Techniques for Dimensionality Reduction

PCA and Other Matrix Factorization Methods

Outline

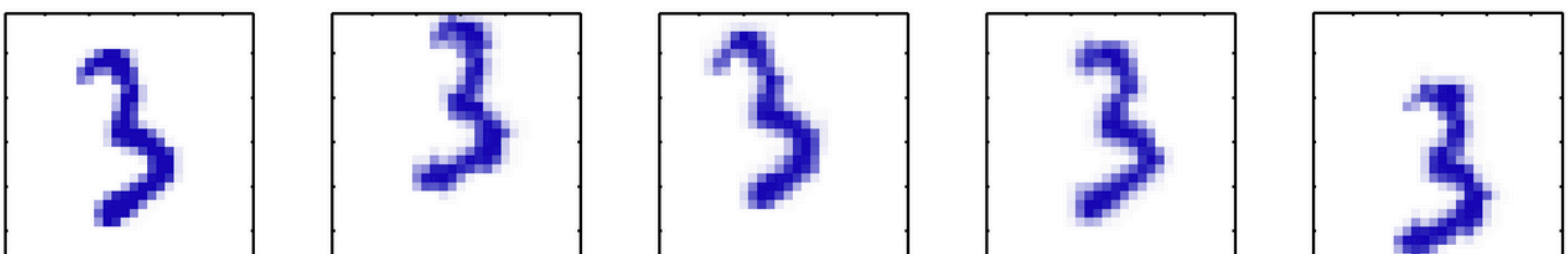
- Principle Compoments Analysis (PCA)
  - Example (Bishop, ch 12)
  - PCA as a mixture model variant
    - With a continuous latent variable
  - Breaking down PCA
    - Optimization problem
    - Solution
    - Intuitions
- General matrix factorization
  - Application to collaborative filtering
  - Algorithms
  - Wrap-up

A Motivating Example

- The MNist digits problem was simplified because the digits were
  - Centered
  - In a canonical position
  - Scaled to the same size
- What if they weren't?

A Motivating Example

- Take a *single* 64*64 digit and create a dataset by repeatedly
  - Move it to a 100*100 image
  - Shift by *x*,*y* and rotate by θ
- Dataset has 10,000 features but really only needs 3

