

Linear Factor Models

Sargur N. Srihari
srihari@cedar.buffalo.edu

Topics in Linear Factor Models

1. Definition of Linear Factor Analysis
2. Related methods
 1. Principal Components Analysis
 2. Factor Analysis
3. Linear Factor Models generalize the above
4. Independent Component Analysis (ICA)
5. Slow Feature Analysis
6. Sparse Coding
7. Manifold Interpretation of PCA

Deep Learning is about Models

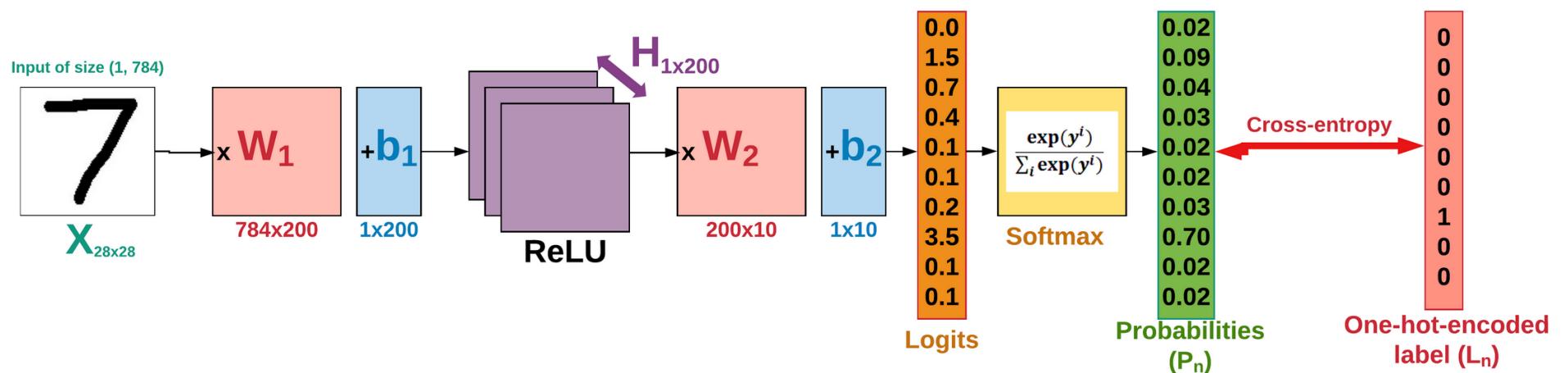
- Many research frontiers of deep learning involves building a probabilistic model of input $p_{\text{model}}(\mathbf{x})$
- Such a model can be used with probabilistic inference to predict any variables given any other variables

Simplest model with latent variables

- Deep Learning is often to construct $p_{\text{model}}(\mathbf{x})$
 - Useful to predict variables given other variables
- With latent variables $p_{\text{model}}(\mathbf{x}) = E_h p_{\text{model}}(\mathbf{x} | \mathbf{h})$
 - Latent variables provide another means of representing the data
 - Representations using latent variables obtain all advantages of feedforward and recurrent networks
- Latent Factor Models are the simplest models with latent variables

Models with Latent Variables

- Many models also have latent variables, h
 - We can write $p_{\text{model}}(x) = E_h p_{\text{model}}(x|h)$ Since $p(x) = \sum_h p(x,h) = \sum_h p(x|h)p(h) = E_h p(x|h)$
 - These latent variables provide another means of representing the data



Models with Latent Variables

- Much of deep learning involves building a probabilistic model of input $p_{\text{model}}(\mathbf{x})$
 - From which we can infer any other variables
- Many models also have latent variables, \mathbf{h}
 - We can write $p_{\text{model}}(\mathbf{x}) = \mathbb{E}_{\mathbf{h}} p_{\text{model}}(\mathbf{x}|\mathbf{h})$
 - These latent variables provide another means of representing the data
- Distributed representations based on latent variables can have all the advantages of representation learning with deep feed-forward and recurrent networks

$$\text{Since } p(\mathbf{x}) = \sum_h p(\mathbf{x}, \mathbf{h}) = \sum_h p(\mathbf{x} | \mathbf{h})p(\mathbf{h}) = \mathbb{E}_{\mathbf{h}} p(\mathbf{x} | \mathbf{h})$$

Linear factor models

- Linear factor models are the simplest probabilistic models
 - They are used as building blocks for:
 - Mixture models
 - Deep probabilistic models
- They are basic approaches to build generative models that are extended by deep models
- Defined by using a stochastic, linear decoder function that generates x by adding noise to a linear transformation of h , i.e.,

