Inference and Representation
Lecture 5

Joan Bruna
Courant Institute, NYU

Lecture 4 Reminder

- Approximate Sampling is the only viable strategy on large, complicated graphical models.

- We saw two techniques:
  - Monte-Carlo methods.
  - Gibbs Sampling (also covered in PS4).

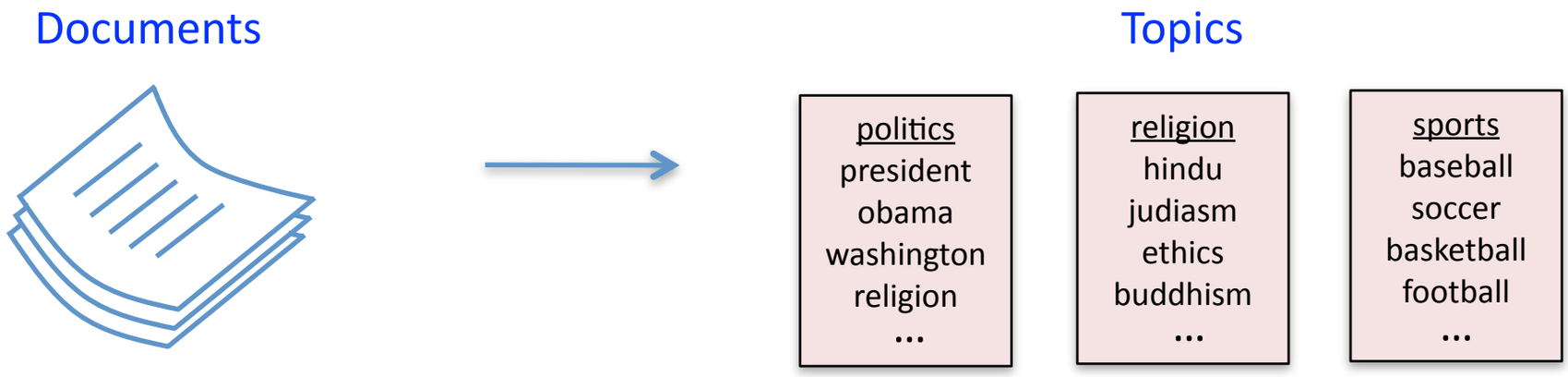
Reminder: Gibbs Sampler

- The GS algorithm:
  1. Suppose the graphical model contains variables $x_1,\ldots,x_n$
  2. Initialize starting values for $x_1,\ldots,x_n$
  3. Do until convergence:
     1. Pick an ordering of the n variables (can be fixed or random)
     2. For each variable x_i in order:
        1. Sample $x \sim P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, i.e. the conditional distribution of x_i given the current values of all other variables
        2. Update $x_i \leftarrow x$

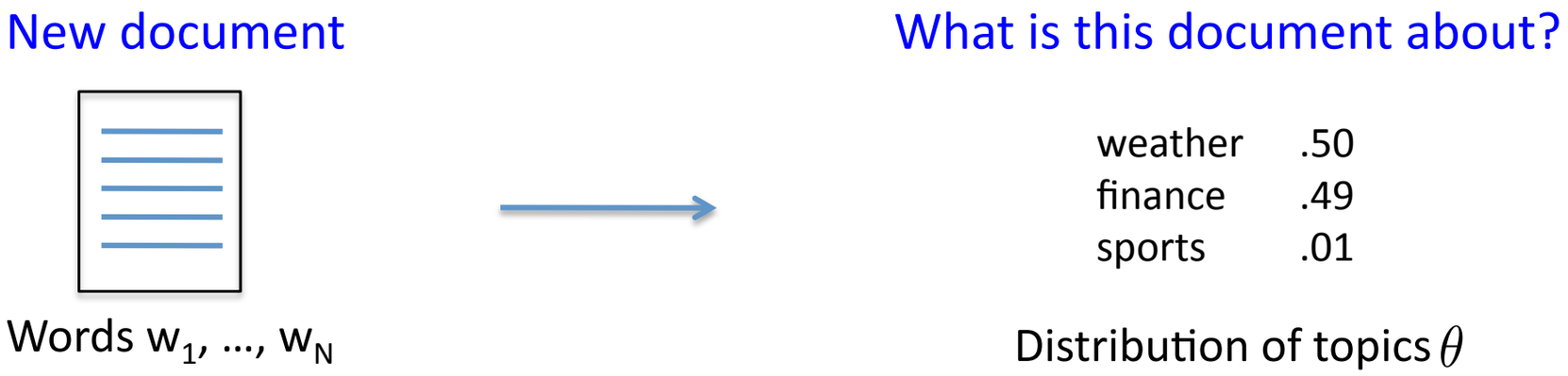
- When we update x_i, we <u>immediately</u> use its new value for sampling other variables x_j

Reminder: Topic Modeling

- **Topic models** are powerful tools for exploring large data sets and for making inferences about the content of documents



- Many applications in information retrieval, document summarization, and classification



- LDA is one of the simplest and most widely used topic models

Reminder: Latent Dirichlet Allocation

1. Sample the document's **topic distribution** θ (aka topic vector)

$$\theta \sim \text{Dirichlet}(\alpha_{1:T})$$

where the $\{\alpha_t\}_{t=1}^T$ are fixed hyperparameters. Thus θ is a distribution over T topics with mean $\theta_t = \alpha_t / \sum_{t'} \alpha_{t'}$

2. For $i = 1$ to N, sample the **topic** z_i of the i'th word

$$z_i | \theta \sim \theta$$

3. ... and then sample the actual **word** w_i from the z_i'th topic

$$w_i | z_i \sim \beta_{z_i}$$

where $\{\beta_t\}_{t=1}^T$ are the *topics* (a fixed collection of distributions on words)

Reminder: LDA

Topics

Documents

Topic proportions and assignments

β_1

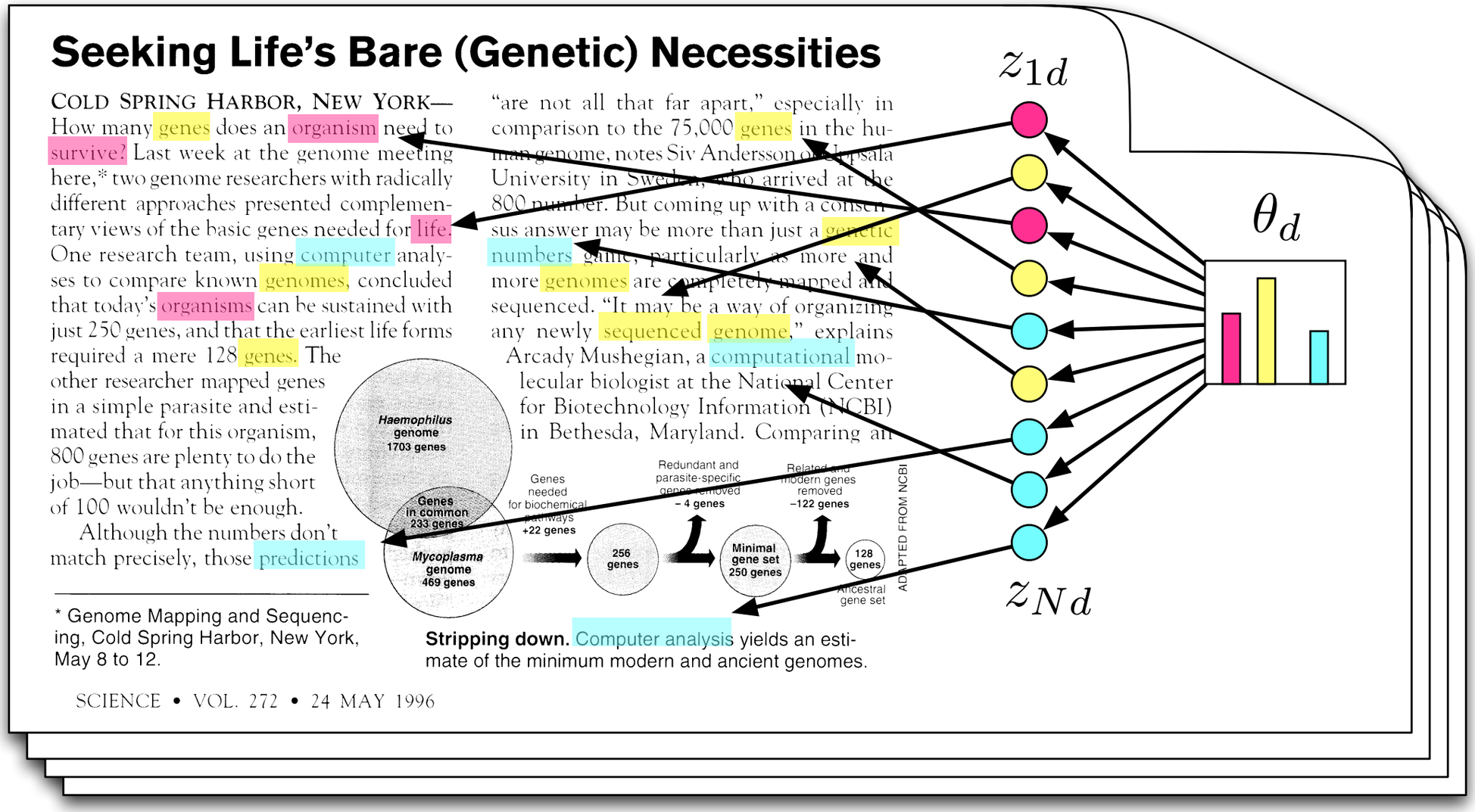
gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

β_T

data	0.02
number	0.02
computer	0.01
...	