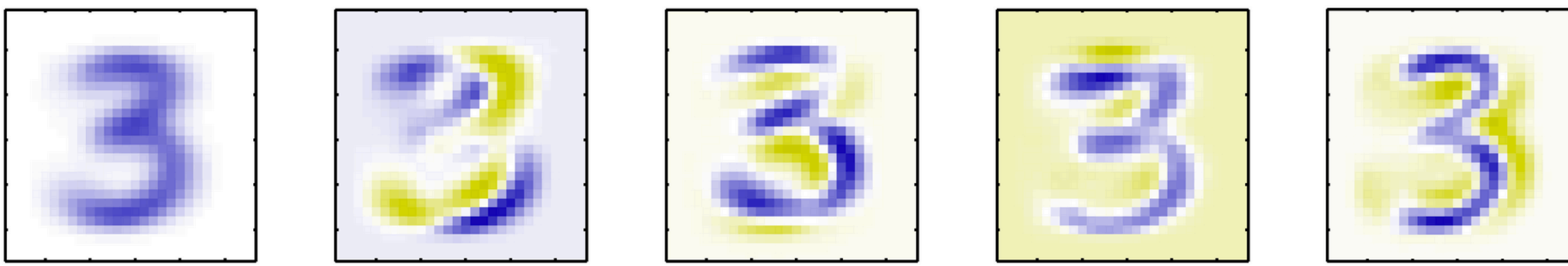


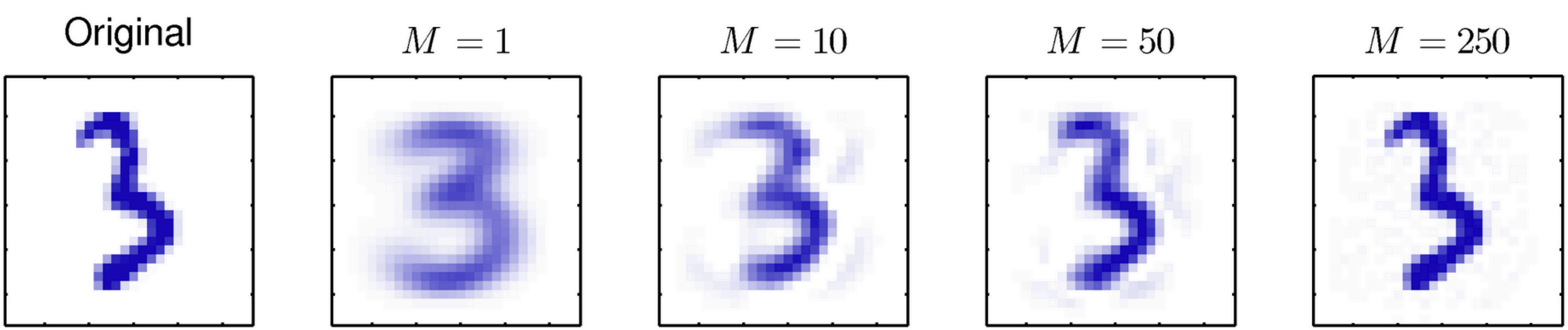
A Motivating Example

"prototype" = a vector of the same dimension as the instances

- PCA: reduces each instance to a linear combination of a few "prototypes" (blue+, green-). These are the first 5:

*A specific choice of prototypes are the **principle components***

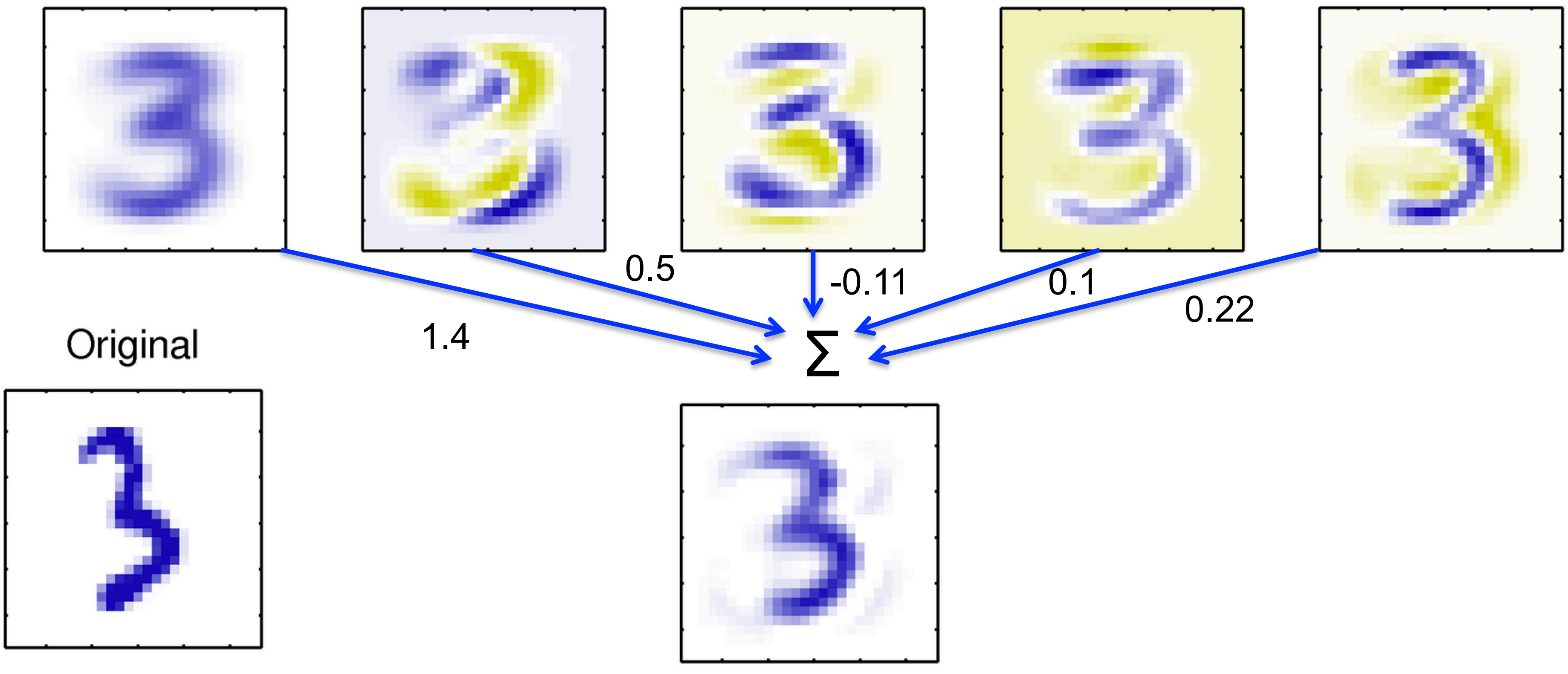




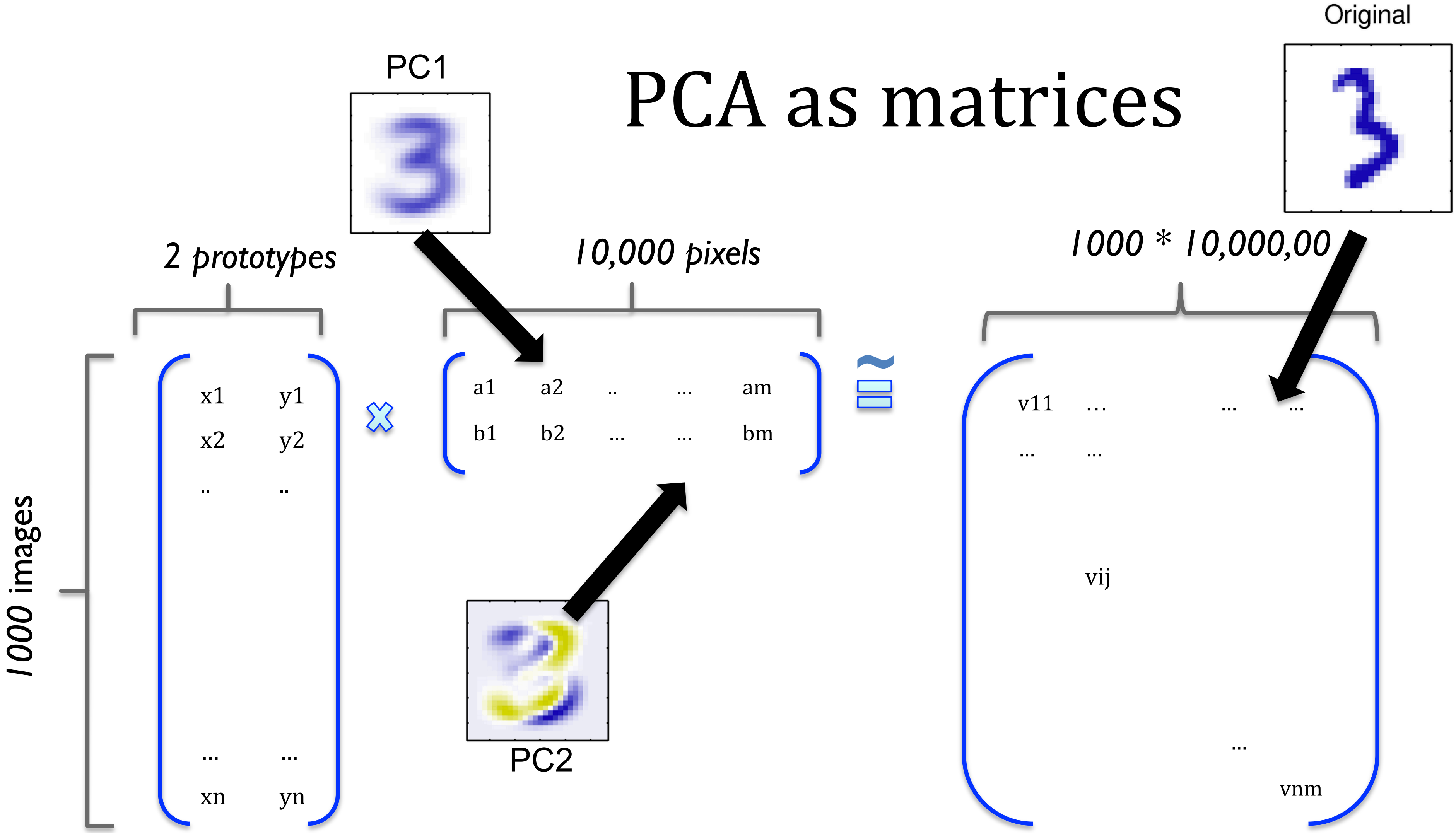
A Motivating Example