$$x = Wh + b + \text{noise}$$

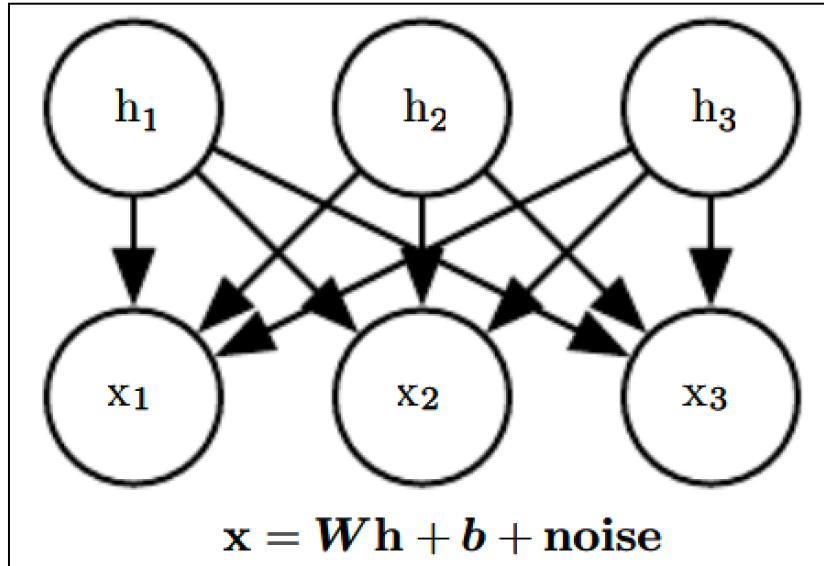
Linear Factor Model Definition

- A linear factor model describes a data generating process as follows
 - First we sample the explanatory factors \mathbf{h} from a distribution $\mathbf{h} \sim p(\mathbf{h})$
 - Where $p(\mathbf{h})$ is a factorial distribution
$$p(\mathbf{h}) = \prod_i p(h_i)$$
 - So that it is easy to sample from
 - Next we sample the real-valued observable variables given the factors

$$\mathbf{x} = \mathbf{W}\mathbf{h} + \mathbf{b} + \text{noise}$$

- where noise is Gaussian and diagonal (independent dimensions)

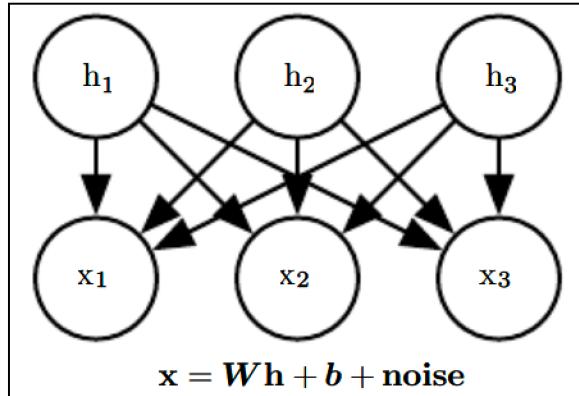
Graphical Representation of Linear Factor Model



$$\mathbf{h} \sim p(\mathbf{h}) \text{ with } p(\mathbf{h}) = \prod_i p(h_i)$$

$$\mathbf{x} = \mathbf{W}\mathbf{h} + \mathbf{b} + \text{noise}$$

Special cases of Linear Factor Model

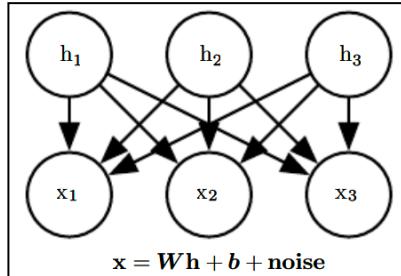


$\mathbf{h} \sim p(\mathbf{h})$ with $p(\mathbf{h}) = \prod_i p(h_i)$

$$\mathbf{x} = \mathbf{W}\mathbf{h} + \mathbf{b} + \text{noise}$$

- Special cases of above equations are:
 1. Probabilistic PCA
 2. Factor analysis
 3. Other Linear Factor Models
- They differ in choices about form of *noise* and prior over latent variables \mathbf{h} before observing \mathbf{x}
 - Factor Analysis and Probabilistic PCA shown next₁₀

Factor Analysis



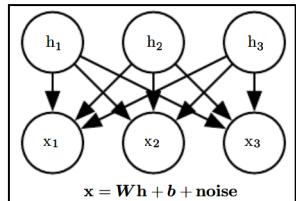
$\mathbf{h} \sim p(\mathbf{h})$ with

$$p(\mathbf{h}) = \prod_i p(h_i)$$

$$\mathbf{x} = \mathbf{W}\mathbf{h} + \mathbf{b} + \text{noise}$$

- Prior $p(\mathbf{h})$ is a unit variance Gaussian $\mathbf{h} \sim N(\mathbf{h}; \mathbf{0}, \mathbf{I})$
- x_i are conditionally independent given \mathbf{h}
 - noise from Gaussian with covariance matrix
 $\psi = \text{diag}(\sigma^2)$ with $\sigma^2 = [\sigma_1^2, \dots, \sigma_n^2]$, vector of variances
 - Latent variables capture dependencies between x_i
- It can be shown that \mathbf{x} is multivariate Gaussian
 $\mathbf{x} \sim N(\mathbf{x}; \mathbf{b}, \mathbf{W}\mathbf{W}^T + \psi)$

