Dimensionality Reduction (continued)
Lecture 25

David Sontag
New York University

Slides adapted from Carlos Guestrin and Luke Zettlemoyer

Basic PCA algorithm

- Start from m by n data matrix **X**
- **Recenter**: subtract mean from each row of **X**
  - $\mathbf{X}_c \leftarrow \mathbf{X} - \overline{\mathbf{X}}$
- **Compute covariance** matrix:
  - $\Sigma \leftarrow 1/m\ \mathbf{X}_c^T \mathbf{X}_c$
- Find **eigen vectors and values** of Σ
- **Principal components:** k eigen vectors with highest eigen values

Linear projections, a review

$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

- Project a point into a (lower dimensional) space:
  - **point**: $\mathbf{x} = (x_1,\ldots,x_n)$
  - **select a basis** – set of unit (length 1) basis vectors $(\mathbf{u}_1,\ldots,\mathbf{u}_k)$
    - we consider orthonormal basis:
      - $\mathbf{u}_i \bullet \mathbf{u}_i = 1$, and $\mathbf{u}_i \bullet \mathbf{u}_j = 0$ for $i \neq j$
  - **select a center** – $\bar{\mathbf{x}}$, defines offset of space
  - **best coordinates** in lower dimensional space defined by dot-products: $(z_1,\ldots,z_k)$, $z_i = (\mathbf{x}-\bar{\mathbf{x}}) \bullet \mathbf{u}_i$

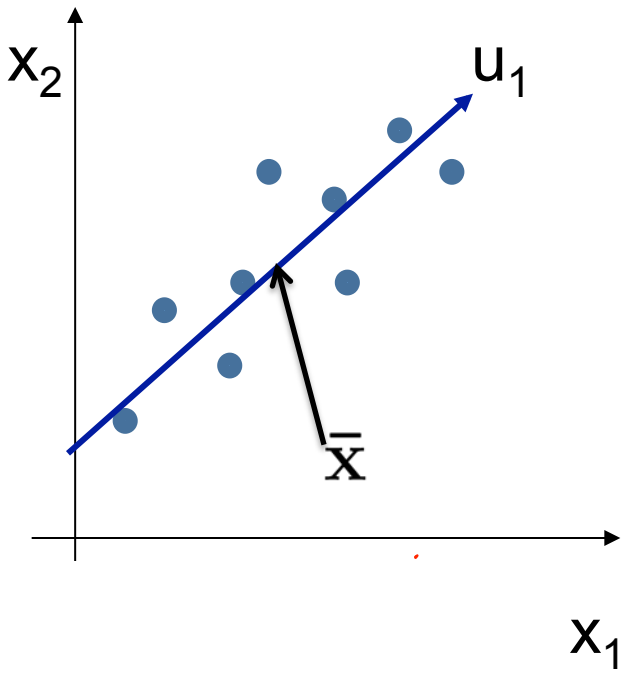
PCA finds projection that minimizes reconstruction error

- Given m data points: $\mathbf{x}^i = (x_1^i,\ldots,x_n^i)$, i=1…m
- Will represent each point as a projection:

$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}^i$$

$$z_j^i = (\mathbf{x}^i - \bar{\mathbf{x}}) \cdot \mathbf{u}_j$$



- PCA:
  - Given k<n, find $(\mathbf{u}_1,\ldots,\mathbf{u}_k)$ minimizing reconstruction error:

$$error_k = \sum_{i=1}^{m} (\mathbf{x}^i - \hat{\mathbf{x}}^i)^2$$

Understanding the reconstruction error

- Note that $\mathbf{x}^i$ can be represented exactly by n-dimensional projection:

$$\mathbf{x}^i = \bar{\mathbf{x}} + \sum_{j=1}^{n} z_j^i \mathbf{u}_j$$

$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

$$z_j^i = (\mathbf{x}^i - \bar{\mathbf{x}}) \cdot \mathbf{u}_j$$

- Given k<n, find ($\mathbf{u}_1,\ldots,\mathbf{u}_k$) minimizing reconstruction error:

$$error_k = \sum_{i=1}^{m} (\mathbf{x}^i - \hat{\mathbf{x}}^i)^2$$

- Rewriting error:

$$error_k = \sum_{i=1}^{m} \left(x^i - \left[\bar{x} + \sum_{j=1}^{k} z_j^i u_j \right] \right)^2 = \sum_{i=1}^{m} \left(\left[\bar{x} + \sum_{j=1}^{n} z_j^i u_j \right] - \left[\bar{x} + \sum_{j=1}^{k} z_j^i u_j \right] \right)^2$$

$$= \sum_{i=1}^{m} \left(\sum_{j=k+1}^{n} z_j^i u_j \right)^2 = \sum_{i=1}^{m} \sum_{j=k+1}^{n} z_j^i u_j \cdot u_j z_j^i + \sum_{i=1}^{m} \sum_{j=k+1}^{n} \sum_{l=k+1, l \neq k}^{n} z_j^i u_j \cdot u_l z_l^i$$

$$= \sum_{i=1}^{m} \sum_{j=k+1}^{n} (z_j^i)^2$$

Error is the sum of squared weights for dimensions that have been cut!

Reconstruction error and covariance matrix

$$error_k = \sum_{i=1}^{m} \sum_{j=k+1}^{n} [\mathbf{u}_j \cdot (\mathbf{x}^i - \bar{\mathbf{x}})]^2$$

$$= \sum_{i=1}^{m} \sum_{j=k+1}^{n} u_j^T (x^i - \bar{x})(x^i - \bar{x})^T u_j$$

$$= \sum_{j=k+1}^{n} u_j^T \left[\sum_{i=1}^{m} (x^i - \bar{x})(x^i - \bar{x})^T \right] u_j$$

$$error_k = \sum_{j=k+1}^{n} u_j^T \Sigma u_j$$

$$\Sigma = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{x}^i - \bar{\mathbf{x}})(\mathbf{x}^i - \bar{\mathbf{x}})^T$$

Thus, to minimize the reconstruction error we want to **minimize**

$$\sum_{j=k+1}^{n} u_j^T \Sigma u_j$$

Recall that to maximize the variance we want to **maximize**

$$\sum_{j=1}^{k} u_j^T \Sigma u_j$$

These are **equivalent**!

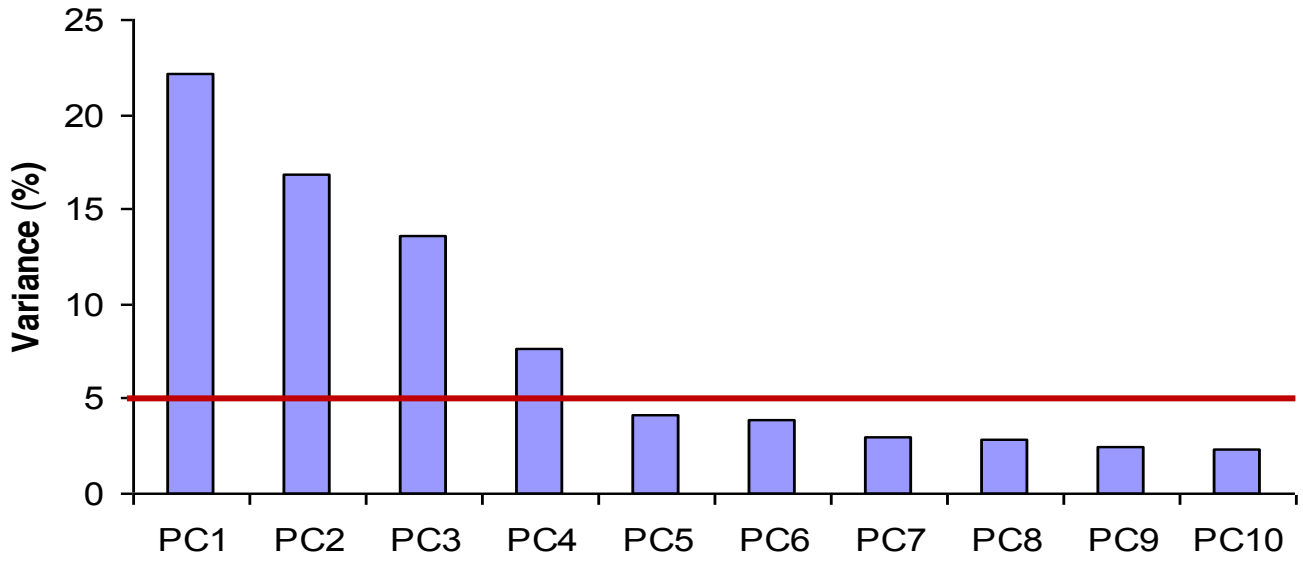
$$\sum_{j=1}^{k} u_j^T \Sigma u_j + \sum_{j=k+1}^{n} u_j^T \Sigma u_j = \sum_{j=1}^{n} u_j^T \Sigma u_j = \text{trace}(\Sigma)$$

Dimensionality reduction with PCA

In high-dimensional problem, data usually lies near a linear subspace, as noise introduces small variability

Only keep data projections onto principal components with **large** eigenvalues

Can *ignore* the components of lesser significance.