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

z_{1d}

θ_d

z_{Nd}

(Blei, *Introduction to Probabilistic Topic Models*, 2011)

Lecture 5 Objectives

- Modeling Survey Data

- Factor Analysis
  - Principal Component Analysis
  - Independent Component Analysis

- Gaussian Mixture Models

- The Expectation-Maximization (EM) algorithm

Survey Data

- How are these statements inferred?

FiveThirtyEight

Politics Sports Science & Health Economics Culture

MAY 3, 2016 AT 2:45 PM

The Mythology Of Trump's 'Working Class' Support

His voters are better off economically compared with most Americans.

By Matt Silver

Filed under 2016 Election

FiveThirtyEight

Politics Sports Science & Health Economics Culture

MAR 14, 2016 AT 6:02 AM

What Trump Supporters Were Doing Before Trump

By Dan Hopkins

Filed under 2016 Election

JUNE 2, 2016

Election 2016

More 'warmth' for Trump among GOP voters concerned by immigrants, diversity

BY BRADLEY JONES AND JOCELYN KILEY | 10 COMMENTS

Survey Data

- We typically see these sort of data

How the GOP candidates' supporters differed on issues

POSITION	YEAR	STANCE OF SUPPORTERS: TRUMP	CRUZ	RUBIO
Raise taxes on rich	2007	0.25	0.24	0.27
Pro-gay marriage	2007	0.31	0.20	0.28
Conservative ideology	2007	0.64	0.76	0.67
Pro-choice	2007	0.63	0.39	0.42
Stay in Iraq	2007	0.62	0.81	0.67
Hawk (vs. dove)	2006	0.68	0.67	0.52
No special help for blacks	2012	0.86	0.82	0.80
Obama rating	2012	0.21	0.17	0.20
Anti-Obamacare	2012	0.76	0.82	0.80
Very critical of system	2012	0.70	0.64	0.62
Pro-government spending	2012	0.20	0.15	0.15
Create pathway to citizenship	2012	0.21	0.29	0.37
Anti-Hispanic prejudice	2012	0.53	0.50	0.49
Anti-black prejudice	2012	0.58	0.54	0.55
Pro-NAFTA	2012	0.40	0.50	0.52

Stance on a position ranges from 0-1, with 0 being totally against and 1 being totally in agreement with

SOURCE: HOPKINS/MUTZ

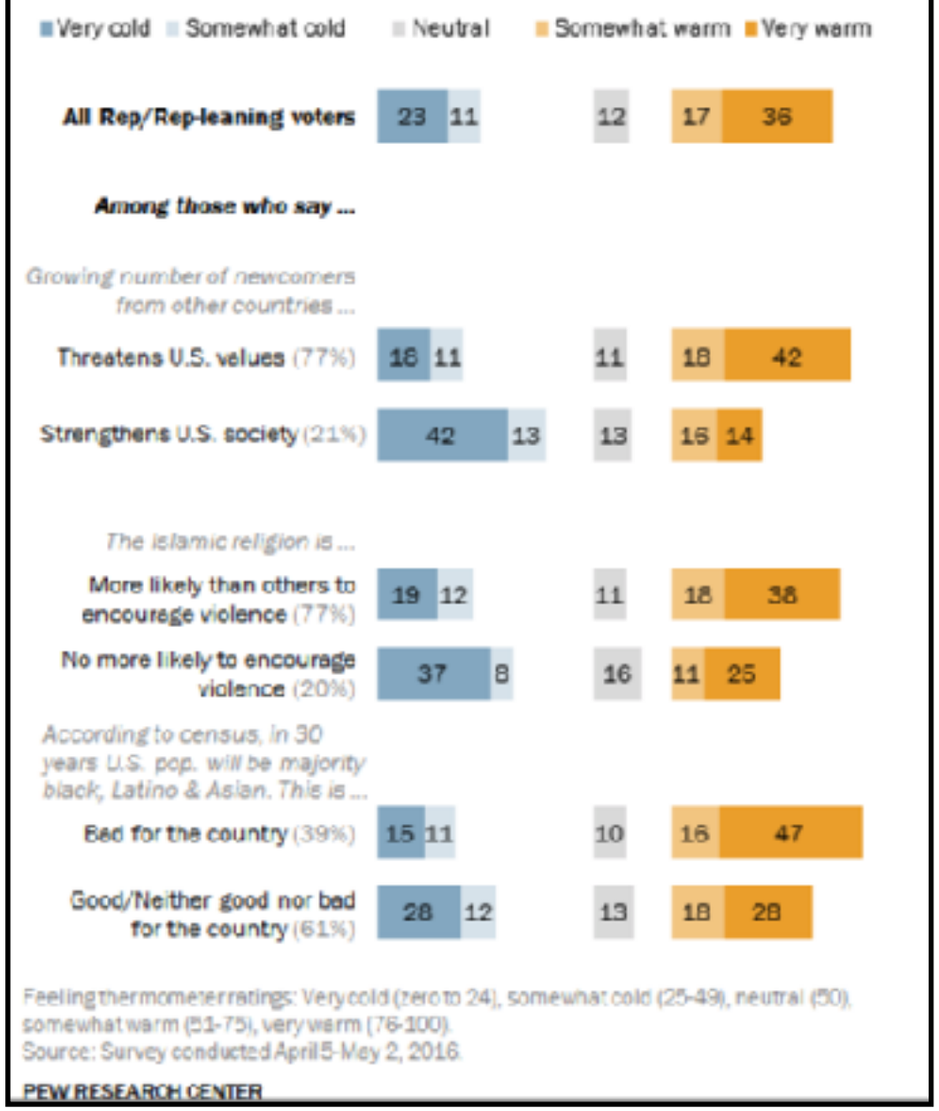
Share of Republican electorate with household income below $50,000

STATE	2012	2016
Alabama	37%	41%
Florida	34	33
Georgia	24	26
Illinois	28	23
Maryland	19	19
Massachusetts	24	20
Michigan	35	37
Mississippi	36	37
New Hampshire	26	27
Ohio	32	30
Oklahoma	41	30
South Carolina	36	27
Tennessee	35	33
Vermont	37	30
Virginia	25	19
Wisconsin	32	28
Average	31	29

SOURCE: EDISON RESEARCH EXIT POLLS

Within GOP, views of immigration, Islam, diversity strongly associated with ratings of Trump

% of Republican and Republican-leaning registered voters who rate Trump on a 'feeling thermometer' from 0 (coldest rating) to 100 (warmest rating) ...



Feeling thermometer ratings: Very cold (zero to 24), somewhat cold (25-49), neutral (50), somewhat warm (51-75), very warm (76-100).
Source: Survey conducted April 5-May 2, 2016.

PEW RESEARCH CENTER

Survey Data: other prime example

- Medical surveys
- Financial Markets
- EEG recordings

- Children's Depression Inventory [3]
  - 27 items scored 0,1,2 assessing aspects of depressive symptoms for children and adolescents
  - 1 total scale
    - sum of the 27 items after reverse coding 13 of them
    - higher scores indicate higher depressive symptom levels
  - 5 subscales measuring different aspects of depressive symptoms
    - negative mood, interpretation problems, ineffectiveness, anhedonia, and negative self-esteem
    - the total scale equals the sum of the subscales
  - total scale used in practice rather than subscales

(credit: G. Knafl, ohsu)

Factor Analysis for Survey Data

- **Goal**: extract interpretable, summary information out of a series of correlated survey responses.