Probabilistic PCA

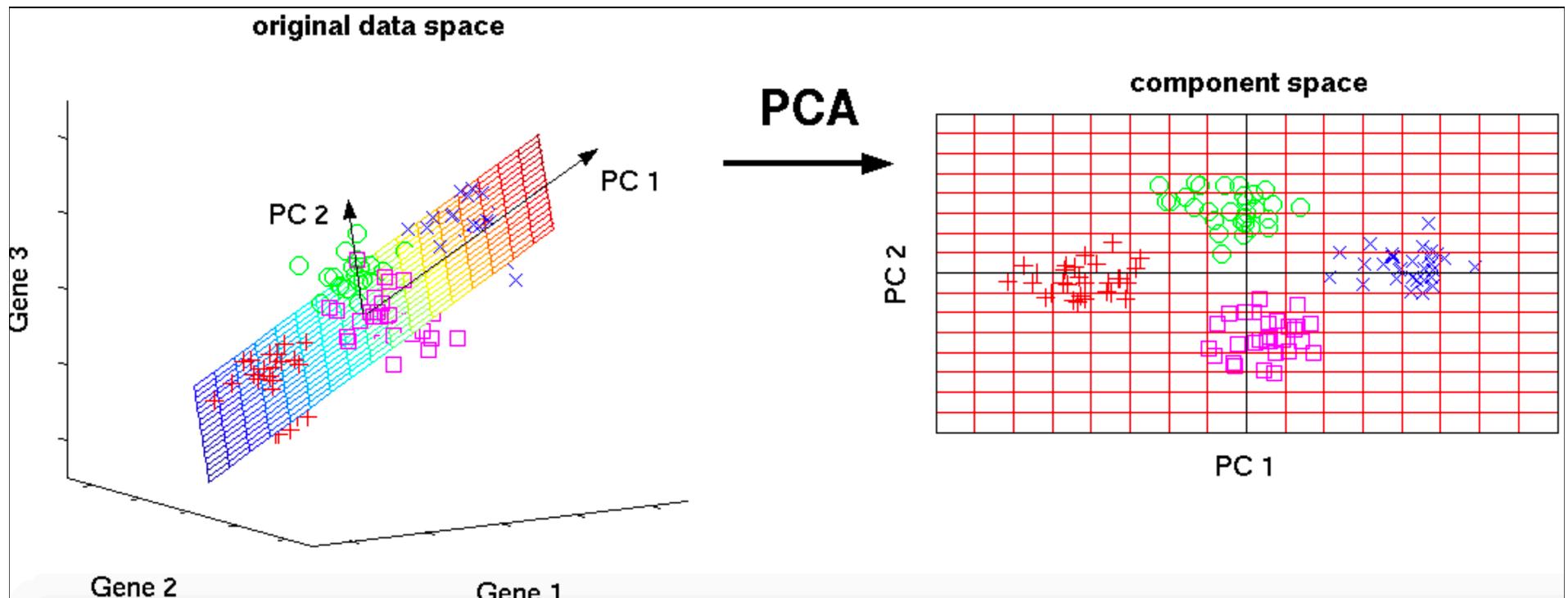


$$\mathbf{h} \sim p(\mathbf{h}) \text{ with } p(\mathbf{h}) = \prod_i p(h_i)$$

$$\mathbf{x} = \mathbf{W}\mathbf{h} + \mathbf{b} + \text{noise}$$

- A slightly modified factor analysis model
- Assume equal conditional variances: $\sigma^2 = \sigma_1^2 = \dots = \sigma_n^2$
 - Thus $\mathbf{x} \sim N(\mathbf{x}; \mathbf{b}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I})$
Or equivalently $\mathbf{x} = \mathbf{W}\mathbf{h} + \mathbf{b} + \sigma\mathbf{z}$
where $\mathbf{z} \sim N(\mathbf{z}; \mathbf{0}, \mathbf{I})$ is Gaussian noise
 - Iterative EM can be used to estimate \mathbf{W} and σ^2
 - Takes advantage of observation that most variations are captured by the latent variables \mathbf{h} , upto small residual reconstruction error σ^2
- Probabilistic PCA becomes PCA as $\sigma \rightarrow 0$