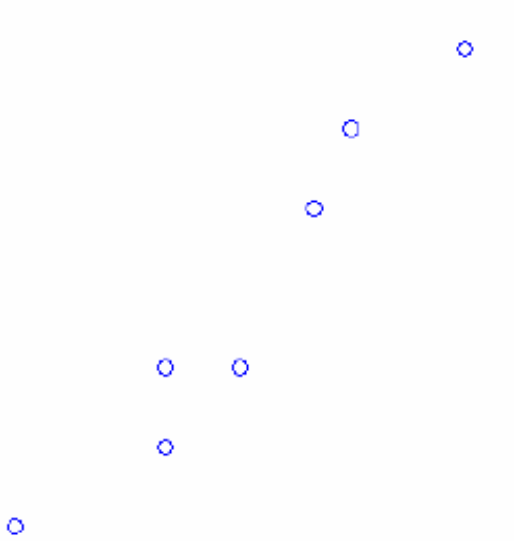
You might lose some information, but if the eigenvalues are small, you don't lose much

[Slide from Aarti Singh]

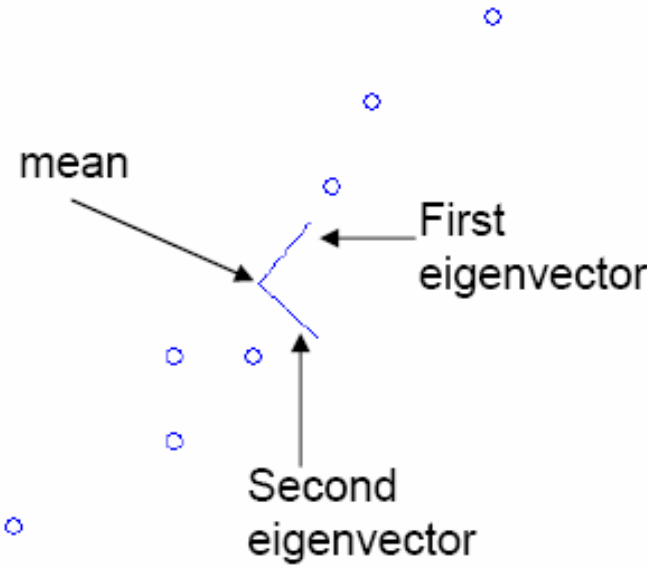
PCA example

$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

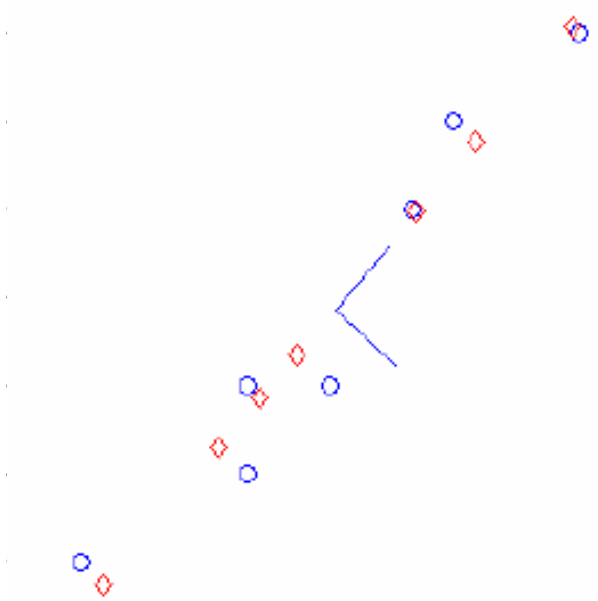
Data:



Projection:

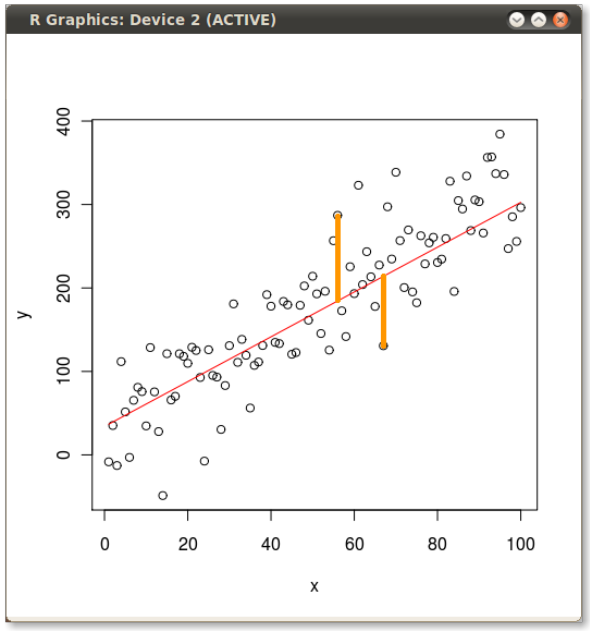


Reconstruction:

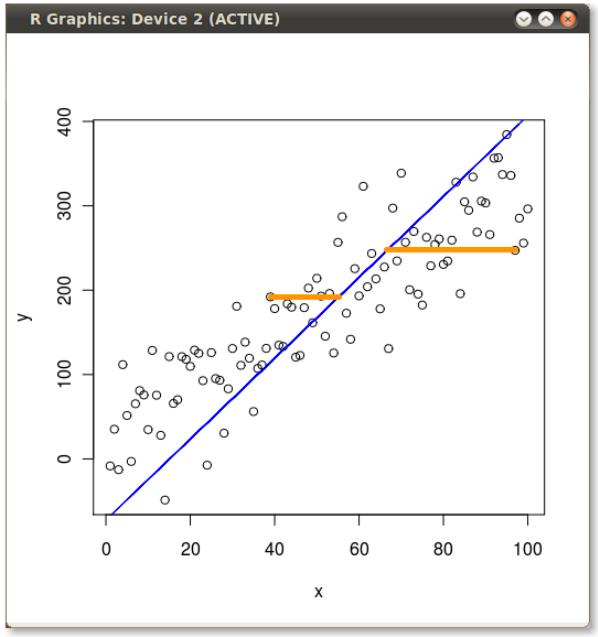


What's the difference between the first eigenvector and linear regression?