"prototype" = a vector of the same dimension as the instances

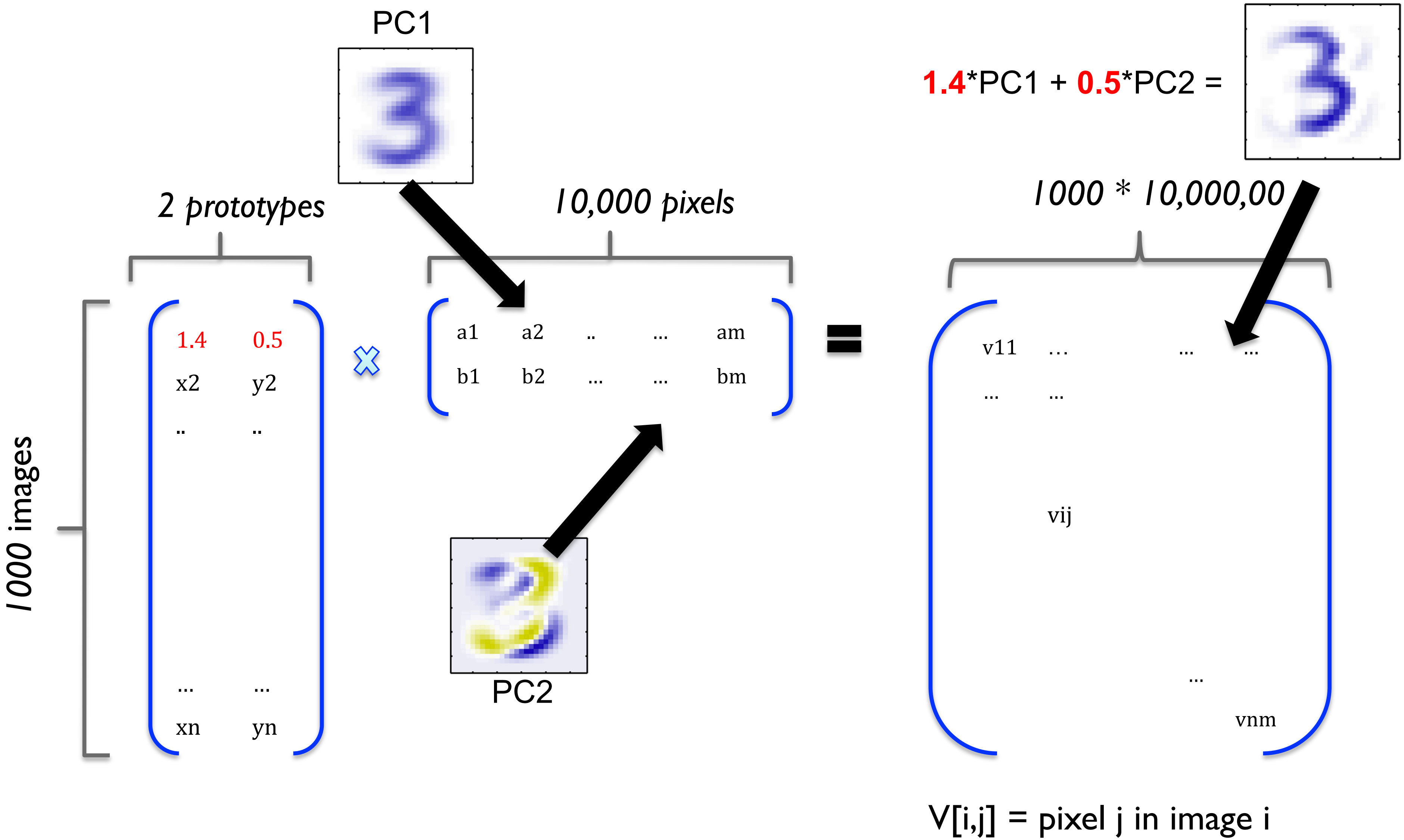
- PCA: reduces each instance to a linear combination of a few "prototypes" (blue+, green-). These are the first 5:



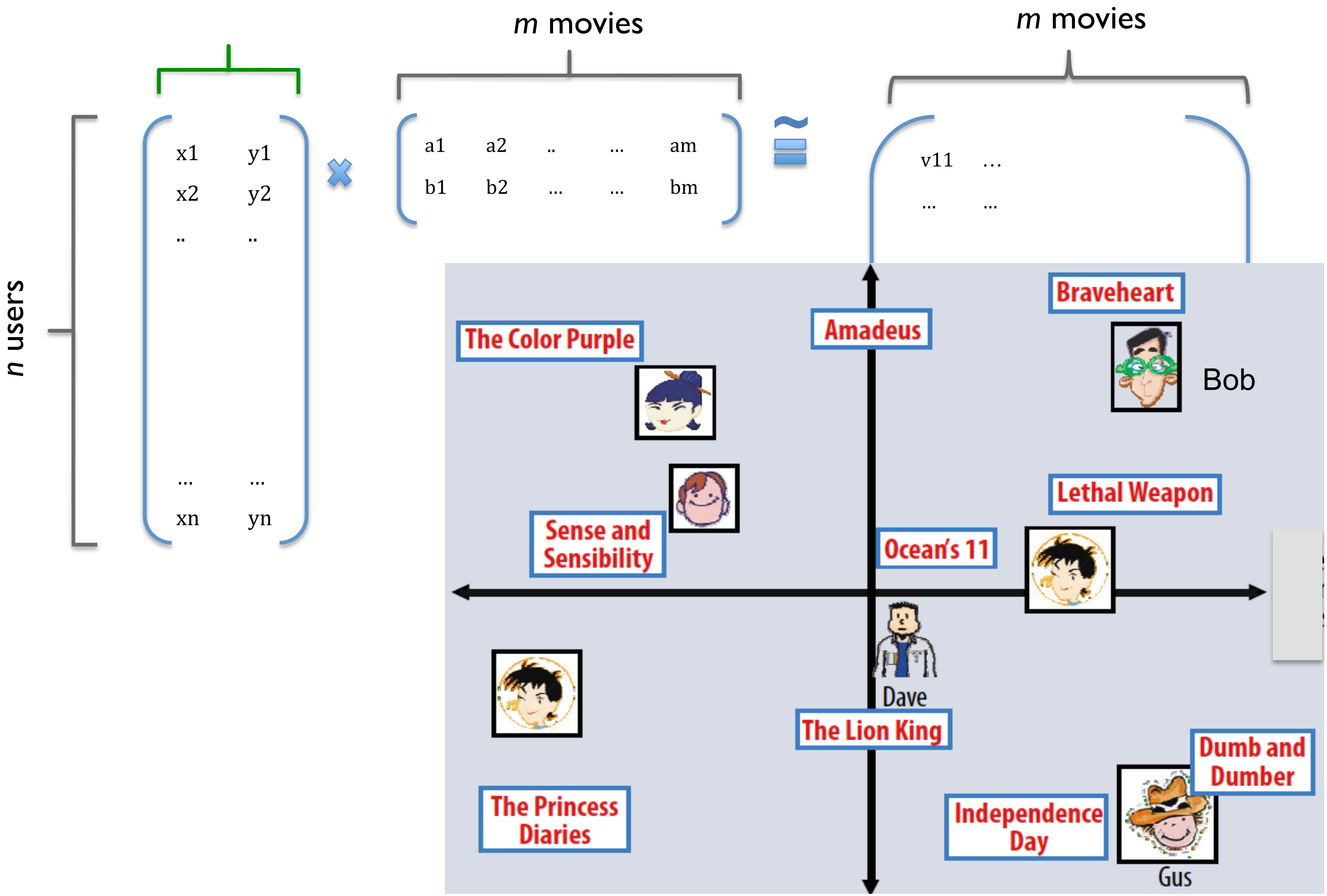
PCA as matrices

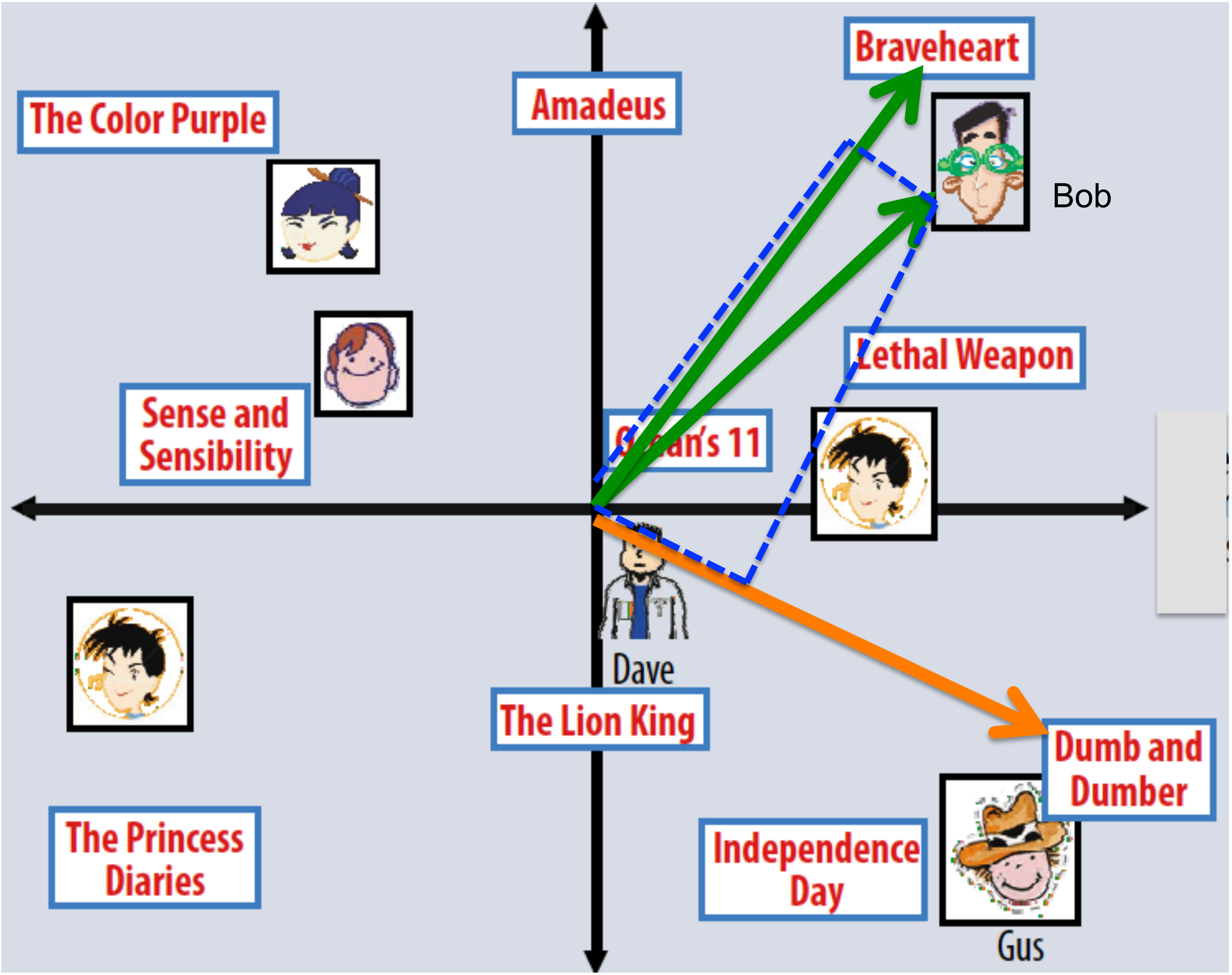


V[i,j] = pixel j in image i