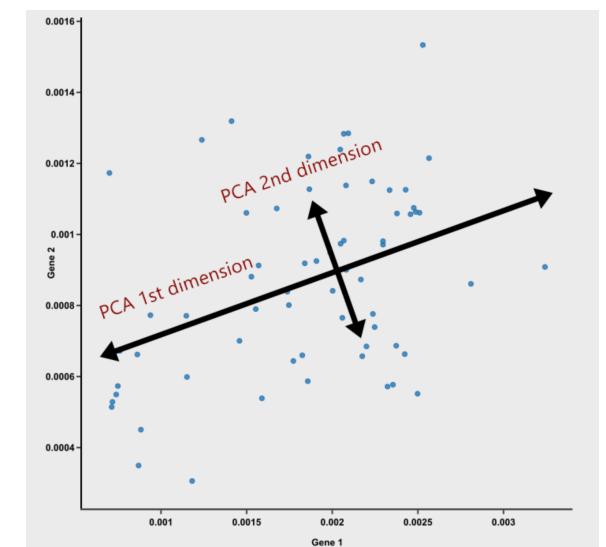
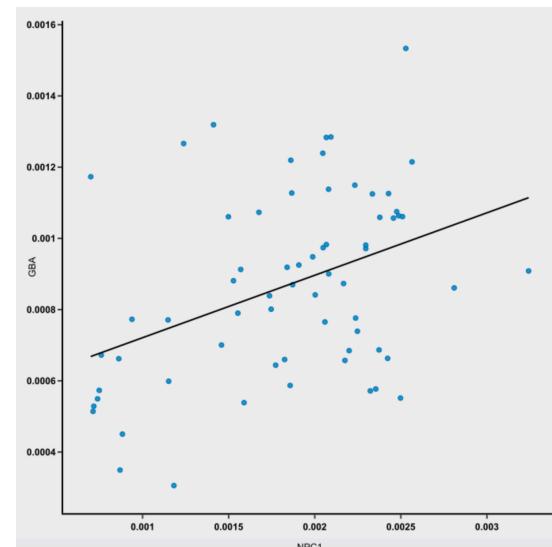
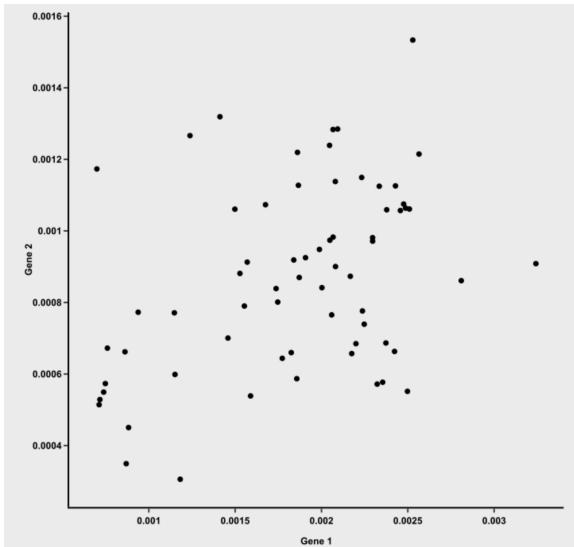
PCA (Principal Components Analysis)



PCA

Principal components capture the most variation in a dataset

PCA deals with the curse of dimensionality by capturing the essence of data into a few principal components.



PC1 must convey the *maximum Variation* among data points and contain *minimum error*.

PC2 is the second line that meets PC1, perpendicularly, at the center of the cloud, and describes second most variation in the data

PCA Algorithm (Linear Algebra)

- Given $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ in R^n represent using $R^l \quad l < n$
 - For point $\mathbf{x}^{(i)}$ find code vector $\mathbf{c}^{(i)}$ in R^l
 - Find encoder $f(\mathbf{x}) = \mathbf{c}$ and decoder $\mathbf{x} \approx g(f(\mathbf{x}))$
 - One decoding function is: $g(\mathbf{c}) = D\mathbf{c}$ where
 - D is a matrix with l mutually orthogonal columns to minimize distance between \mathbf{x} and reconstruction
 $r(\mathbf{x}) = g(f(\mathbf{x})) = DD^T\mathbf{x}$

$$\mathbf{c}^* = \arg \min_{\mathbf{c}} \|\mathbf{x} - g(\mathbf{c})\|_2$$

$$D^* = \arg \min_D \left(\sum_{i,j} \left(\mathbf{x}_j^{(i)} - r(\mathbf{x}_j^{(i)}) \right)^2 \right)^{\frac{1}{2}}$$

- Solution D
 - The l eigenvectors of design matrix X correspond to the largest eigenvalues

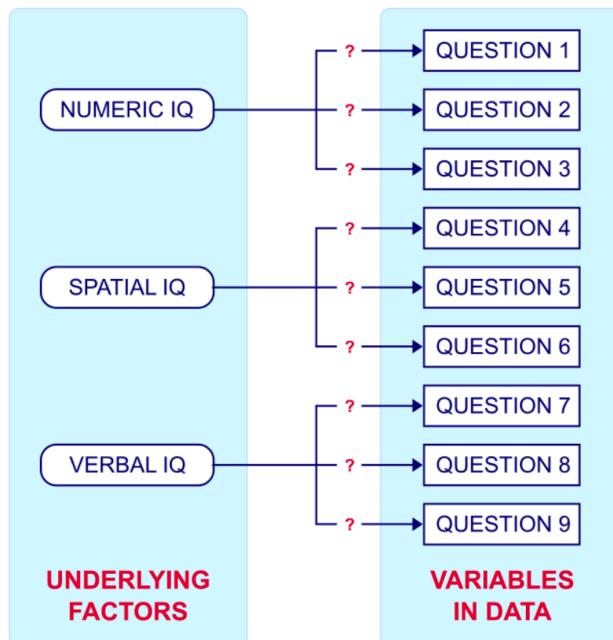
$$X \in \mathbb{R}^{m \times n}$$

Confirmatory Factor Analysis

Factor analysis is a technique for identifying which underlying factors are measured by a (much larger) number of observed variables.