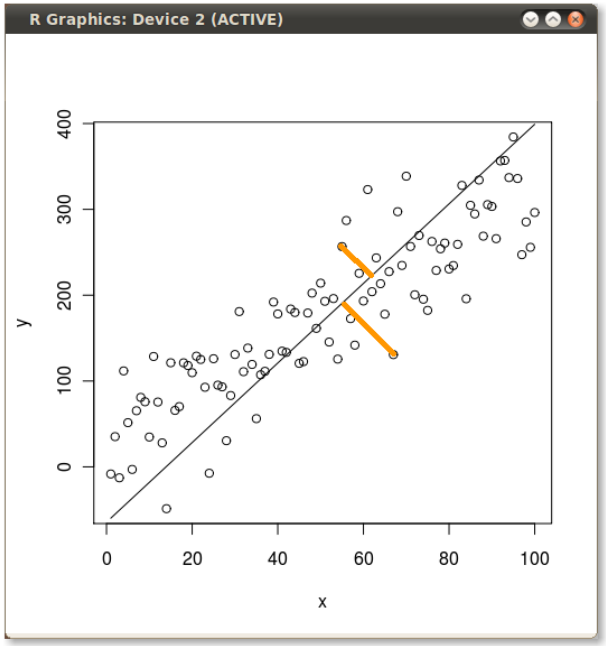
Suppose we have data { (*x*,*y*) }



Predict *y* from *x*



Predict *x* from *y*



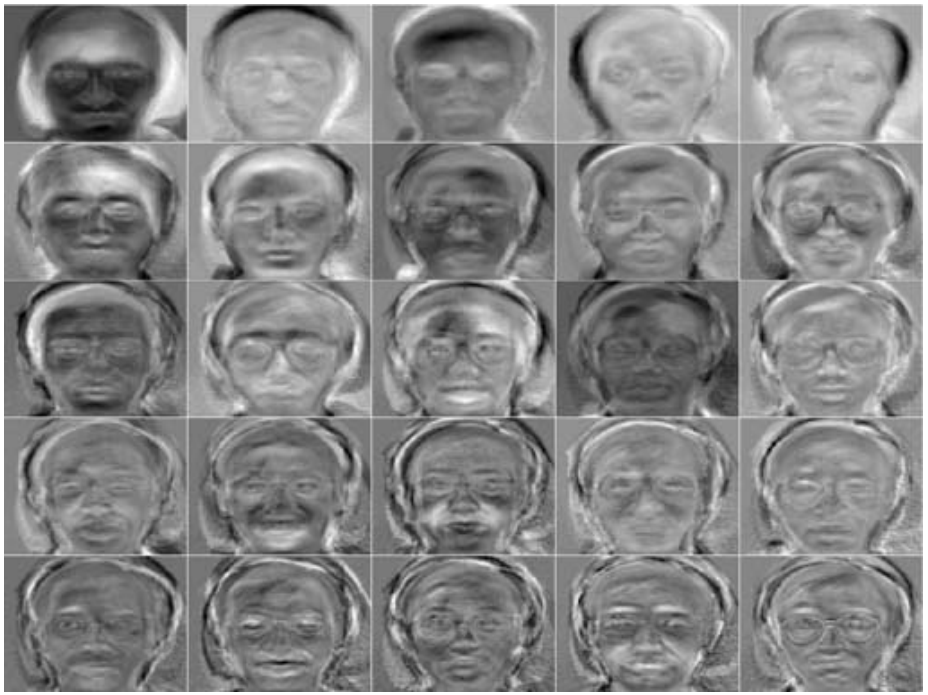
PCA

[Pictures from "Cerebral Mastication" blog]

Eigenfaces [Turk, Pentland ’91]

- Input images:

- Principal components:



Eigenfaces reconstruction

- Each image corresponds to adding together the principal components:

Scaling up

- Covariance matrix can be really big!
  - Σ is n by n
  - 10000 features can be common!
  - finding eigenvectors is very slow…
- Use singular value decomposition (SVD)
  - finds to k eigenvectors
  - great implementations available, e.g., Matlab svd

SVD

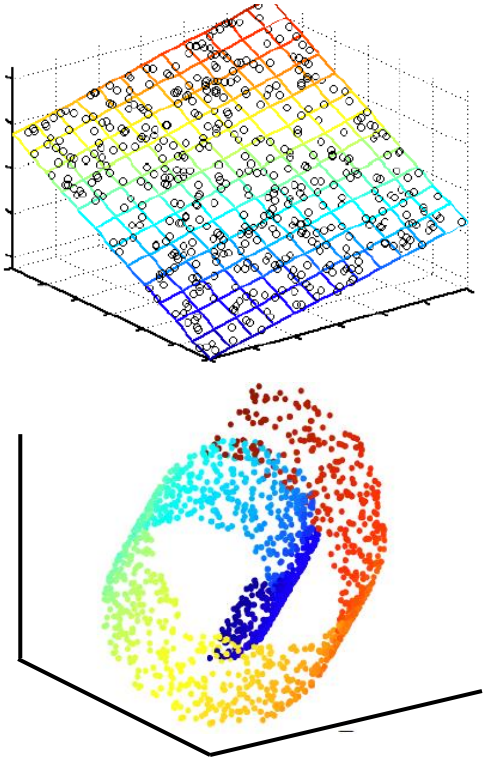
- Write $\mathbf{X} = \mathbf{W}\,\mathbf{S}\,\mathbf{V}^T$
  - $\mathbf{X} \leftarrow$ data matrix, one row per datapoint
  - $\mathbf{W} \leftarrow$ weight matrix, one row per datapoint – coordinate of $\mathbf{x}^i$ in eigenspace
  - $\mathbf{S} \leftarrow$ singular value matrix, diagonal matrix
    - in our setting each entry is eigenvalue λ_j
  - $\mathbf{V}^T \leftarrow$ singular vector matrix
    - in our setting each row is eigenvector $\mathbf{v}_j$

