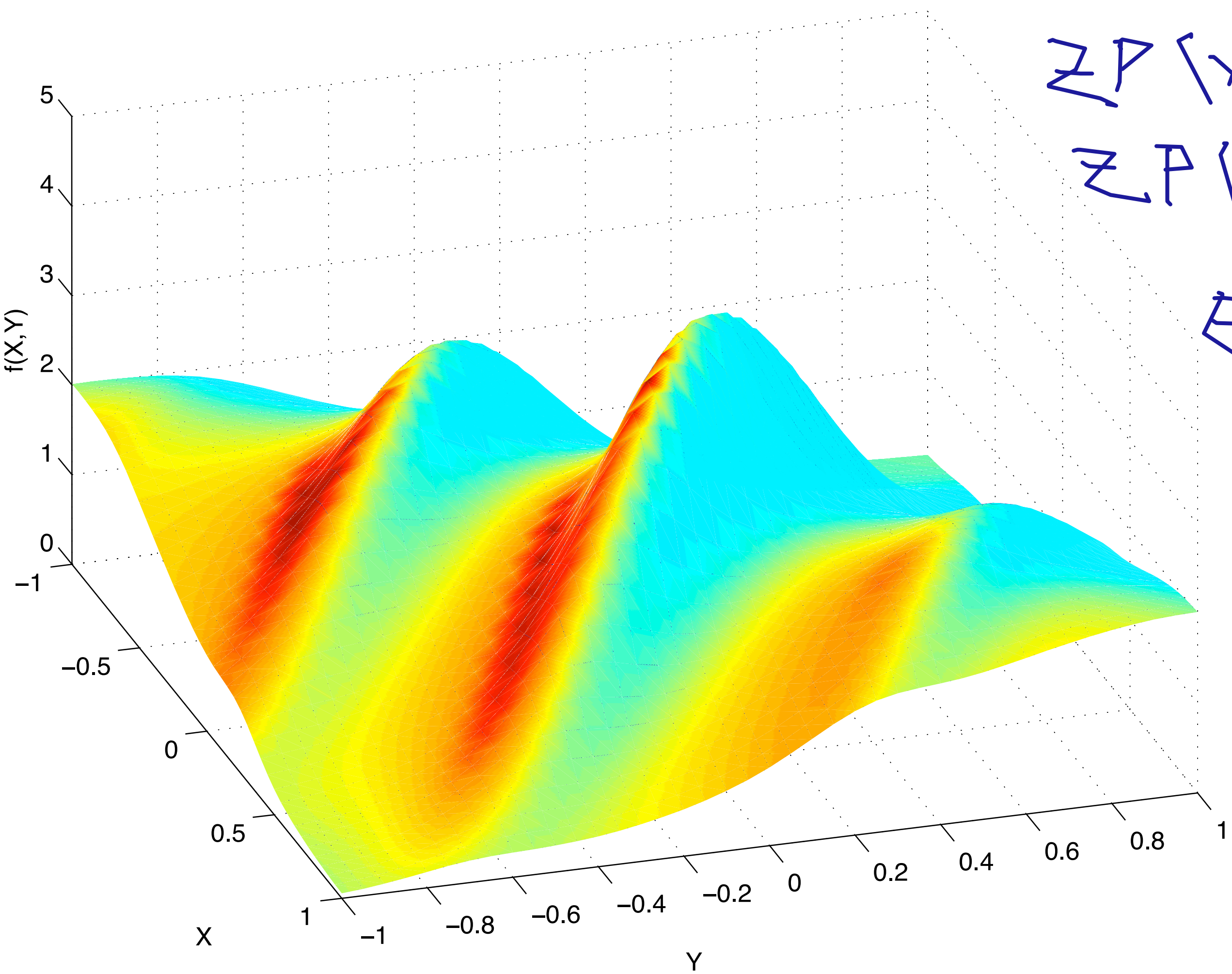
Review

- Parallel importance sampling
  - bias due to 1/normalizer
  - particle filter = recursive parallel IS
- MCMC
  - randomized search for high P(x)
  - burn-in, mixing
  - approx. iid: $\{ X_t, X_{t+\Delta}, X_{t+2\Delta}, X_{t+3\Delta}, \ldots \}$
  - use to construct estimator of $E_P(g(X))$

Review

- Metropolis-Hastings
  - way to design chain w/ stationary dist'n P(X)
  - proposal distribution Q(X' | X)
    - e.g., random walk N(X' | X, σ^2I)
  - accept w.p. min(1, $\frac{P(x')}{P(x_t)} \frac{Q(x_t | x')}{Q(x' | x_t)}$)
  - tension btwn long moves, high accept rate

MH example

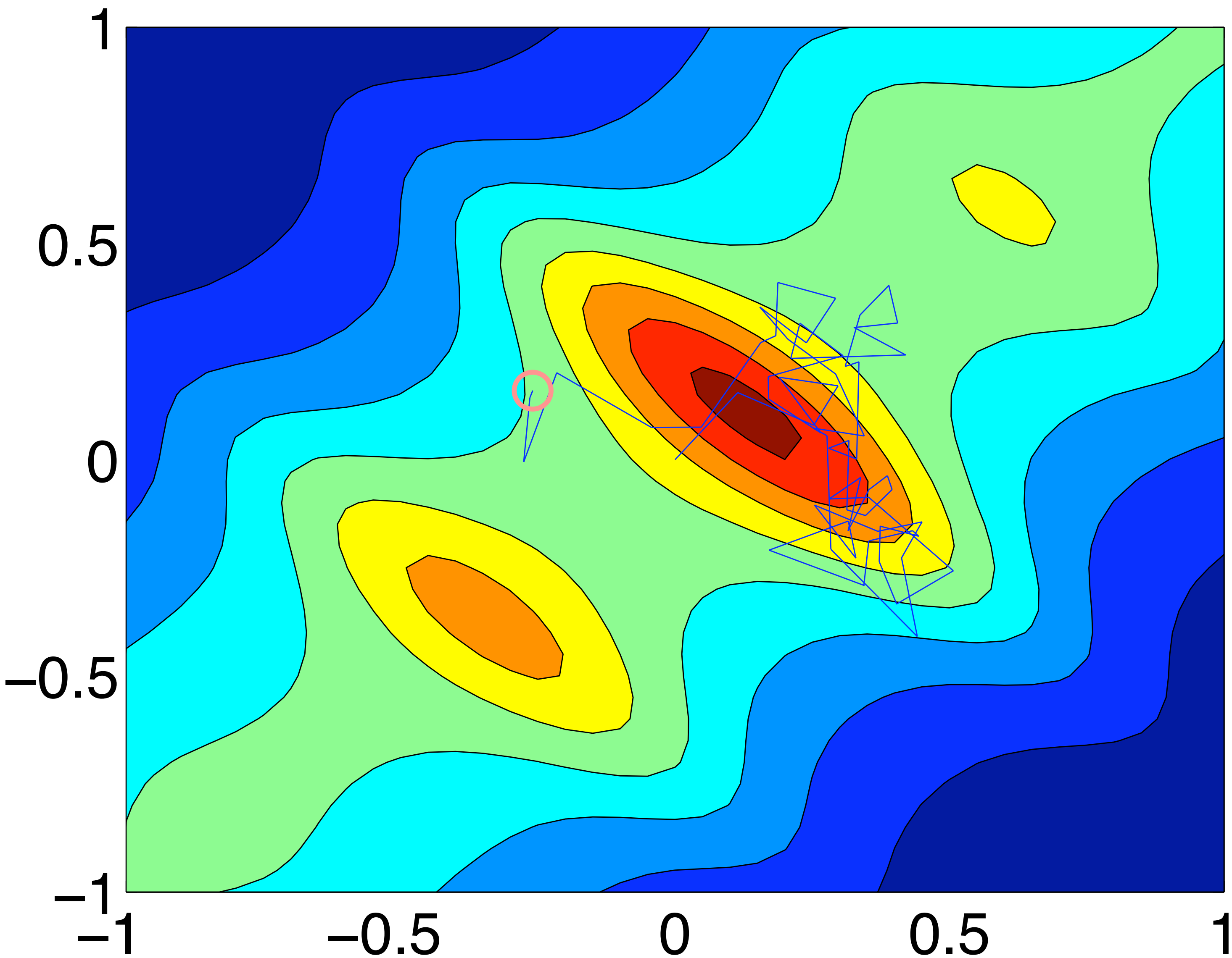


$Z P(x)$

$Z P(x)$

$E_P(g)$

MH example



In example

- g(x) = x^2
- True E(g(X)) = 0.28…
- Proposal: $Q(x' \mid x) = N(x' \mid x, 0.25^2 I)$
- Acceptance rate 55–60%
- After 1000 samples, minus burn-in of 100:

```
final estimate 0.282361
final estimate 0.271167
final estimate 0.322270
final estimate 0.306541
final estimate 0.308716
```