Such “underlying factors” are difficult to measure , e.g., IQ, depression or extraversion.

Researcher's Hypothesis



Confirmatory factor analysis

	Question 1	Question 2	Question 3	Question 4	Question 5	Question 6	Question 7	Question 8	Question 9
Question 1	1.00	.75	.76	.04	.11	.10	.04	-.06	.01
Question 2	.75	1.00	.78	-.01	.00	.02	.00	-.06	.02
Question 3	.76	.78	1.00	-.06	-.03	-.02	.08	-.05	.02
Question 4	.04	-.01	-.06	1.00	.85	.82	.10	.00	.05
Question 5	.11	.00	-.03	.85	1.00	.86	-.06	-.08	-.04
Question 6	.10	.02	-.02	.82	.86	1.00	.04	.04	.02
Question 7	.04	.00	.08	.10	-.06	.04	1.00	.71	.78
Question 8	-.06	-.06	-.05	.00	-.08	.04	.71	1.00	.78
Question 9	.01	.02	.02	.05	-.04	.02	.78	.78	1.00

if questions 1, 2 and 3 all measure numeric IQ, then the Pearson correlations among these items should be substantial: respondents with high numeric IQ will typically score high on all 3 questions

Exploratory Factor Analysis

No clue to which -or even how many- factors are represented by the data

Exploratory Factor Analysis

- Psychologist's hypothesis: there are two kinds ($k=2$) of latent intelligence
 - Verbal (factor F_1) and mathematical (factor F_2)
- Evidence for hypothesis is sought in the examination scores (\mathbf{x}) from $p=6$ academic fields (e.g., astronomy) of $n=1000$ students
 - Observable variables x_1, \dots, x_6 with means μ_1, \dots, μ_6
 $x_i - \mu_i = l_{i1}F_1 + l_{i2}F_2 + \varepsilon_i, \quad i = 1, \dots, 6, \quad l_i$ are the loadings
- In matrix form $\mathbf{x} - \boldsymbol{\mu} = \mathbf{LF} + \boldsymbol{\varepsilon}$; \mathbf{x} is $p \times n$, \mathbf{L} is $p \times k$, \mathbf{F} is $k \times n$
 - Values of \mathbf{L} , $\boldsymbol{\mu}$, and variances of errors $\boldsymbol{\varepsilon}$ must be estimated from data \mathbf{x} and \mathbf{F} (assumption about levels of the factors is fixed for a given \mathbf{F})
 - Solution: for astronomy, average student aptitude is $10F_1 + 6F_2$

Factor Analysis

- A method to describe variability among observed, correlated variables in terms of a lower no. of latent variables called *factors*
 - E.g., variations in six observed variables mainly reflect the variations in two latent variables
- Observed variables modeled as linear combinations of latent factors, plus error terms
 - Factor analysis aims to find independent latent variables

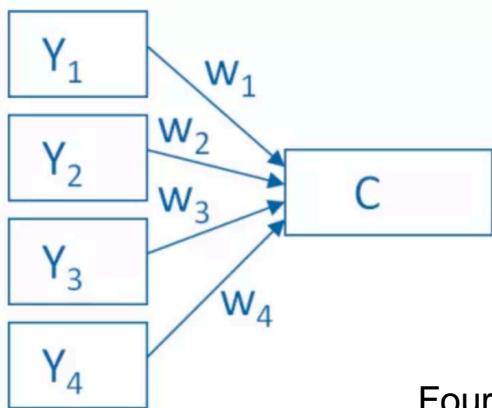
PCA vs. Factor Analysis

- Both are data reduction techniques
- Both involve choosing components or factors
- Fundamental difference between them:
 - PCA is a linear combination of variables
 - Factor Analysis is a measurement model of a latent variable
- PCA is a more basic version of Factor Analysis

PCA vs Factor Analysis

- Principal Components Analysis

- create index variables from larger set of measured variables



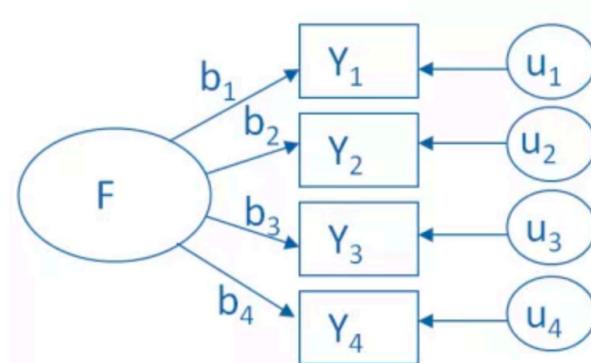
Four measured variables y combined into a single component c

Model set up as:

$$c = w_1y_1 + w_2y_2 + w_3y_3 + w_4y_4$$

- Factor Analysis

- A model for measuring an unobservable latent variable



F , the latent Factor, is causing the responses on the four measured y variables, u 's are the variance in each y that is unexplained by the factor.