PCA using SVD algorithm

- Start from m by n data matrix **X**
- **Recenter**: subtract mean from each row of **X**
  - $\mathbf{X}_c \leftarrow \mathbf{X} - \overline{\mathbf{X}}$
- **Call SVD** algorithm on $\mathbf{X}_c$ – ask for k singular vectors
- **Principal components:** k singular vectors with highest singular values (rows of $\mathbf{V}^T$)
  - **Coefficients:** project each point onto the new vectors

Non-linear methods

- Linear
  - **Principal Component Analysis (PCA)**
  - Factor Analysis
  - Independent Component Analysis (ICA)
- Nonlinear
  - **Laplacian Eigenmaps**
  - ISOMAP
  - Local Linear Embedding (LLE)
  - Latent Dirichlet allocation



"swiss roll"

Slide from Aarti Singh

Probabilistic topic models

Topics | *Documents* | *Topic proportions and assignments*

β_1

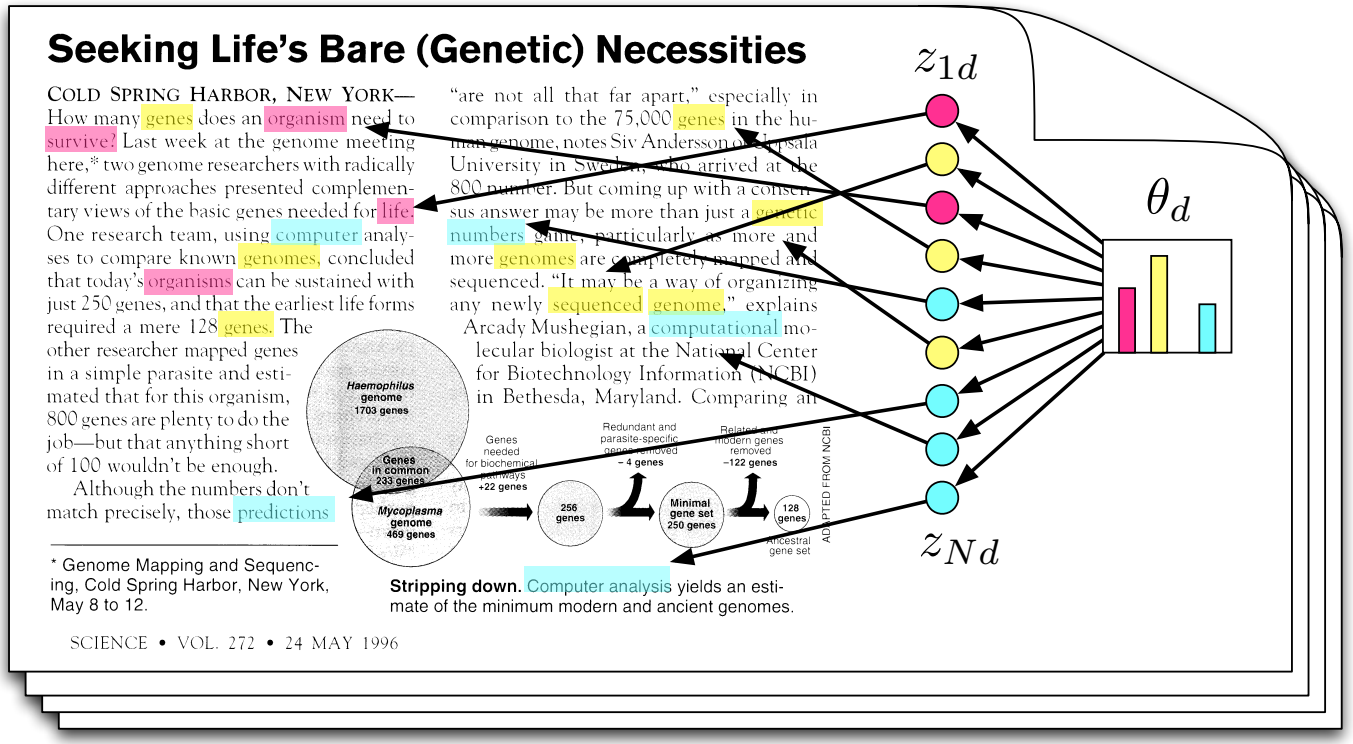
gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