Gibbs sampler

- Special case of MH
- Divide $\mathbf{X}$ into blocks of r.v.s B(1), B(2), …
- Proposal Q:
  - pick a block i uniformly
  - sample $\mathbf{X}_{B(i)} \sim P(\mathbf{X}_{B(i)} \mid \mathbf{X}_{\neg B(i)})$

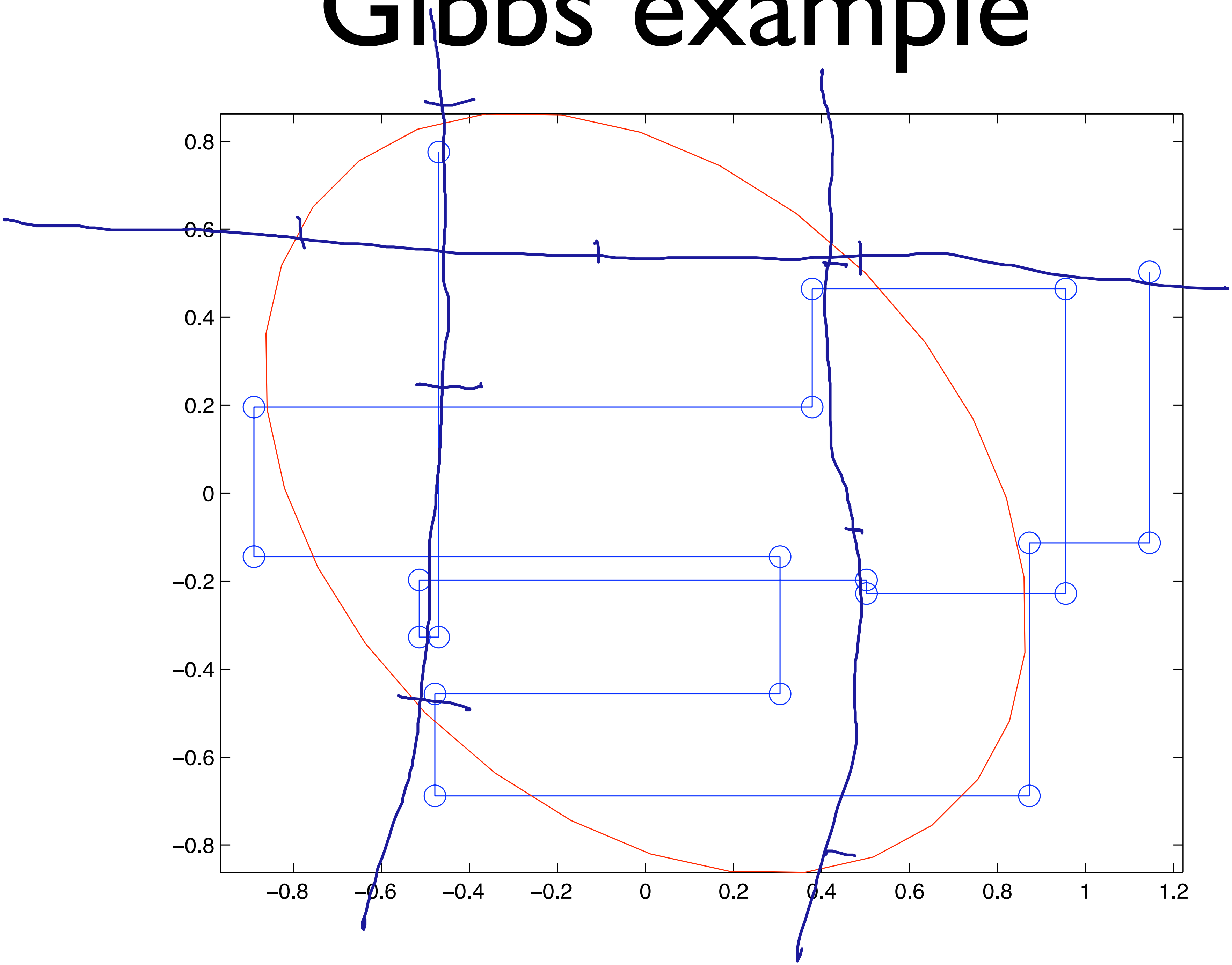
might be hard, but easier than joint

$$q(x' \mid x) = P(x' \mid x) = \frac{P(x', x)}{P(x)}$$

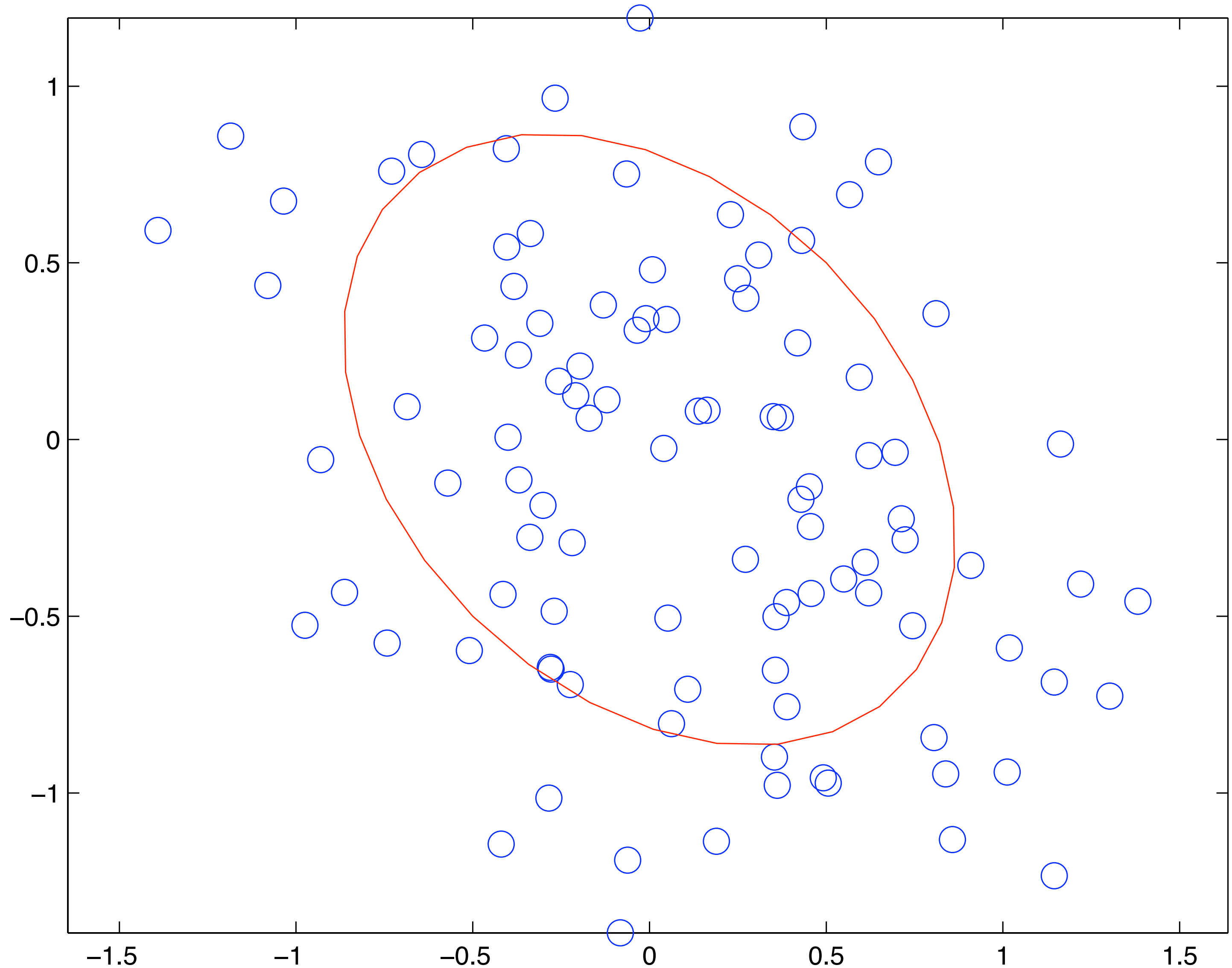
- Useful property: acceptance rate p = 1