Model set up as
Regression equations:

$$\begin{aligned} y_1 &= b_1 * F + u_1 \\ y_2 &= b_2 * F + u_2 \\ y_3 &= b_3 * F + u_3 \\ y_4 &= b_4 * F + u_4 \end{aligned}$$

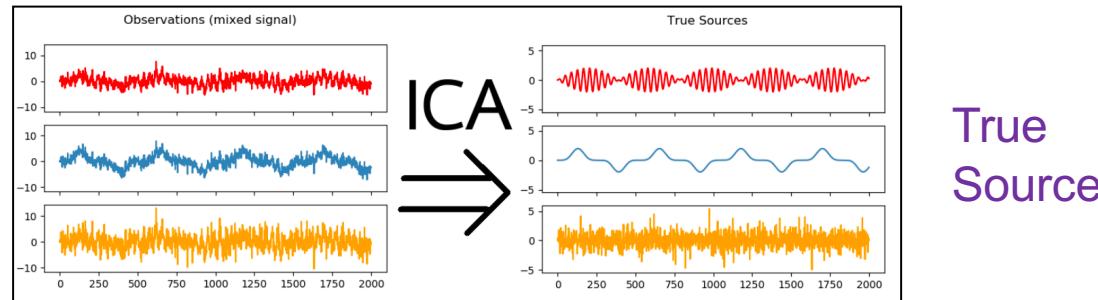
Independent Component Analysis

- Approach to modeling linear factors
- To separate observed signal into underlying independent signals
 - That are scaled and added together to form the observed data

Examples of ICA

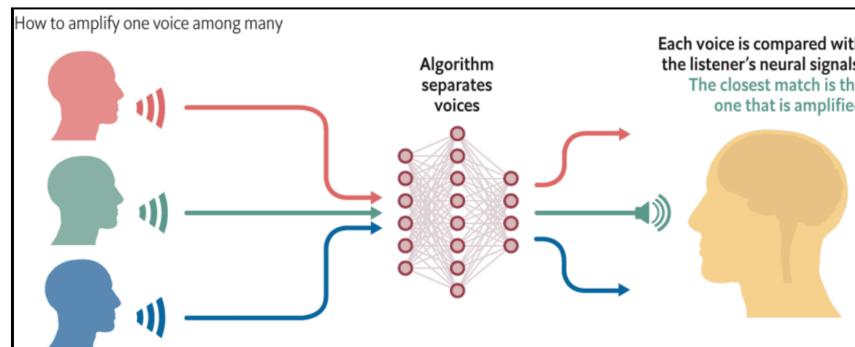
1. Extracting source from noisy signal

Mixed
Signal



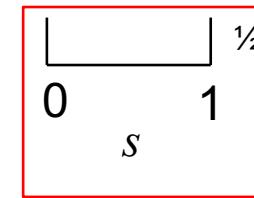
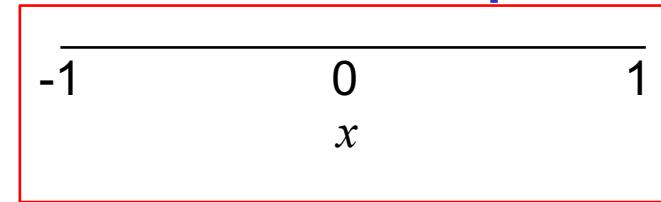
True
Source

2. Cocktail party problem: speech signals of people talking simultaneously are separated



ICA requires independent signals

- Signals are intended to be fully independent rather than merely decorrelated from each other
 - Independence is stronger than zero covariance
- Ex: No covariance doesn't mean independence
 - We sample x from $[-1,1]$
 - Let s be 1 with probability 0.5,
otherwise $s = 0$
 - Let $y=sx$
 - Clearly x and y not independent, since y generated from x
 - But x and y have zero covariance