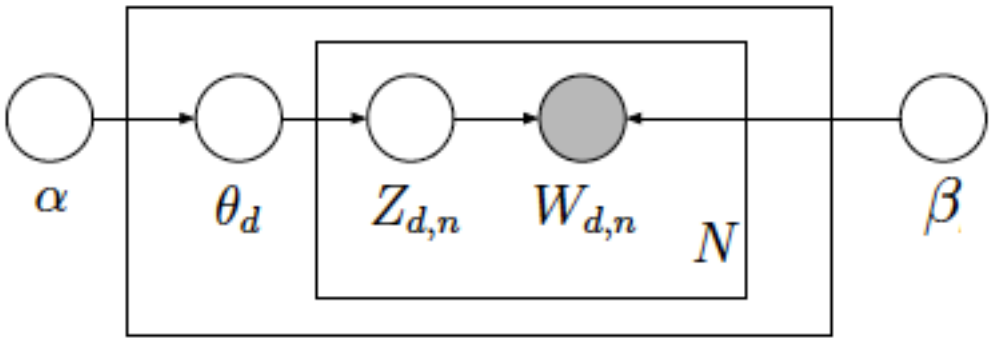
β_T

data	0.02
number	0.02
computer	0.01
...	



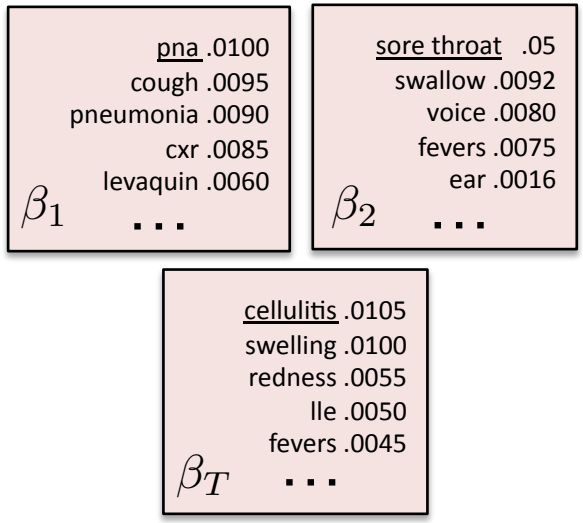
(Blei, Ng, Jordan JMLR '03)

Probabilistic topic models



Graphical model for Latent Dirichlet Allocation (LDA)

Topic word distributions



Triage note

Inference

	θ_d
Pneumonia	0.50
Common cold	0.49
Diabetes	0.01

Low Dimensional representation:
distribution of topics for the note

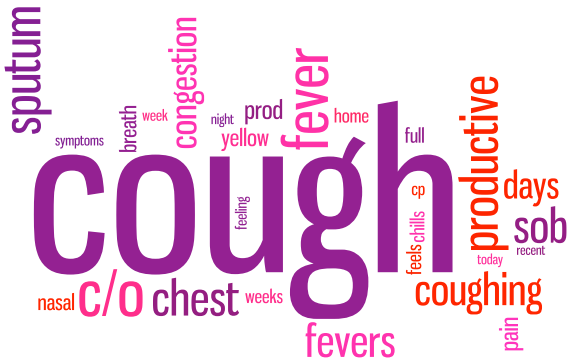
(Blei, Ng, Jordan JMLR '03)

Example of learned representation

Paraphrased note:

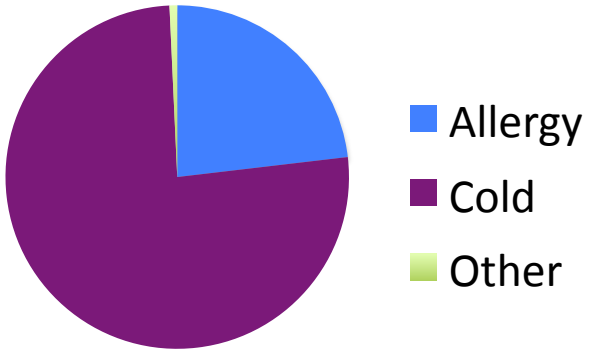
*"Patient has **URI [upper respiratory infection] symptoms** like **cough, runny nose, ear pain**. **Denies fevers**. **history of seasonal allergies**"*

Cold/URI



Allergy

Inferred Topic Distribution



What you need to know

- Dimensionality reduction
  - why and when it's important
- Simple feature selection
- Regularization as a type of feature selection
- Principal component analysis
  - minimizing reconstruction error
  - relationship to covariance matrix and eigenvectors
  - using SVD