$$\frac{p(x')}{p(x)} \frac{q(x \mid x')}{q(x' \mid x)} = \frac{\cancel{p(x')}}{\cancel{p(x)}} \frac{\cancel{P(x, x')} / \cancel{P(x')}}{\cancel{P(x, x')} / \cancel{p(x)}} = 1$$

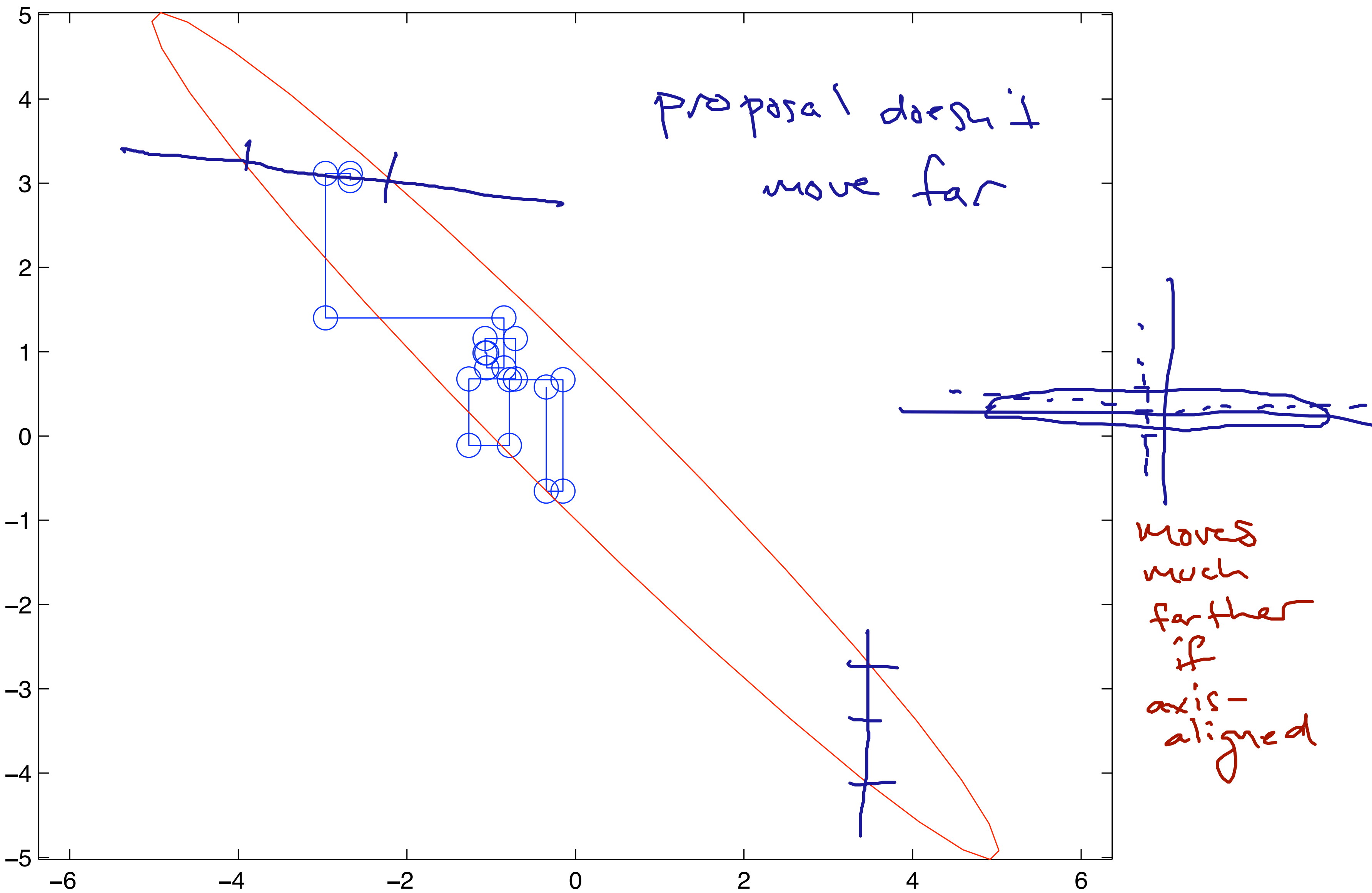
Gibbs example



Gibbs example



Gibbs failure example



Relational learning

- Linear regression, logistic regression: ***attribute-value*** learning

  attributes

  label

  - set of i.i.d. samples from P(X,Y)
- Not all data is like this
  - an attribute is a property of a single entity
  - what about properties of sets of entities?

Application: document clustering

10-601 Machine Learning Fall 2009

Geoff Gordon and Miroslav Dudik
School of Computer Science, Carnegie Mellon University

About | People | Lectures | Recitations | Homework | Exams | Projects

- Mailing lists
- Textbooks
- Grading
- Auditing
- Homework policy
- Collaboration policy
- Late policy
- Regrade policy
- Final project

Class lectures: Mondays and Wednesdays 10:30-11:50 in Newell Simon Hall 1305

Recitations: Wednesday, 6:00-8:00 pm GHC 8102

HW3 is out! It's due on Wednesday Oct 7, 10:30 am

Machine Learning is concerned with computer programs that learn to make better predictions or take better actions given increasing numbers of observations (e.g., programs that learn to spot high-risk medical patients, recognize human faces, recommend music and movies, or drive autonomous robots). This course covers theory and practical algorithms for machine learning from a variety of perspectives. We cover topics such as Bayesian networks, boosting, support-vector machines, dimensionality reduction, and reinforcement learning. The course also covers theoretical concepts such as bias-variance trade-off, PAC learning, margin-based generalization bounds, and Occam's Razor. Short programming assignments include hands-on experiments with various learning algorithms. Typical assignments include learning to automatically classify email by topic, and learning to automatically classify the mental state of a person from brain image data. The course will include a term project where the students will have opportunity to explore some of the class topics on a real-world data set in more detail.

Students entering the class with a pre-existing working knowledge of probability, statistics and algorithms will be at an advantage, but the class has been designed so that anyone with a strong numerate background can catch up and fully participate. This class is intended for Masters students and advanced undergraduates.