- Factor Analysis refers to a series of statistical techniques to achieve that.

- That is, given answers $x_1, \ldots x_L$ to L questions, infer latent variables (=factors) that explain the underlying phenomena under study.

Principal Component Analysis

- We start with the simplest setting: suppose that

$$X_l = \sum_{j=1}^{J} \alpha_{j,l} Y_j \ , \ l = 1 \ldots L \ .$$

with $Y_{1 \ldots J}$ iid and $X_{1 \ldots L}$ jointly Gaussian.

- Q: How to discover the 'latent' factors Y?

Principal Component Analysis

- We start with the simplest setting: suppose that

$$X_l = \sum_{j=1}^{J} \alpha_{j,l} Y_j \ , \ l = 1 \ldots L \ .$$

$$\text{with } Y_{1\ldots J} \text{ iid} \quad \text{and } X_{1\ldots L} \text{ jointly Gaussian.}$$

- Q: How to discover the 'latent' factors Y?
- Observation 1:

$$\mathbf{X} \text{ Gaussian} \Rightarrow \mathbf{Y} = A\mathbf{X} \text{ also Gaussian.}$$

Principal Component Analysis

- We start with the simplest setting: suppose that

$$X_l = \sum_{j=1}^{J} \alpha_{j,l} Y_j \ , \ l = 1 \dots L \ .$$

with $Y_{1\dots J}$ iid and $X_{1\dots L}$ jointly Gaussian.

- Q: How to discover the 'latent' factors Y?

- Observation 1:

$$\mathbf{X} \text{ Gaussian} \Rightarrow \mathbf{Y} = A\mathbf{X} \text{ also Gaussian.}$$

- Observation 2:

$$\text{If } \mathbf{Y} = (Y_1, \dots, Y_J) \text{ is jointly Gaussian,}$$
$$Y_i, Y_j \text{ independent} \Leftrightarrow Y_i, Y_j \text{ decorrelated.}$$

Principal Component Analysis

- We define $\mu_X = \mathbb{E}(X)\ ,\ \Sigma_X = \mathbb{E}\{(X - \mu_X)(X - \mu_X)^T\}\ .$
- **Reminder:** Let $\mathbf{Y} = A\mathbf{X} + b$. Then
  - $\mu_Y = A\mu_X + b.$
  - $\Sigma_Y = A\Sigma_X A^T.$

Principal Component Analysis

- We define $\mu_X = \mathbb{E}(X)\ ,\ \Sigma_X = \mathbb{E}\{(X - \mu_X)(X - \mu_X)^T\}\ .$
- **Reminder:** Let $\mathbf{Y} = A\mathbf{X} + b$. Then
  - $\mu_Y = A\mu_X + b.$
  - $\Sigma_Y = A\Sigma_X A^T.$

- In our previous model, if $A_{l,j} = \alpha_{l,j}$, we have $X = AY$.
- Hence $\Sigma_Y = A\Sigma_X A^T.$
- Q: How to find A such that $A\Sigma_X A^T$ defines an uncorrelated random vector?

Principal Component Analysis

Reminder: [Real Spectral Theorem]: If $A \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite ($A \succ 0$), then A diagonalizes
in a real orthonormal basis with non-negative eigenvalues.

Principal Component Analysis

Reminder: [Real Spectral Theorem]: If $A \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite ($A \succ 0$), then A diagonalizes
in a real orthonormal basis with non-negative eigenvalues.

Fact: The covariance operator $\Sigma_X = \mathbb{E}((X - \mu_X)(X - \mu_X)^T)$
is **symmetric** and **positive semidefinite**.

Principal Component Analysis

Reminder: [Real Spectral Theorem]: If $A \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite ($A \succ 0$), then A diagonalizes in a real orthonormal basis with non-negative eigenvalues.

Fact: The covariance operator $\Sigma_X = \mathbb{E}((X - \mu_X)(X - \mu_X)^T)$ is **symmetric** and **positive semidefinite**.

Therefore, Σ_X admits a real eigenbasis:

$$\Sigma_X = U \Lambda U^T \ , \ \Lambda = \mathrm{diag}(\lambda_i) \succeq 0 \ .$$

Thus $U^T \Sigma_X U = \Lambda$.

Principal Component Analysis

Reminder: [Real Spectral Theorem]: If $A \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite ($A \succ 0$), then A diagonalizes in a real orthonormal basis with non-negative eigenvalues.

Fact: The covariance operator $\Sigma_X = \mathbb{E}((X - \mu_X)(X - \mu_X)^T)$ is **symmetric** and **positive semidefinite**.

Therefore, Σ_X admits a real eigenbasis:

$$\Sigma_X = U \Lambda U^T \ , \ \Lambda = \mathrm{diag}(\lambda_i) \succeq 0 \ .$$

Thus $U^T \Sigma_X U = \Lambda$.

Moreover, we can write $\Lambda = S \cdot S$, with $s_{i,i} = \sqrt{\lambda_i}$.
If $\min_i \lambda_i > 0$, it results that $\tilde{U} = U S^{-1}$ satisfies $\tilde{U}^T \Sigma_X \tilde{U} = \mathbf{1}$.

Principal Component Analysis

- We have just shown that $\mathbf{Y} = \tilde{U}(\mathbf{X} - \mu_X)$ provides variables that are uncorrelated and jointly Gaussian, thus independent.

Principal Component Analysis

- We have just shown that $\mathbf{Y} = \tilde{U}(\mathbf{X} - \mu_X)$ provides variables that are uncorrelated and jointly Gaussian, thus independent.

- Remarks
  - The decomposition is not unique: Any orthogonal transformation of Y also satisfies the same property.
  - PCA provides linear compression: if $J < \mathbf{rank}(\Sigma_X)$, what is the best linear approximation of X with J independent components?

$$\min_{A \in \mathbb{R}^{L \times J}} \mathbb{E}(\|X - AX\|^2) \ .$$

$$A = \{ \text{ eigenvectors of } \Sigma_X \text{ corresponding to } J \text{ largest eigenvalues.} \}$$

(again, A is determined up to an orthogonal transformation)

Estimating the Principal Components

- So far, we have seen how to extract information from the covariance of X.

Estimating the Principal Components

- So far, we have seen how to extract information from the covariance of X.
- In practice, we will observe $x_1, \ldots, x_N$ iid samples of X
- Empirical Covariance:

$$\widehat{\Sigma}_N = \frac{1}{N} \sum_{n<N} (x_n - \hat{\mu})(x_n - \hat{\mu})^T \ . \quad \in \mathbb{R}^{L \times L} \ .$$

Estimating the Principal Components

- So far, we have seen how to extract information from the covariance of X.
- In practice, we will observe $x_1, \ldots, x_N$ iid samples of X
- Empirical Covariance:

$$\widehat{\Sigma}_N = \frac{1}{N} \sum_{n<N} (x_n - \hat{\mu})(x_n - \hat{\mu})^T \ . \quad \in \mathbb{R}^{L \times L} \ .$$

- $\widehat{\Sigma}_N$ is symmetric, positive definite. (why?)

- Estimated Principal Components:

$$\widehat{\Sigma}_N = \widehat{U} \widehat{\Lambda} \widehat{U}^T \ .$$

Estimating the Principal Components

- Q: How good are these estimates?
  - i.e. for a desired accuracy ϵ, how many samples N=N(L) are required?

Estimating the Principal Components

- Q: How good are these estimates?
  - i.e. for a desired accuracy ϵ, how many samples N=N(L) are required?

- **Theorem** [Vershynin]: For distributions with bounded order-q moment, the empirical covariance satisfies

$$\|\widehat{\Sigma}_N - \Sigma\| \lesssim O(\log\log N)^2 \left(\frac{N}{L}\right)^{1/2-2/q}.$$

Estimating the Principal Components

- Q: How good are these estimates?
  - i.e. for a desired accuracy ϵ, how many samples N=N(L) are required?

- **Theorem** [Vershynin]: For distributions with bounded order-q moment, the empirical covariance satisfies

$$\|\widehat{\Sigma}_N - \Sigma\| \lesssim O(\log\log N)^2 \left(\frac{N}{L}\right)^{1/2-2/q}.$$

- It results that for a desired approximation $\|\widehat{\Sigma}_N - \Sigma\| \leq \epsilon$ we need $O((\log\log L)^\alpha L) \approx O(L)$ samples.