An ICA model

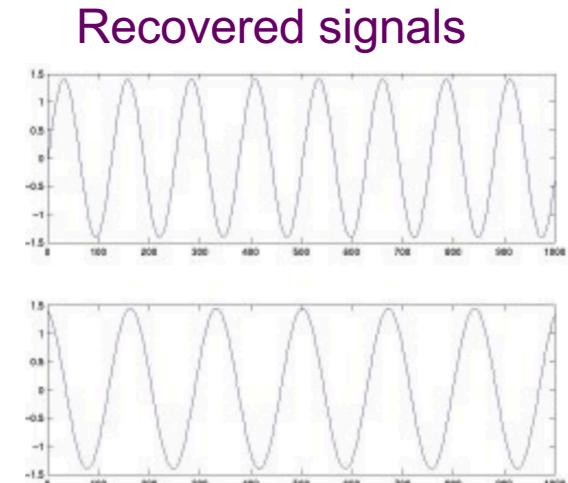
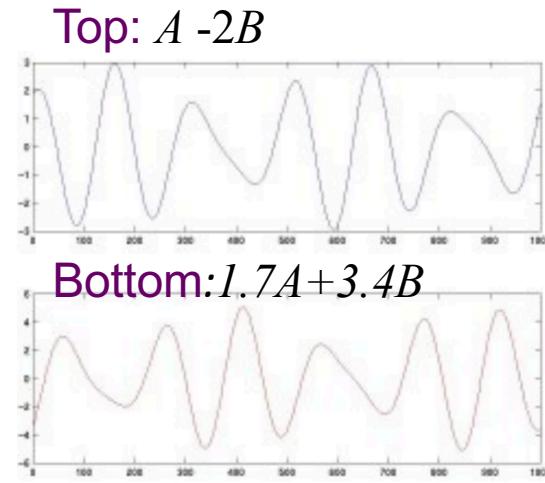
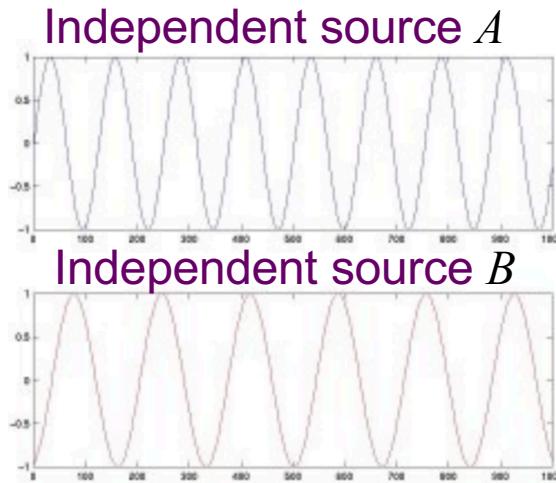
- Prior $p(\mathbf{h})$ fixed ahead of time
- Model deterministically generates $\mathbf{x} = \mathbf{W}\mathbf{h}$
 - Use nonlinear change of variables to determine $p(\mathbf{x})$

$$p_{\mathbf{x}}(\mathbf{x}) = p_{\mathbf{y}}(g(\mathbf{x})) \left| \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \right|$$

- Learning proceeds using maximum likelihood
- By choosing independent $p(\mathbf{h})$ we can recover underlying factors that are close to independent
 - Used to recover low level signals mixed together

ICA signal separation

- Each example is one moment in time
- Each x_i is a sensor observation of mixed signals
- Each h_i is one estimate of the original signals



Choice of $p(\mathbf{h})$ in ICA

- All ICA variants require $p(\mathbf{h})$ be non-Gaussian
 - This is because if $p(\mathbf{h})$ is an independent prior with Gaussian components then W is not identifiable
- This is different from probabilistic PCA and factor analysis, where $p(\mathbf{h})$ is Gaussian
- Typical choice is $p(h_i) = [d/dh_i]\sigma(h_i)$
 - Have larger peaks near 0 than does Gaussian
 - So ICA is learning sparse features

Generalization of ICA

- PCA generalizes to nonlinear autoencoders
- ICA generalizes to a nonlinear generative model
 - Use a nonlinear f to generate observed data

Slow Feature Analysis

- It is a Linear factor model
- Uses information from time signals to learn invariant features
- Motivation: Slowness principle
 - Important characteristics change slowly compared to individual measurements that make up a scene
 - Computer vision example shown next

SFA in computer vision

- Individual pixels can change very rapidly
- Ex: zebra moves from right to left
 - Pixels change rapidly from black to white to black
 - Feature indicating whether zebra is in image changes slowly
- Regularize model to learn features that change slowly with time



Slowness Principle

- Can apply slowness principle to any model trained with gradient descent
- Slowness principle is introduced by adding a term to the cost function of the form

$$\lambda \sum_t L(f(\mathbf{x}^{(t+1)}), f(\mathbf{x}^{(t)}))$$

- where f is feature extractor to be regularized
- λ is the strength of the slowness regularization term
- L is a loss function measuring the distance between $f(\mathbf{x}^{(t)})$ and $f(\mathbf{x}^{(t+1)})$
 - Common choice of L is the mean squared difference

Sparse Coding

- A linear factor model
- Studied as unsupervised feature learning and extraction
- Terminology
 - Sparse Coding refers to inferring the values of h in the model
 - Sparse modeling refers to process of designing and learning the model
 - But sparse coding often refers to both

Sparse Coding definition

- It uses a linear decoder plus noise to obtain reconstructions of x as specified by