Announcement Emails

contains$(doc_i, word_j]$

count(d_i, w_j)

In Header

In Body

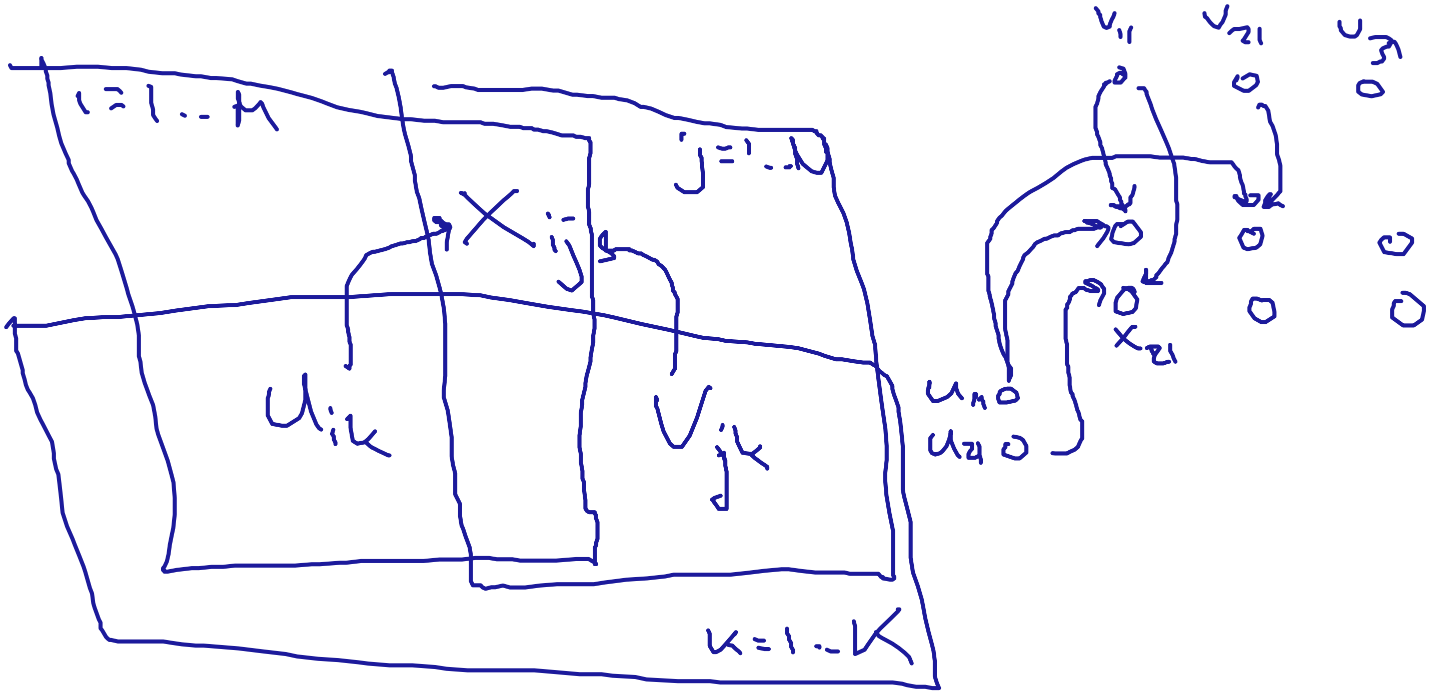
⋮

Application: recommendations

Rents(u, M)

	T2	Harry Potter	Koyaanisqatsi	...
A	✓	✓		
B		✓		
C			✓	
⋮				

Latent-variable models



$$\text{posterior}(X, U, V) \propto \prod_{ij} P(X_{ij} \mid U_{i\cdot}, V_{j\cdot}) \prod_{ik} P(U_{ik}) \prod_{jk} P(V_{jk})$$

$$E(X \mid U, V) = UV^T$$

$E(X \mid U,V)$: # users × # movies; U: # users × K; V^T: K × # movies

Best-known LVM: PCA

neater version

$$E(X \mid U,V) = UV^T$$

$E(X \mid U,V)$: # users × # movies; U: # users × K; V^T: K × # movies

$$U_{ik} \sim N(0, \tau^2)$$

$$V_{jk} \sim N(0, \tau^2)$$

$$X_{ij} \sim N\left(\sum_k U_{ik} V_{kj}, \sigma^2\right)$$

$$\text{rank}(UV^T) \leq K$$

- Suppose X_{ij}, U_{ik}, V_{jk} all ~ Gaussian
  - yields ***principal components analysis*** $\frac{\sigma}{\tau} \to 0$
  - or ***probabilistic PCA***
  - or ***Bayesian PCA***