- **Very Important Consequence: PCA does not suffer from the curse of dimensionality!**

Estimating Principal Components

- Q: What is the computational complexity of computing PCA?
  - Naïve estimation of covariance: $O(NL^2)$
  - Diagonalizing covariance: $O(L^3)$
  - So, in big data applications we are doomed.

Estimating Principal Components

- Q: What is the computational complexity of computing PCA?
  - Naïve estimation of covariance: $O(NL^2)$
  - Diagonalizing covariance: $O(L^3)$
  - So, in big data applications we are doomed.

- Reminder: Given a data matrix, the principal components are directly given by

The **Singular Value Decomposition** (SVD) of $B \in \mathbb{R}^{n \times p}$ is defined as $B = U \Lambda V^T$, with

$$U \in \mathbb{R}^{n \times n}, V \in \mathbb{R}^{p \times p}, \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{\min(n,p)}),$$
$$UU^T = \mathbf{1}, V^T V = \mathbf{1} \ .$$

Computing the first p principal components costs $O(pNL)$.

Estimating Principal Components

- Q: What is the computational complexity of computing PCA?
  - Naïve estimation of covariance: $O(NL^2)$
  - Diagonalizing covariance: $O(L^3)$
  - So, in big data applications we are doomed.

- Reminder: Given a data matrix, the principal components are directly given by

The **Singular Value Decomposition** (SVD) of $B \in \mathbb{R}^{n \times p}$ is defined as $B = U \Lambda V^T$, with

$$U \in \mathbb{R}^{n \times n}, V \in \mathbb{R}^{p \times p}, \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{\min(n,p)}),$$
$$UU^T = \mathbf{1}, V^T V = \mathbf{1} \ .$$

Computing the first p principal components costs $O(pNL)$.

- Alternatives?

Interlude: Randomized PCA

- Suppose we suspect that the data matrix $X \in \mathbb{R}^{N \times L}$ has rank $p \ll \min(N, L)$.
- Q: Can we leverage that prior into a faster algorithm?

Interlude: Randomized PCA

- Suppose we suspect that the data matrix $X \in \mathbb{R}^{N \times L}$ has rank $p \ll \min(N, L)$.
- Q: Can we leverage that prior into a faster algorithm?

- **Idea**: Break the estimation into two steps:
  1. Compute an approximate basis for the range of X:

$$X \approx QQ^T X \ , \ \text{with } Q \in \mathbb{R}^{N \times n} \ , \ n \ll N, \ Q^T Q = \mathbf{1} \ .$$

  2. Form $B = Q^T X \in \mathbb{R}^{n \times L}$ and compute its SVD: $B = \tilde{U} \Lambda V^T$.
  3. Set $U = Q\tilde{U}$.

- How to solve stage 1?

Interlude: Randomized PCA

- Suppose we suspect that the data matrix $X \in \mathbb{R}^{N \times L}$ has rank $p \ll \min(N, L)$.
- Q: Can we leverage that prior into a faster algorithm?

- **Idea**: Break the estimation into two steps:
  1. Compute an approximate basis for the range of X:

$$X \approx QQ^T X \ , \ \text{with } Q \in \mathbb{R}^{N \times n} \ , \ n \ll N, \ Q^T Q = \mathbf{1} \ .$$

  2. Form $B = Q^T X \in \mathbb{R}^{n \times L}$ and compute its SVD: $B = \tilde{U} \Lambda V^T$.
  3. Set $U = Q\tilde{U}$.

- How to solve stage 1? Randomize!!

Interlude: Randomized PCA

- If we target a rank-k approximation, we simply draw a random matrix $\Omega \in \mathbb{R}^{L \times (k+p)}$, form the matrix $Y = X\Omega \in \mathbb{R}^{N \times (k+p)}$, and perform a SVD of Y: $Y = V \Lambda Q^T$.

Interlude: Randomized PCA

- If we target a rank-k approximation, we simply draw a random matrix $\Omega \in \mathbb{R}^{L\times(k+p)}$, form the matrix $Y = X\Omega \in \mathbb{R}^{N\times(k+p)}$, and perform a SVD of Y: $Y = V\Lambda Q^T$.

- Strong guarantees from concentration of measure:

Theorem: [Halko, Martinsson, Tropp] Given data matrix $X \in \mathbb{R}^{N\times L}$ and $\Omega \in \mathbb{R}^{L\times(k+p)}$ drawn from iid standard Gaussian, the resulting Q satisfies

$$\|X - QQ^T X\| \leq \left(1 + C\sqrt{(k+p)\min(N,L)}\right)\lambda_{k+1} \ ,$$

whp, where λ_{k+1} is the $k+1$-th singular value of X.

Interlude: Randomized PCA

- If we target a rank-k approximation, we simply draw a random matrix $\Omega \in \mathbb{R}^{L\times(k+p)}$, form the matrix $Y = X\Omega \in \mathbb{R}^{N\times(k+p)}$, and perform a SVD of Y: $Y = V\Lambda Q^T$.

- Strong guarantees from concentration of measure:

Theorem: [Halko, Martinsson, Tropp] Given data matrix $X \in \mathbb{R}^{N\times L}$ and $\Omega \in \mathbb{R}^{L\times(k+p)}$ drawn from iid standard Gaussian, the resulting Q satisfies

$$\|X - QQ^T X\| \leq \left(1 + C\sqrt{(k+p)\min(N,L)}\right)\lambda_{k+1} \ ,$$

whp, where λ_{k+1} is the $k+1$-th singular value of X.

- Resulting computational gains:

from $O(NLk)$ to $O(NL\log(k))$ for k ppal components.

Factor Analysis

- We have seen that PCA can be interpreted as a linear latent model:

$$X_i = \sum_j \alpha_{i,j} Y_j + \mu_i \ , \ (i = 1, \ldots L) \ ,$$

$$\text{with } Y_j \text{ uncorrelated, unit variance.}$$

Factor Analysis

- We have seen that PCA can be interpreted as a linear latent model:

$$X_i = \sum_j \alpha_{i,j} Y_j + \mu_i \ , \ (i = 1, \dots L) \ ,$$

with Y_j uncorrelated, unit variance.

- Lack of unicity: given an $L \times L$ orthogonal matrix R, we also have

$$X = \mu + AR^T RY = \mu + \tilde{A}\tilde{Y} \ ,$$

where $\tilde{Y}_j$ are also uncorrelated and unit variance.

Factor Analysis

- We have seen that PCA can be interpreted as a linear latent model:

$$X_i = \sum_j \alpha_{i,j} Y_j + \mu_i \ , \ (i = 1, \dots L) \ ,$$

with Y_j uncorrelated, unit variance.

- Lack of unicity: given an $L \times L$ orthogonal matrix R, we also have

$$X = \mu + AR^T RY = \mu + \tilde{A}\tilde{Y} \ ,$$

where $\tilde{Y}_j$ are also uncorrelated and unit variance.

- Also, an underlying assumption is that data has *low-rank,* i.e. covariance directly reveals dependencies in data.

Factor Analysis

- An alternative to PCA is *Factor Analysis*. Suppose a generative model of the form

$$X_i = \sum_{j \leq J} \alpha_{i,j} Y_j + \mu_i \ + \epsilon_i \ , \ (i = 1, \ldots L) \ ,$$

with $J < L$ and ϵ_i uncorrelated, zero-mean.

Factor Analysis

- An alternative to PCA is *Factor Analysis*. Suppose a generative model of the form

$$X_i = \sum_{j \leq J} \alpha_{i,j} Y_j + \mu_i \ + \epsilon_i \ , \ (i = 1, \dots L) \ ,$$

with $J < L$ and ϵ_i uncorrelated, zero-mean.

- Interpretation:
  - Latent variables Y_j are common factors of variability.
  - Latent variables ϵ_i explain the remaining individual variability, uncorrelated from the rest.
- Example:
  - Factor analysis on the topics of your final project.

Factor Analysis

- Gaussian joint likelihood model:

$$X \sim \mathcal{N}(\mu, AA^T + \mathrm{diag}(\beta))$$

  - with $\beta_i = \mathbf{Var}(\epsilon_i)$.
  - Parameter Estimation? The covariance is a sufficient statistic:

$$\Sigma_X = AA^T + \mathrm{diag}(\beta) \ .$$

$$\uparrow$$

$$\text{low rank}$$

  - SVD is still useful, but does not automatically yield the solution.
  - We will soon see an alternative estimation algorithm.

Independent Component Analysis

- We have seen that asking latent variables only to be decorrelated leads to lack of unicity (any orthogonal transformation is also decorrelated).