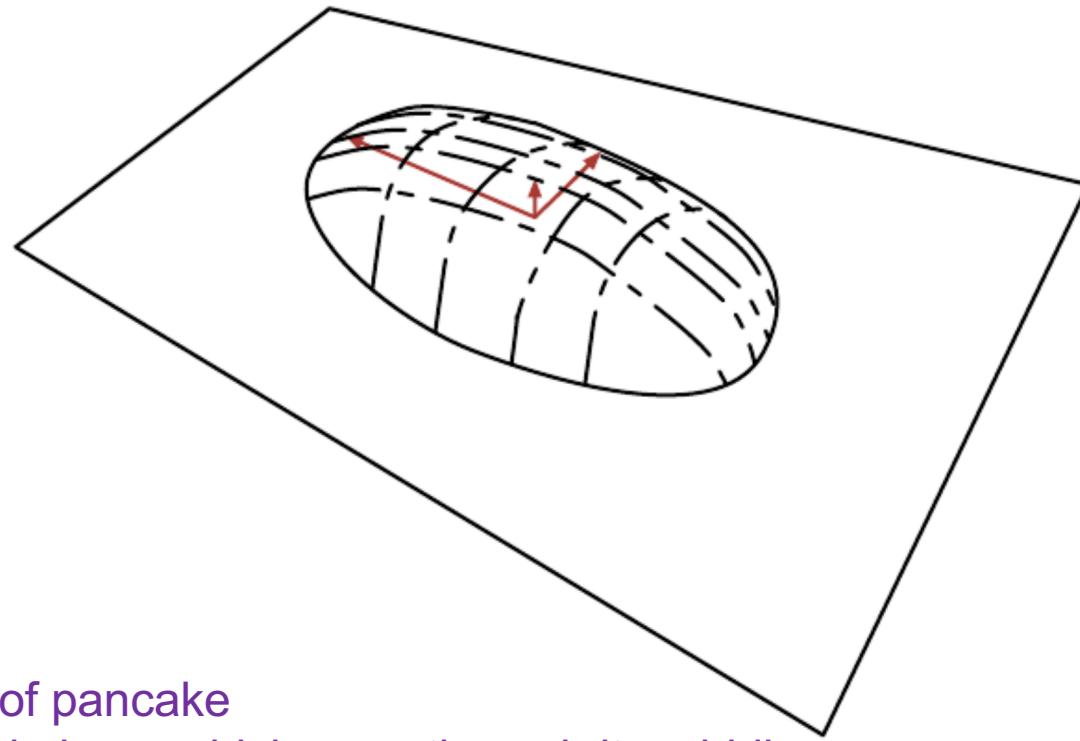
$$\mathbf{x} = \mathbf{W}\mathbf{h} + \mathbf{b} + \text{noise}$$

- Sparse coding models typically assume that the linear factors have Gaussian noise with isotropic precision β
 - $p(\mathbf{x}|\mathbf{h}) = N(\mathbf{x}; \mathbf{W}\mathbf{h} + \mathbf{b}, (1/\beta)\mathbf{I})$
 - $p(\mathbf{h})$ is chosen to be one with sharp peaks near 0
 - ℓ_1 or ℓ_0 prior parameterized with sparsity penalty

Manifold Interpretation of PCA

- Linear factor models including PCA and factor analysis can be interpreted as learning a manifold
- Probabilistic PCA learns a flat pancake of high probability
- Illustrated next

Flat Gaussian near low-dimensional manifold



Shows upper half of pancake
above the manifold plane, which goes through its middle

Variance orthogonal to manifold is small (can be considered as noise)
while other variances are large (correspond to signal)

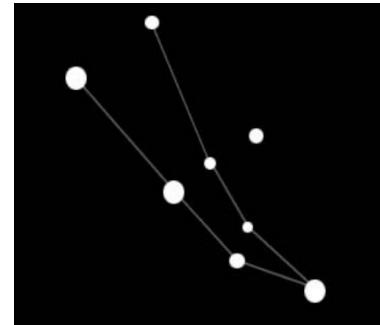
Generality of Interpretation

- Manifold interpretation applies to not just to PCA but also to any linear autoencoder that learns Matrices W and V with the goal of making the reconstruction of \mathbf{x} lie as close to \mathbf{x} as possible
- Let the encoder be $\mathbf{h} = f(\mathbf{x}) = W^T(\mathbf{x} - \mu)$
- The encoder computes a low-dimensional representation of \mathbf{h}
- With the autoencoder view, we have a decoder computing the reconstruction $\hat{\mathbf{x}} = g(\mathbf{h}) = \mathbf{b} + V\mathbf{h}$

Summary of Linear Factor Models

- Linear factor models are
 - The simplest generative models
 - Simplest models that learn a representation of data
- Analogy between linear classifiers and linear factor models
 1. Linear classifier/regression models are extended to deep feedforward networks
 2. Linear factor models are extended to autoencoder networks and deep probabilistic models
 - Perform the same tasks but with a much more powerful and flexible model family

Distribution of stars: Galaxy M31 in Andromeda



Andromeda
constellation



M31 is 2.1 million light years away and heading on a collision course with the Milky Way. They should collide in about 4 billion years. We won't feel much from the mash-up as there is so much empty space between the stars of both galaxies that few if any will notice.

The 3-dimensional data is largely present on a 2-dimensional plane. Both PCA and Factor Analysis aim to find the plane using different approaches.