MAP

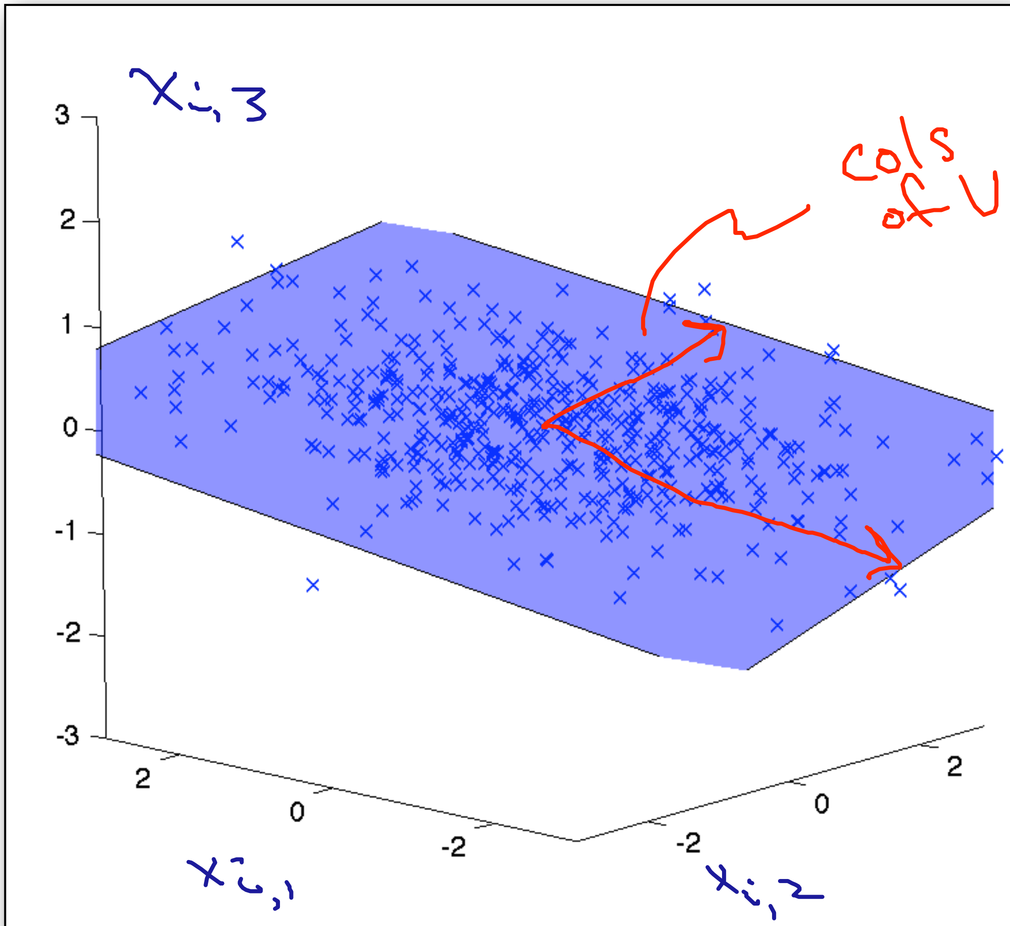
general σ, τ

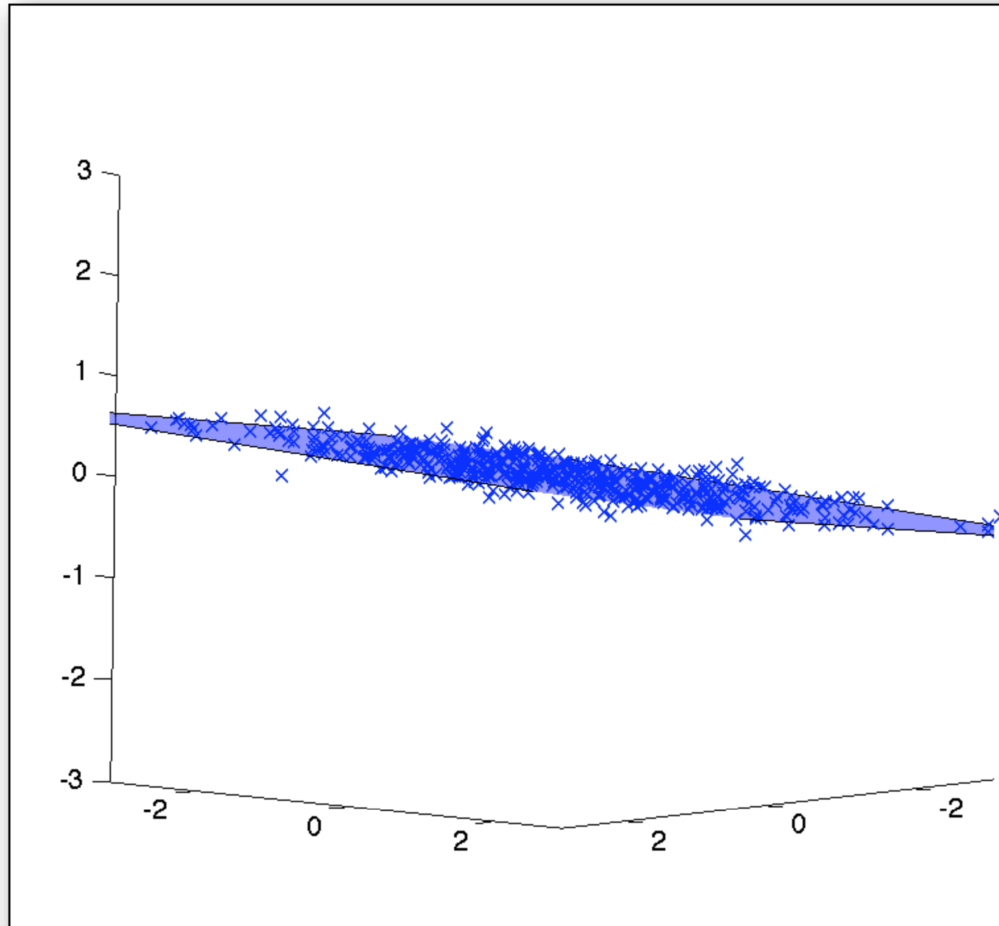
$X \in \mathbb{R}^{M\times 3}$ $N = 3$

$U \in \mathbb{R}^{M\times 2}$

$V \in \mathbb{R}^{3\times 2}$

PCA: the picture





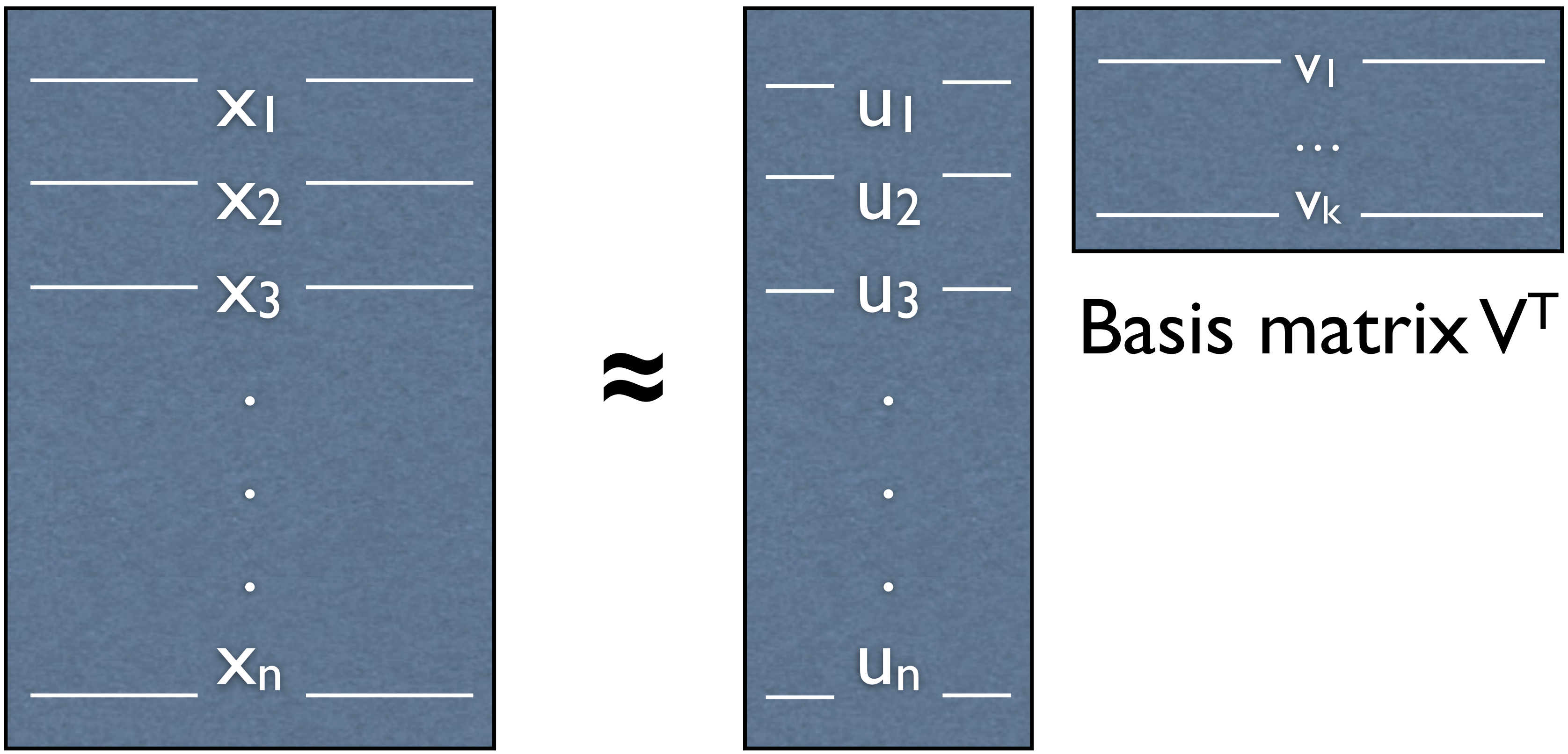
PCA: cartoon example

Movie

User	1	2	3	4	5	6	...
A	1	1	0	0	1	0	...
B	0	1	1	0	0	0	...
C	1	1	0	1	1	0	...
D	1	0	0	1	1	0	...
E	0	1	0	1	0	0	...
F	0	1	1	1	0	1	...
...	...	...	...	...	...	...	...