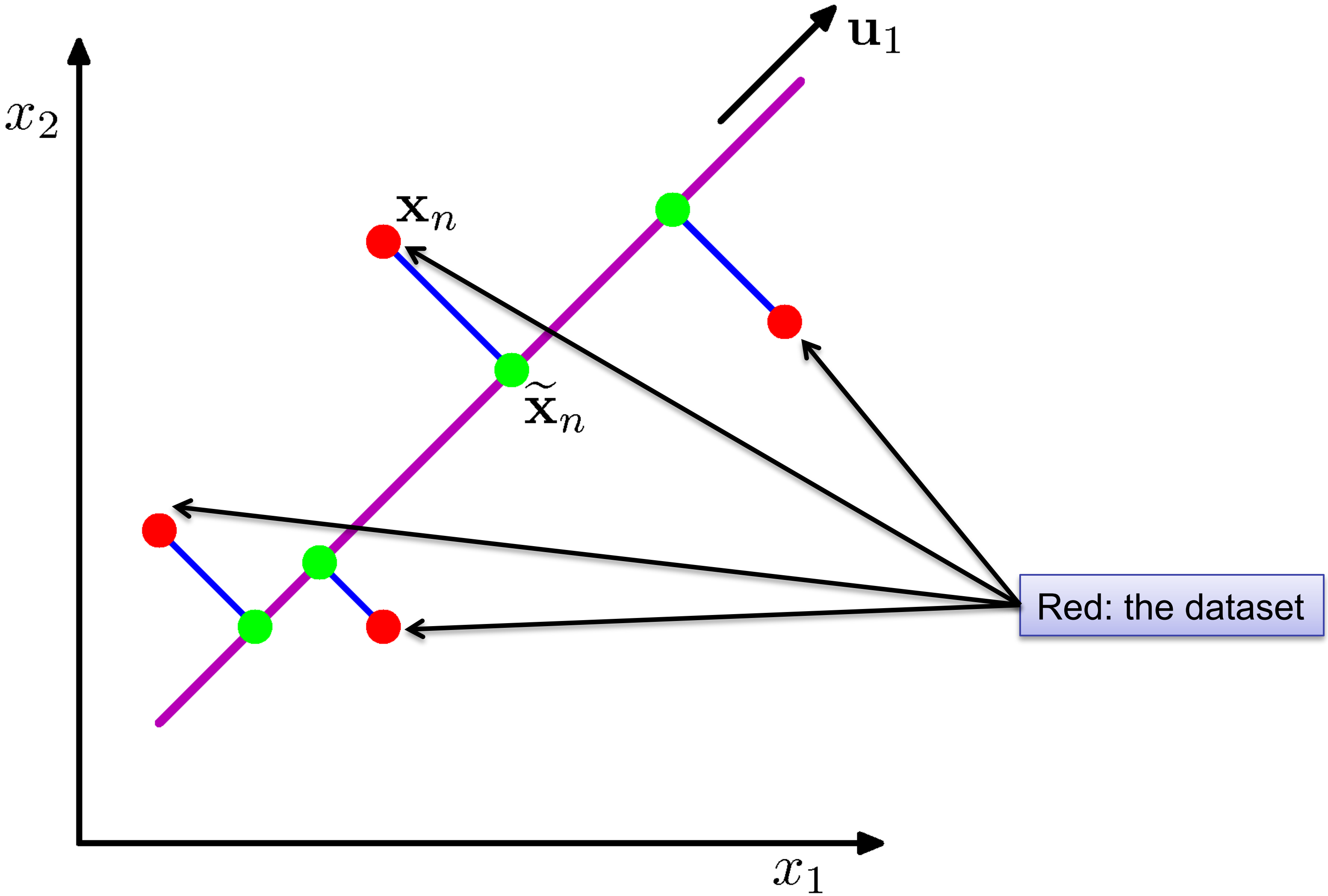

PC1
1.4*PC1 + 0.5*PC2 =
2 prototypes
10,000 pixels
1000 * 10,000,00
1000 images
1.4 0.5
x2 y2
xn yn
a1 a2 .. ... am
b1 b2 ... ... bm
v11
vij
vnm
PC2
V[i,j] = pixel j in image i