Independent Component Analysis

- We have seen that asking latent variables only to be decorrelated leads to lack of unicity (any orthogonal transformation is also decorrelated).
- What happens if we tighten our requirements, e.g. asking that

$$Y_i \text{ and } Y_j \text{ independent}$$

Independent Component Analysis

- We have seen that asking latent variables only to be decorrelated leads to lack of unicity (any orthogonal transformation is also decorrelated).
- What happens if we tighten our requirements, e.g. asking that

$$Y_i \text{ and } Y_j \text{ independent}$$

- We have seen that in the Gaussian case, this does not alleviate the problem. (why?)

Independent Component Analysis

- We have seen that asking latent variables only to be decorrelated leads to lack of unicity (any orthogonal transformation is also decorrelated).
- What happens if we tighten our requirements, e.g. asking that

$$Y_i \text{ and } Y_j \text{ independent}$$

- We have seen that in the Gaussian case, this does not alleviate the problem. (why?)

- But, as it turns out, it is the exception. The model becomes uniquely identifiable if

$$Y_i \text{ and } Y_j \text{ independent and non-Gaussian.}$$

Independent Component Analysis

- Without loss of generality, we can assume X has trivial covariance: $\Sigma_X = \mathbf{1}$.
- Since $\Sigma_Y = \mathbf{1}$ as well, it results that $\mathbf{1} = A\Sigma_Y A^T = AA^T$

Independent Component Analysis

- Without loss of generality, we can assume X has trivial covariance: $\Sigma_X = \mathbf{1}$.
- Since $\Sigma_Y = \mathbf{1}$ as well, it results that $\mathbf{1} = A\Sigma_Y A^T = AA^T$

- so we are reduced to finding an orthogonal transformation such that $Y = A^T X$ becomes independent and non-Gaussian.

Independent Component Analysis

- Without loss of generality, we can assume X has trivial covariance: $\Sigma_X = \mathbf{1}$.
- Since $\Sigma_Y = \mathbf{1}$ as well, it results that $\mathbf{1} = A\Sigma_Y A^T = AA^T$

- so we are reduced to finding an orthogonal transformation such that $Y = A^T X$ becomes independent and non-Gaussian.

- It is a form of "inverse" Central Limit Theorem method.

- Q: How to measure/estimate statistical independence?

Entropy

- For a continuous random variable X with density p(x), the *differential entropy* is

$$H(X) = -\int p(x) \log p(x) dx = -\mathbb{E}(\log p(X)) \ .$$

  – it extends the notion of uncertainty/information carried by X.

Entropy

- For a continuous random variable X with density p(x), the *differential entropy* is

$$H(X) = -\int p(x) \log p(x) dx = -\mathbb{E}(\log p(X)) \ .$$

  – it extends the notion of uncertainty/information carried by X.

- Entropy measures independence through the *mutual information*:

Given $X_1, \ldots, X_n$, the mutual information is

$$I(Y) = \sum_{i \leq n} H(Y_i) - H(Y) \ \geq 0 \ .$$

Entropy

- For a continuous random variable X with density p(x), the *differential entropy* is

$$H(X) = -\int p(x) \log p(x) dx = -\mathbb{E}(\log p(X)) \ .$$

  – it extends the notion of uncertainty/information carried by X.

- Entropy measures independence through the *mutual information*:

Given $X_1, \ldots, X_n$, the mutual information is

$$I(Y) = \sum_{i \leq n} H(Y_i) - H(Y) \ \geq 0 \ .$$

- **Fact:** $I(Y) = 0$ iff Y_i and Y_j are mutually independent.
- **Fact:** If A is unitary and $Y = A^T X$, then $H(Y) = H(X)$.

Independent Component Analysis

- So ICA attempts to solve the following problem:

$$\arg \min_{A^T A = \mathbf{1}} \sum_{i \leq L} H(\langle A_i, X \rangle) - H(X) = \arg \min_{A^T A = \mathbf{1}} \sum_{i \leq L} H(\langle A_i, X \rangle) .$$

- Ex from ESLL:

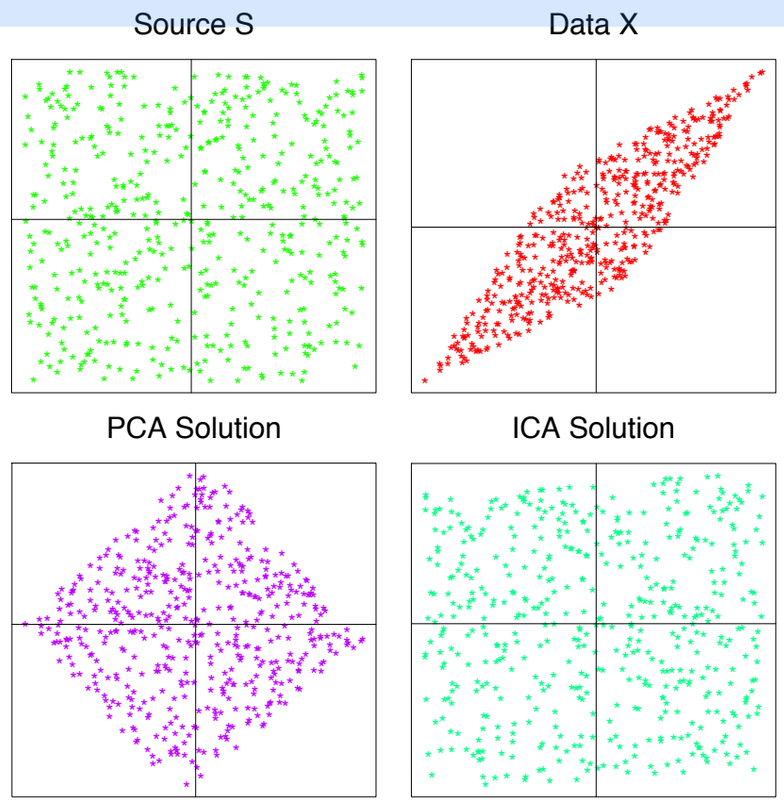


FIGURE 14.38. *Mixtures of independent uniform random variables. The upper left panel shows* 500 *realizations from the two independent uniform sources, the upper right panel their mixed versions. The lower two panels show the PCA and ICA solutions, respectively.*

- Challenge: computing entropy requires estimating the density: exposed to curse of dimensionality!