$= X$

PCA: cartoon example

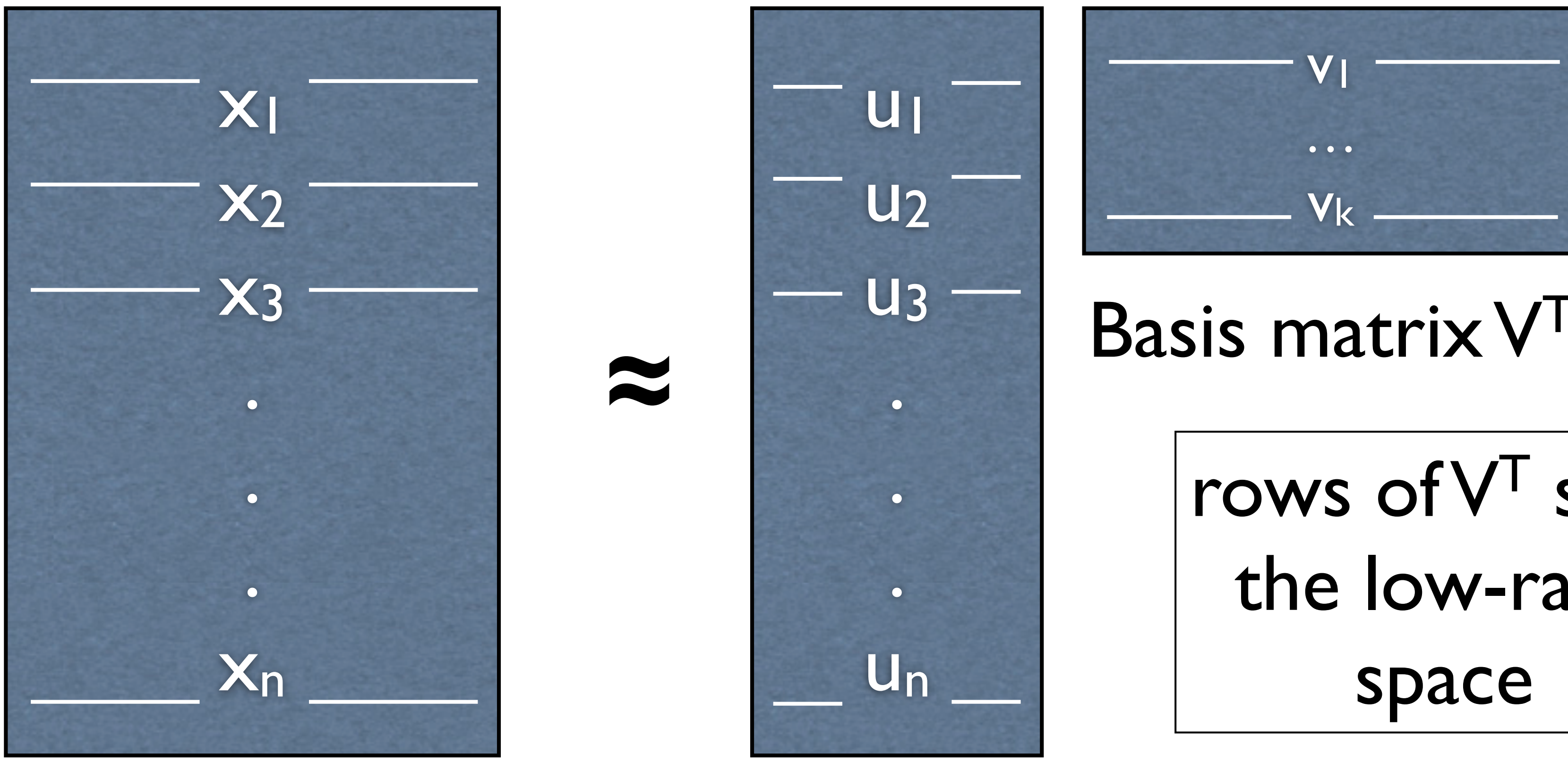


Basis matrix V^T

Data matrix X

Compressed matrix U

PCA: cartoon example



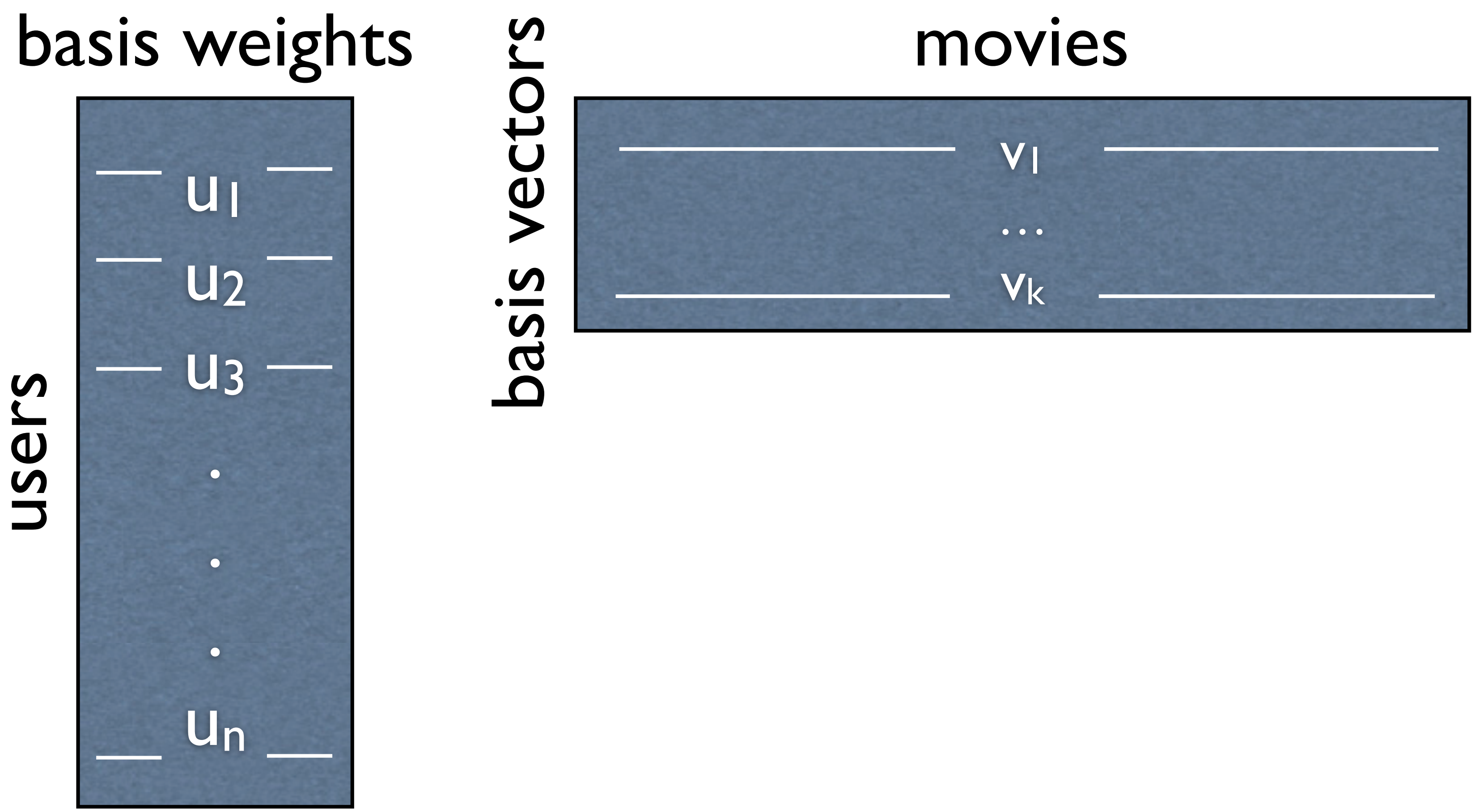
Basis matrix V^T

rows of V^T span the low-rank space

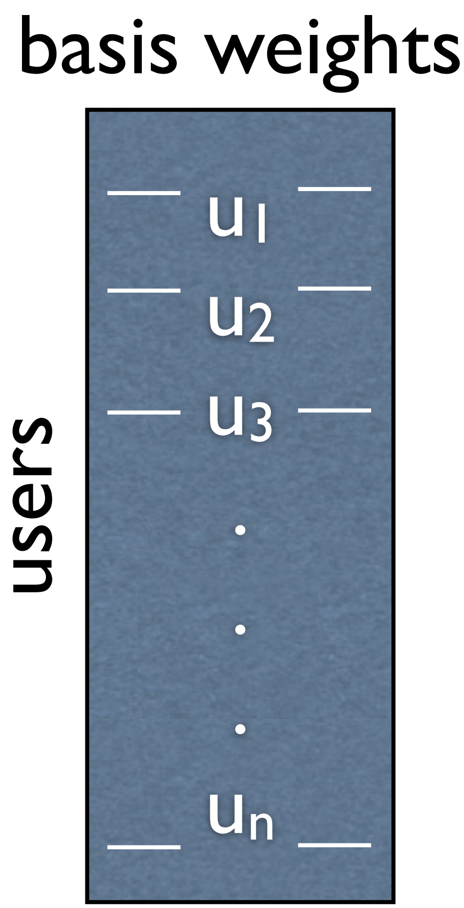
Data matrix X

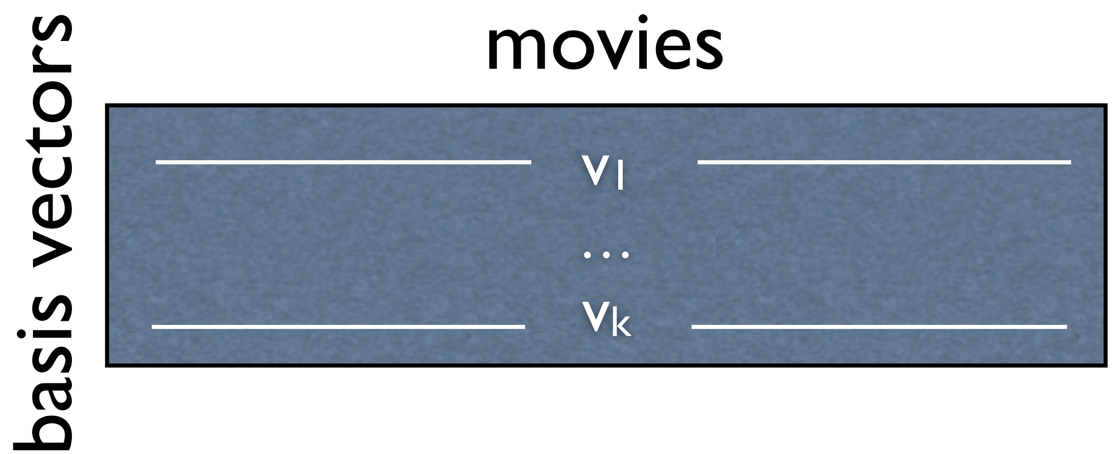
Compressed matrix U

Interpreting PCA



Interpreting PCA





Basis vectors represent movies that ***vary together***

Weights say how much each user cares about each type of movie

Mean subtraction

- $U_{ik} \sim N(0, \nu^2)$
- $V_{jk} \sim N(0, \nu^2)$
- $X_{ij} \sim N(U_i \cdot V_j, \sigma^2)$

$\mu, \mu_i^{row}, \mu_j^{col} \sim N(0, \tau^2)$

$\rightarrow X_{ij} \sim N(\mu + \mu_i^{row} + \mu_j^{col} + U_i \cdot V_j, \sigma^2)$

```
>> mu = mean(X(:));
>> colmu = mean(X - mu);
>> rowmu = mean(X' - mu)';
>> X = X - mu - repmat(colmu, size(X,1), 1) -
      repmat(rowmu, 1, size(X,2));
```

Code for MLE of means

Data weights

- Let W_{ij} = $\begin{cases} 1 & \text{if } x_{ij} \text{ observed} \\ 0 & \text{o/w} \end{cases}$

- Likelihood · prior = $\prod_{ij} P(x_{ij} \mid u_i, v_j)^{\overbrace{W_{ij}}} \prod_{ik} P(u_{ik}) \prod_{jk} P(v_{jk})$

