.