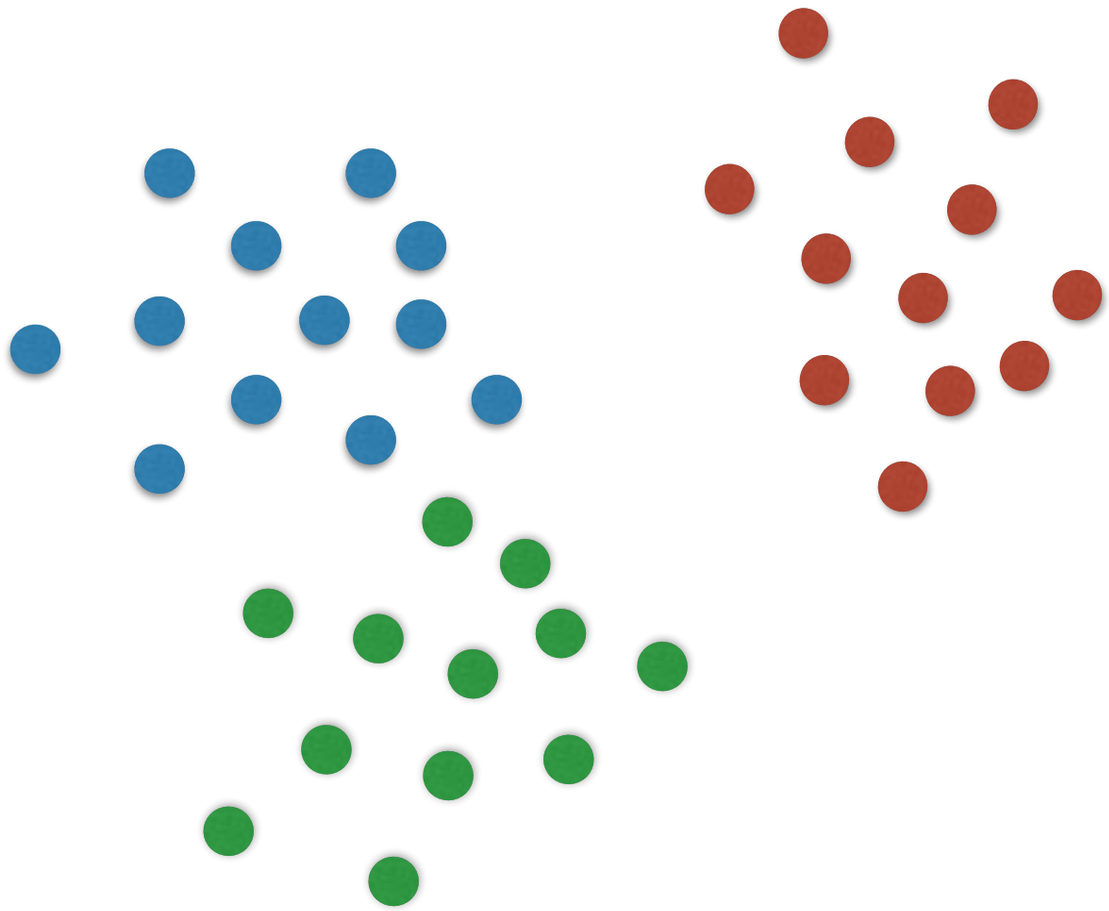
Explaining Away

- The models we have seen so far ultimately construct latent variables by applying a linear transformation over the observed variables.
- Most interesting inferential tasks consider "competing" hypothesis.

– This is known as *explaining away*.

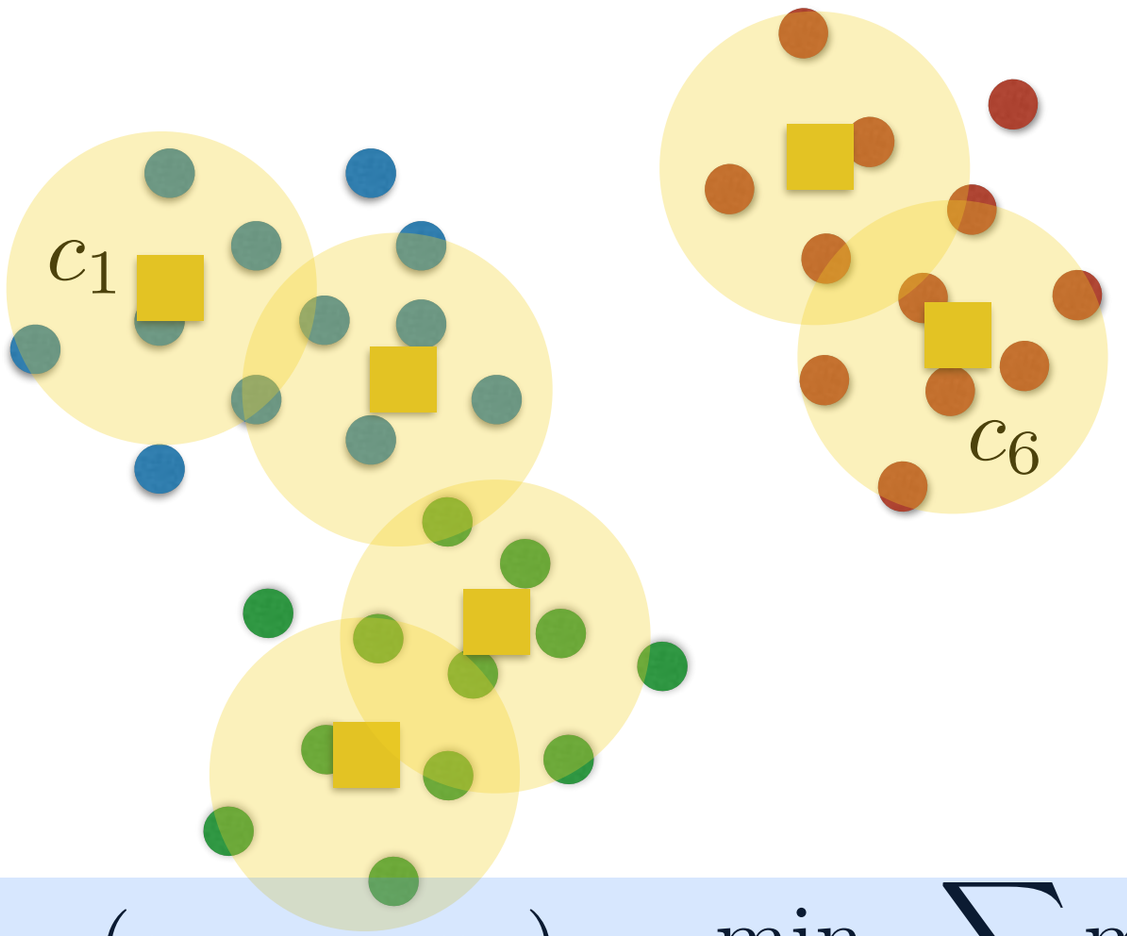
Latent Variables

- The simplest model that captures explaining away is K-means clustering:



Latent Variables

- The simplest model that captures explaining away is K-means clustering:



$K = 6$

Given data $X = (x_1, \ldots, x_n)$, $\min_{c_1,\ldots,c_K} \sum_{i \leq n} \min_j \|x_i - c_j\|^2$

Floyd Algorithm

- For each i, we define r_i a one-hot vector of length K encoding its cluster.
- Cost function is

$$E(c, r) = \sum_i \sum_k r_i(k) \|x_i - c_k\|^2$$

Floyd Algorithm

- For each i, we define r_i a one-hot vector of length K encoding its cluster.

- Cost function is

$$E(c, r) = \sum_i \sum_k r_i(k) \|x_i - c_k\|^2$$

- Fixing c, we optimize r as

$$r_i \leftarrow \arg\min_k \|x_i - c_k\|$$

Given assignments r, optimize E with respect to c:

$$c_k = \frac{\sum_i r_i(k) x_i}{\sum_i r_i(k)}$$

mean of all datapoints falling in cluster k

Floyd Algorithm

- This iterative algorithm converges towards a local optimum (each step decreases the cost).

- It is in fact an instance of the Expectation-Maximization algorithm (EM).

- In that case, the discrete latent variables are the cluster assignments.

Gaussian Mixture Models (GMM)

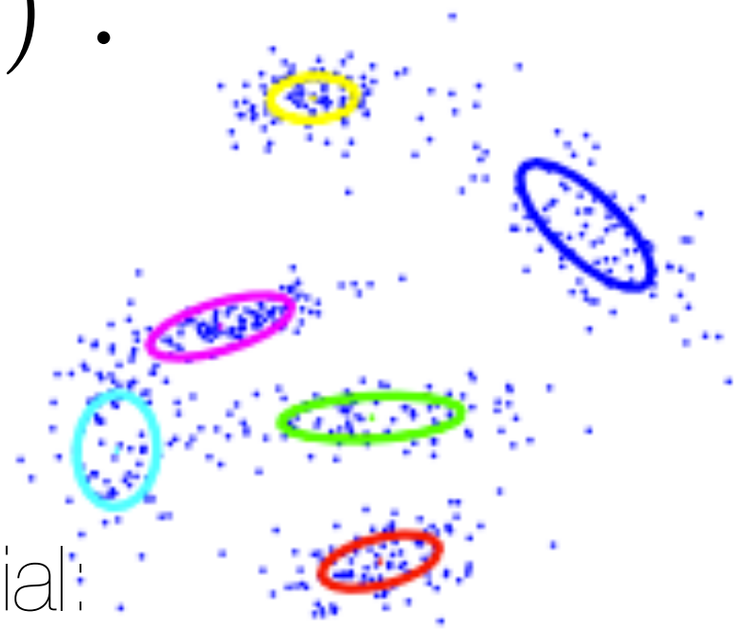
- A generalization of K-Means is given by a Gaussian Mixture:

$$k \sim \mathbf{Mult}(\pi) \ , \ x \sim \mathcal{N}(\mu_k, \Sigma_k) \ .$$

- This is also a discrete latent variable model:

$$z \in \{0,1\}^K \ , \ \sum_k z_k = 1 \ .$$

- The distribution of the latent variable is multinomial:



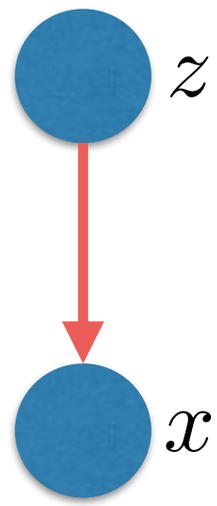
(figure from R.Salakhutdinov)

$$p(z_k = 1) = \pi_k \ , \ 0 \leq \pi_k \leq 1 \ , \ \sum_k \pi_k = 1 \ .$$

Gaussian Mixture Models (GMM)