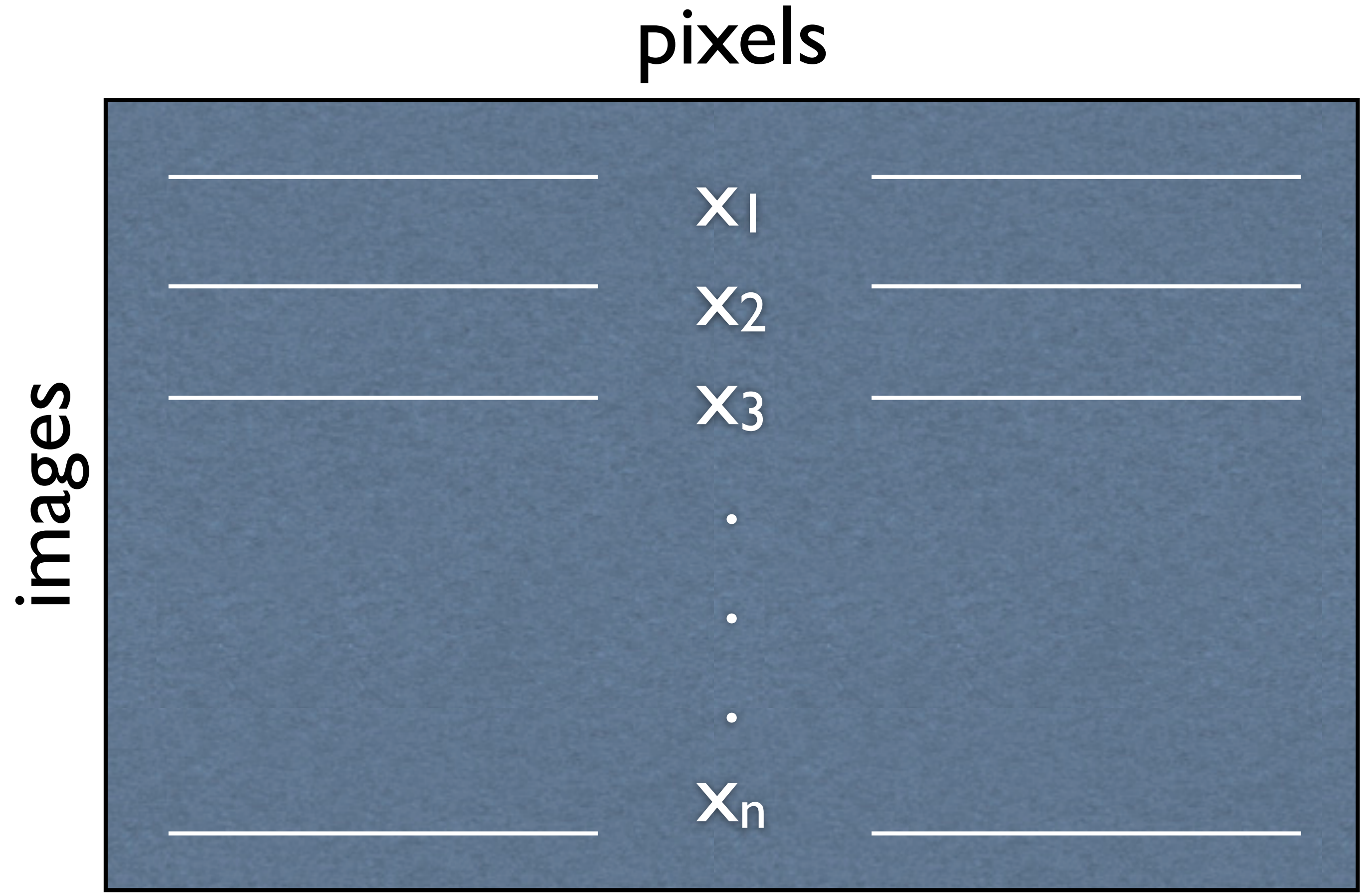
  gates whether this term is included

- More generally, $W_{ij} \geq 0$

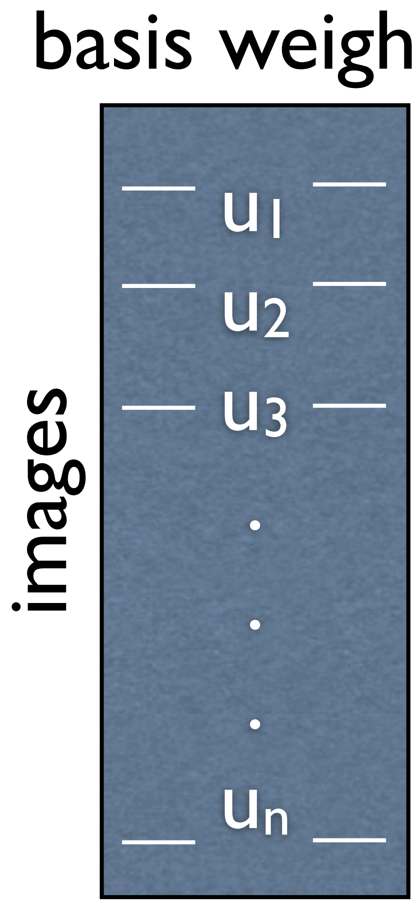
Another use of PCA

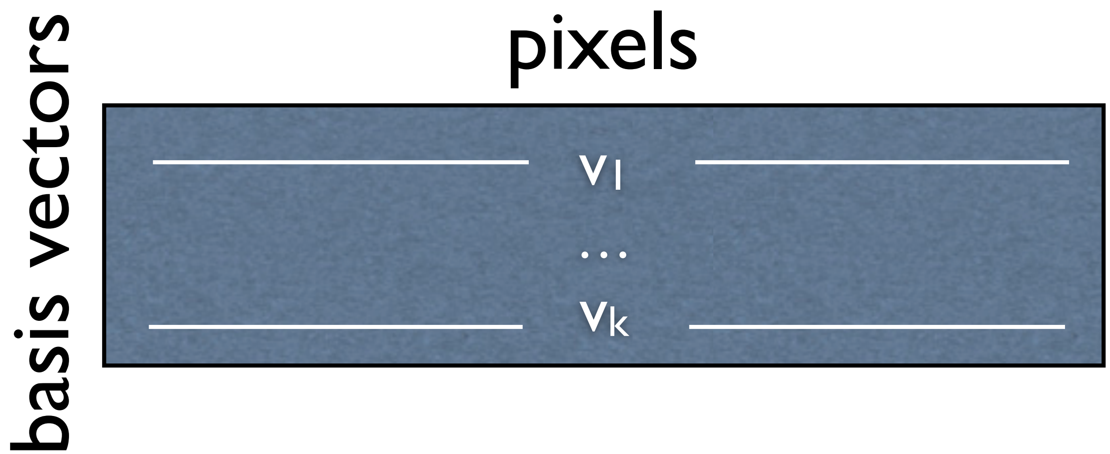
face images from Groundhog Day, extracted by Cambridge face DB project

Image matrix



Result of factoring





Basis vectors are often called "eigenfaces"

Eigenfaces

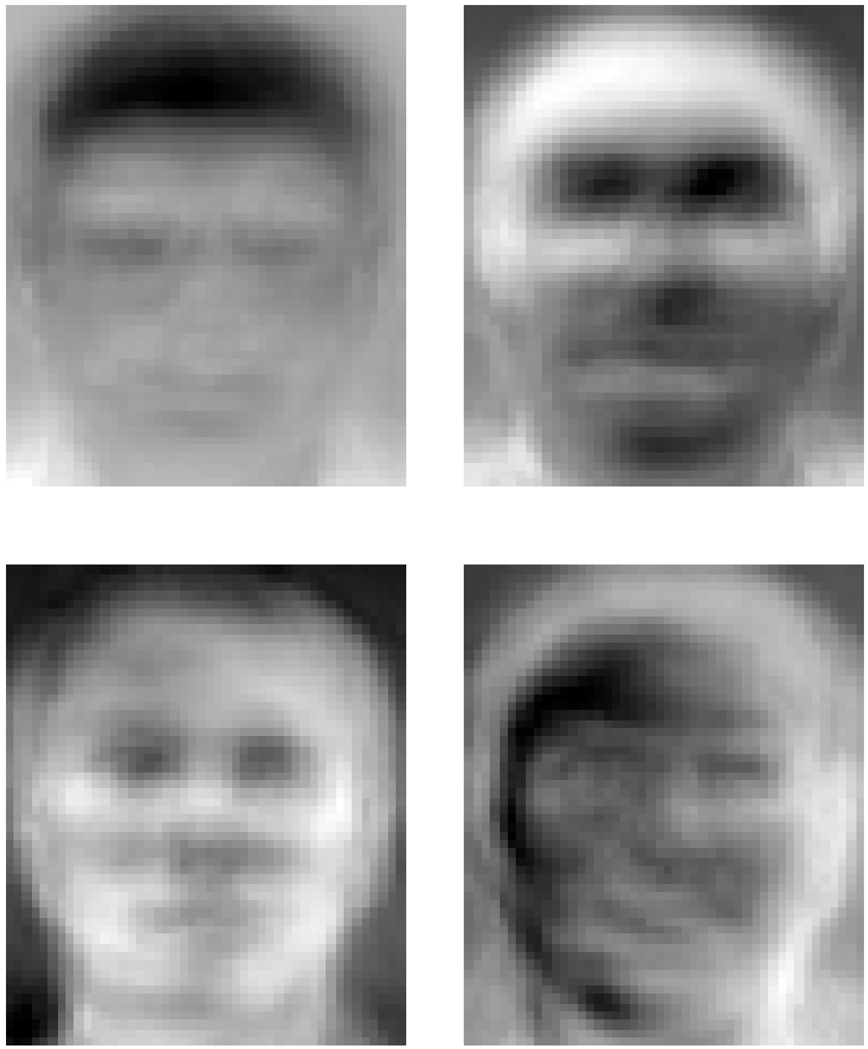


image credit: AT&T Labs
Cambridge

PCA: finding the MLE

$E(X \mid U, V) = \hat{X} = UV^T$

- PCA:
  - $U_{ik} \sim N(0, \nu^2)$
  - $V_{jk} \sim N(0, \nu^2)$
  - $X_{ij} \sim N(U_i \cdot V_j, \sigma^2)$
  - $\sigma/\nu \to 0$

$-\log$ likelihood $=$ const $+$

$\sum_{ij} (X_{ij} - \hat{X}_{ij})^2 / 2\sigma^2$

($+$ irrelevant prior)

$\Rightarrow \min_{UV} \sum_{ij} (X_{ij} - \hat{X}_{ij})^2$

$= \|X - \hat{X}\|_F^2$

Summary: to do PCA, find SVD, set small elements of Σ to 0

PCA & SVD

- The ***singular value decomposition*** is
  - $X = R\,\Sigma\,S^T$ $\mathbb{R}^{M\times N}$, $\mathbb{R}^{M\times N}$, $\mathbb{R}^{N\times N}$
  - R, S orthonormal; $\Sigma \geq 0$ diagonal $R^T R = I$, $S^T S = I$
  - All matrices can be expressed this way
  - See svd, svds in Matlab

$$X_k = R_{\cdot k} S_{\cdot k}^T$$

$$X = \sum_{k=1}^{N} R_{\cdot k} S_{\cdot k}^T \Sigma_{kk} = \sum_{k=1}^{N} \Sigma_{kk} X_k \qquad \|X_k\|_F = 1$$

$$\hat{X} = \sum_{k=1}^{K} \Sigma_{kk} X_k \qquad \|X - \hat{X}\|_F^2 = \sum_{k=K+1}^{N} \Sigma_{kk}$$

$$X_k \text{ ``}\cdot\text{'' } X_{k'} = 0 \qquad A \text{ ``}\cdot\text{'' } B = \sum_{ij} A_{ij} B_{ij}$$

Then, to get UV^T instead of $R\Sigma S^T$, include $\sqrt{\Sigma}$ into both R and S ← componentwise square root

- So, PCA is $U = R_k \Sigma_k^{1/2}$ $V = S_k \Sigma_k^{1/2}$

$$\Sigma_k = \Sigma(1..k, 1..k)$$