- We can write

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k} \qquad p(x \mid z_k = 1) = \mathcal{N}(x; \mu_k, \Sigma_k)$$



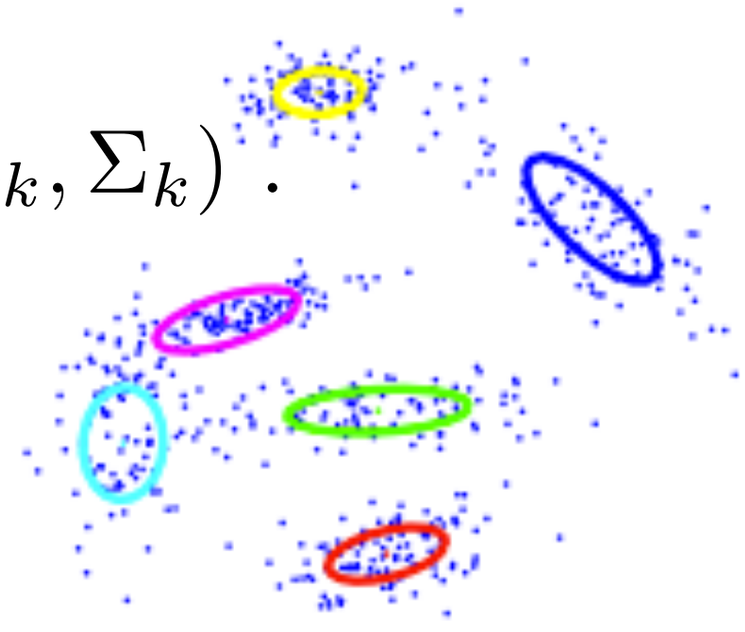
- Thus

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

- Joint and marginal distributions are given by

$$p(x, z) = p(x \mid z)p(z) \ ,$$

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



GMM and Posterior Inference

- What about the conditional p(z | x) ? Ie, given data, which mixture components are "responsible" ?

$$p(z_k = 1 \mid x) = \frac{p(z_k = 1, x)}{\sum_{k' \leq K} p(z_{k'} = 1, x)} = \frac{p(z_k = 1)p(x \mid z_k = 1)}{\sum_{k' \leq K} p(z_{k'} = 1)p(x \mid z_{k'} = 1)}$$

$$= \frac{\pi_k \mathcal{N}(x; \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(x; \mu_{k'}, \Sigma_{k'})}$$

- The posterior probability that $z_k = 1$ is a weighted average of prior probabilities that depends upon the data.
- Q: How to estimate the parameters $\{\pi, \mu, \Sigma\}$?

Maximum Likelihood Estimation

- Given independent samples $X = \{x_1, \ldots, x_n\}$, the total log-likelihood is

$$E(\pi, \mu, \Sigma) = \log p(X \mid \pi, \mu, \Sigma) = \sum_{i \leq n} \log \left(\sum_k \pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k) \right)$$

-

$$\frac{\partial E}{\partial \mu_k} = \sum_i \frac{\pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(x_i; \mu_{k'}, \Sigma_{k'})} \Sigma_k^{-1}(x_i - \mu_k) \ .$$

$$\mu_k = \frac{1}{N_k} \sum_i p(z_{i,k} = 1 \mid x_i) x_i \ , \ N_k = \sum_i p(z_{i,k} = 1 \mid x_i) \ .$$

Thus the mean μ_k is the weighted average of datapoints, with weights given by the posterior probabilities of belonging to component k.

Maximum Likelihood Estimation

- Similarly

$$\frac{\partial E}{\partial \Sigma_k} = 0 \;\Rightarrow\; \Sigma_k = \frac{1}{N_k} \sum_i p(z_{i,k} = 1 \mid x_i)(x_i - \mu_k)(x_i - \mu_k)^T \;.$$

$$\frac{\partial E}{\partial \pi_k} = 0 \;\Rightarrow\; \pi_k = \frac{N_k}{n} \;.$$

- MLE parameters do not have closed-form solution
  - Parameters depend upon posterior probabilities $p(z_k = 1 \mid x)$, which themselves depend upon parameters.
- Iterative algorithm: Expectation-Maximization (EM):
  - E-step: Update posterior probabilities with parameters fixed.
  - M-step: Update parameters with posterior probabilities fixed.

The EM algorithm

- It is designed to find MLE solutions of latent variable models.
- In general, we have log-likelihoods of the form

$$\log p(X \mid \theta) = \log \left(\sum_Z p(X, Z \mid \theta) \right) \ , \ \theta = \text{model parameters} \ .$$
$$Z = \text{latent variables}$$

- Using current parameters θ_{old}, we compute the expected total likelihood of the model (E-step):

$$\mathcal{Q}(\theta, \theta_{old}) = \mathbb{E}_{Z \sim p(Z \mid X, \theta_{old})} \log p(X, Z \mid \theta)$$

- Then we update the parameters to maximize this likelihood:

$$\theta_{new} = \arg\max_\theta Q(\theta, \theta_{old}) \ .$$

The EM algorithm

- It is designed to find MLE solutions of latent variable models.
- In general, we have log-likelihoods of the form

$$\log p(X \mid \theta) = \log \left(\sum_{Z} p(X, Z \mid \theta) \right) \quad , \quad \begin{array}{l} \theta = \text{model parameters} \; . \\ Z = \text{latent variables} \end{array}$$

The EM algorithm

- It is designed to find MLE solutions of latent variable models.
- In general, we have log-likelihoods of the form

$$\log p(X \mid \theta) = \log \left(\sum_Z p(X, Z \mid \theta) \right) , \ \theta = \text{model parameters} \ . \quad Z = \text{latent variables}$$

- Using current parameters θ_{old}, we compute the expected total likelihood of the model (E-step):

$$\mathcal{Q}(\theta, \theta_{old}) = \mathbb{E}_{Z \sim p(Z \mid X, \theta_{old})} \log p(X, Z \mid \theta)$$

- Then we update the parameters to maximize this likelihood:

$$\theta_{new} = \arg\max_{\theta} Q(\theta, \theta_{old}) \ .$$

EM and Variational Bound

- Q: Does this algorithm monotonically improve the likelihood?
- Assume for now that latent variables are discrete.
- For any distribution $q(Z)$over latent variables, we have

$$\log p(X \mid \theta) = \log \left(\sum_Z p(X, Z \mid \theta) \right) = \log \left(\sum_Z q(Z) \frac{p(X, Z \mid \theta)}{q(Z)} \right)$$

$$\geq \sum_Z q(Z) \log \left(\frac{p(X, Z \mid \theta)}{q(Z)} \right) = \mathcal{L}(q, \theta) \ .$$

(Jensen's Inequality: $\mathbb{E}(f(X)) \geq f(\mathbb{E}(X))$ if f is convex)

Variational Bound

- We can express the variational lower bound as

$$\begin{aligned}\mathcal{L}(q,\theta) &= \mathbb{E}_{q(Z)}\left[\log p(X,Z \mid \theta)\right] - \mathbb{E}_{q(Z)} \log q(Z) \\ &= \mathbb{E}_{q(Z)}\left[\log p(X,Z \mid \theta)\right] + H(q) \ .\end{aligned}$$

$$H(q)\text{: Entropy of } q(Z).$$

- Also, we have

$$\log p(X \mid \theta) = \mathcal{L}(q,\theta) + KL(q(z)||p(z \mid x,\theta)) \ , \text{ where}$$

$$KL(q||p) = -\sum_z q(z) \log\left(\frac{p(z)}{q(z)}\right)$$

is the Kullback-Leibler divergence.

Variational Bound

- Thus, the divergence $KL(q||p)$ measures how far our variational approximation $q(z)$ is from the true posterior, and directly controls the bound on the log-likelihood.

- Using

$$\log p(X \mid \theta) = \mathcal{L}(q, \theta) + KL(q(z)||p(z \mid x, \theta))$$

- E-step: maximize lower bound $\mathcal{L}(q, \theta)$ with respect to *q*, holding parameters fixed.
- M-step: maximize lower bound $\mathcal{L}(q, \theta)$ with respect to parameters, holding *q* fixed.

Exponential Families

- Suppose we have iid data $x_1, \dots x_n$ and we consider a collection of *sufficient statistics* $\{\phi_k(X)\}_k$
- The empirical expectations of these statistics are

$$\hat{\mu}_k = \frac{1}{n} \sum_i \phi_k(x_i)$$

- Q: Can we build a distribution $p(x)$ consistent with these empirical moments? i.e.

$$\mathbb{E}_{X \sim p(x)}\{\phi_k(X)\} = \hat{\mu}_k \quad \text{for all } k.$$

- In general, this is an underdetermined problem. How to choose wisely amongst all possible solutions?

Exponential Families and Maximum Entropy

- A reasonable choice is to consider the distribution with *maximum entropy* subject to the empirical moments:

$$p^* = \arg\max_p H(p) \ , \ s.t. \ \mathbb{E}_p\{\phi_k(X)\} = \hat{\mu}_k \text{ for all } k.$$

$$\text{Shannon Entropy: } H(p) = -\mathbb{E}\{\log(p)\} \ .$$

- The general form of maximum entropy is

$$p(x) \propto \exp\left\{\sum_k \lambda_k \phi_k(x)\right\}$$

λ_k: Lagrange multipliers adjusted such that $\mathbb{E}_p \phi_k(X) = \hat{\mu}_k$ for all k.

Exponential Families

- The exponential family associated with ϕ is defined as the parametric family

$$p_\theta(x) = \exp\{\langle \theta, \phi(x) \rangle - A(\theta)\} \ , \ \text{with}$$

$$A(\theta) = \log \int \exp\{\langle \theta, \phi(x) \} dx \qquad \text{log-partition function}$$

- It is well defined for the family of parameters

$$\Omega = \{\theta \ ; \ A(\theta) < \infty\}$$

Exponential Families

- Several well-known models belong to the exponential family
  - Energy based models
  - Gaussian Mixtures
  - Latent Dirichlet Allocation
  - etc.

Exponential Families

- **Proposition:** The log-partition function $A(\theta)$ satisfies

  - $$\frac{\partial A}{\partial \theta_k}(\theta) = \mathbb{E}_\theta\{\phi_k(X)\} = \int \phi_k(x) p_\theta(x) dx \ .$$

  - $A(\theta)$ is convex in its domain Ω.

- Higher order derivatives always exist.

Conjugate Duality

- Conjugate duality representation of convex functions:

$$A^*(\mu) = \sup_{\theta \in \Omega} \{\langle \mu, \theta \rangle - A(\theta)\}$$

$$\text{canonical parameters} \quad \longleftrightarrow \quad \text{moment parameters}$$

$$\theta_k \qquad\qquad\qquad\qquad \mu_k$$

- Q: How to interpret the dual conjugate?

$A^*(\mu)$: Negative entropy of $p_{\theta(\mu)}$, where

$p_{\theta(\mu)}$ is the exponential family distribution such that

$\mathbb{E}_{\theta(\mu)}\phi(X) = \mu$.

- Variational representation:

$$A(\theta) = \sup_{\mu} \{\langle \theta, \mu \rangle - A^*(\mu)\}$$

Variational Inference and Duality

- We derive the exact EM algorithm for exponential families with latent variables. Given observed variables X and latent variables Z, we consider

$$p_\theta(x, z) = \exp\left\{\langle\theta, \phi(x, z)\rangle - A(\theta)\right\} \ , \ \text{with}$$

$$A(\theta) = \log \int_{x,z} \exp\{\langle\theta, \phi(x, z)\rangle\} dx dz$$

- Given observation $X = x$, the posterior distribution is

$$p(z \mid x) = \frac{\exp\{\langle\theta, \phi(x, z)\rangle\}}{\int \exp\{\langle\theta, \phi(x, z')\rangle\} dz'} = \exp\{\langle\theta\phi(x, z)\rangle - A_x(\theta)\}$$

$$A_x(\theta) = \log \int_z \exp\{\langle\theta, \phi(x, z)\rangle\} dz$$

Variational Inference and Conjugate Duality

- The MLE for our parameters θ is obtained by maximizing the incomplete log-likelihood of the data:

$$\mathcal{L}(\theta, x) = \log \int_z \exp\{\langle \theta, \phi(x, z) \rangle - A(\theta)\} dz = A_x(\theta) - A(\theta) \ .$$

- The variational representation gives

$$A_x(\theta) = \sup_{\mu_x} \{\langle \theta, \mu_x \rangle - A_x^*(\mu_x)\}$$

$$A_x^*(\mu_x) = \sup_{\theta} \{\langle \theta, \mu_x \rangle - A_x(\theta)\}$$

- It results in the lower-bound for the incomplete log-likelihood:

$$\mathcal{L}(\theta, x) \geq \langle \mu_x, \theta \rangle - A_x^*(\mu_x) - A(\theta) = \widetilde{\mathcal{L}}(\mu_x, \theta)$$

- EM is thus a coordinate ascent on the lower bound:

$$\mu_x^{(t+1)} = \arg\max_{\mu_x} \widetilde{\mathcal{L}}(\mu_x, \theta^{(t)}) \qquad \text{(E step)}$$

$$\theta^{(t+1)} = \arg\max_{\theta} \widetilde{\mathcal{L}}(\mu_x^{(t+1)}, \theta) \qquad \text{(M step)}$$

- E step is called expectation because the maximizer of $\widetilde{\mathcal{L}}(\mu_x, \theta)$ is, by duality, the expectation $\mu_x^{(t+1)} = \mathbb{E}_{\theta^{(t)}} \phi(x, Z)$

- Also, because $\max_{\mu}\{\langle \mu_x, \theta^{(t)} \rangle - A_x^*(\mu_x)\} = A_x(\theta^{(t)})$, after each E step the inequality becomes an equality, thus M step increases log-likelihood.

Approximate Posterior Inference

- For most models, the posterior is analytically intractable:

$$p(z \mid x) = \frac{p(x \mid z)p(z)}{\int p(x \mid z')p(z')dz'}$$

- ***Variational Bayesian Inference:*** consider a parametric family of approximations $q(z \mid \beta)$ and optimize variational lower bound with respect to the variational parameters β

Mean Field Variational Bayes

- Joint likelihood of observed and latent variables:

$$p(X, Z \mid \theta) \qquad \theta\text{: generative model parameters}$$

- Let us consider a posterior approximation $q(z|\beta)$of the form

$$q(z \mid \beta) = \prod_i q_i(z_i \mid \beta_i) \qquad \beta\text{: Variational parameters}$$

  - Mean-field approximation: we model hidden variables as being independent.

- Corresponding lower-bound is given by

$$\log p(X \mid \theta) \geq \int q(z \mid \beta) \log \frac{p(x, z \mid \theta)}{q(z \mid \beta)} dz = \mathbb{E}_{q(z|\beta)}\{\log(p(X, Z \mid \theta)\} + H(q(z \mid \beta))$$

Mean Field Variational Bayes

- Goal: optimize lower-bound with respect to variational parameters.
- As we have seen, this is equivalent to minimizing the divergence between true and approximate posterior:

$$\log p(X \mid \theta) = \widetilde{\mathcal{L}}(\theta, \beta) + D_{KL}(q_\beta(z) || p(z|x, \theta))$$

- If $q(z \mid \beta)$ is a factorial distribution, the entropy term is tractable:

- Problematic term:

$$H(q(z|\beta)) = \sum_i H(q_i(z_i|\beta_i))$$

$$\nabla_\beta \mathbb{E}_{q(z|\beta)} \log p(X, Z|\theta)$$

Mean Field Variational Bayes

[Paiskey, Blei, Jordan,'12]

- Denote $f(Z) = \log p(X, Z|\theta)$
- Then

$$\nabla_\beta \mathbb{E}_{q(z|\beta)} f(Z) = \nabla_\beta \int f(z) q(z|\beta) dz$$

$$= \int f(z) \nabla_\beta q(z|\beta) dz$$

$$= \int f(z) q(z|\beta) \nabla_\beta \log q(z|\beta) dz$$

$$= \mathbb{E}_q \{ f(Z) \nabla_\beta \log q(z|\beta) \}$$

- Stochastic approximation of $\nabla_\beta \mathbb{E}_{q(z|\beta)} f(Z)$:

$$\nabla_\beta \mathbb{E}_{q(z|\beta)} f(Z) \approx \frac{1}{S} \sum_{s \leq S, z^{(s)} \sim q(z|\beta)} f(z^{(s)}) \nabla_\beta \log q(z^{(s)}|\beta)$$

Mean Field Variational Bayes

- The estimator of the gradient is unbiased, but it may suffer from large variance.
  - We may need a large number S of samples to stabilize the descent.

- Faster alternative?