PCA for movie recommendation…

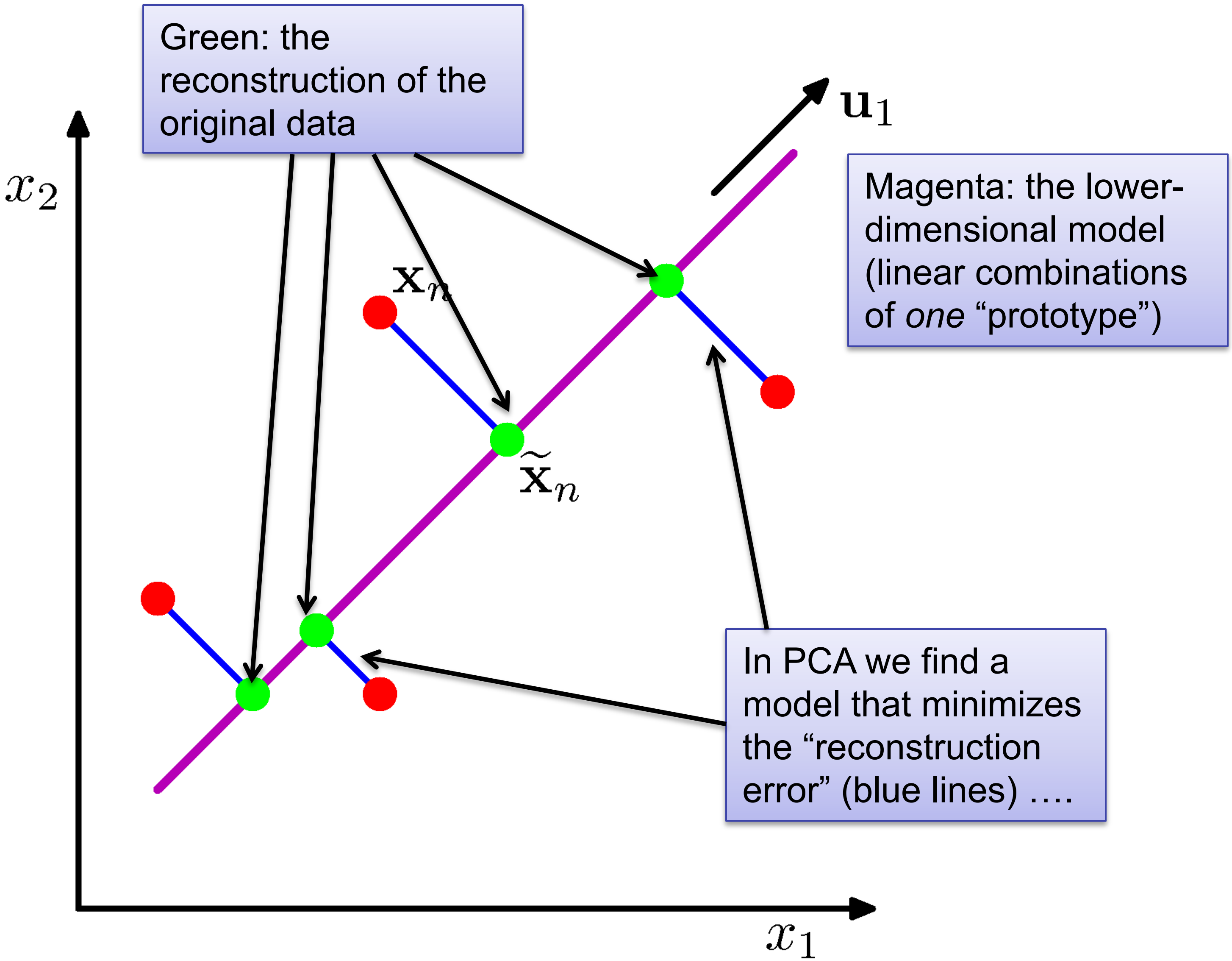



Braveheart
Amadeus
The Color Purple
Bob
Lethal Weapon
Sense and Sensibility
Dave
The Lion King
Dumb and Dumber
The Princess Diaries
Independence Day
Gus

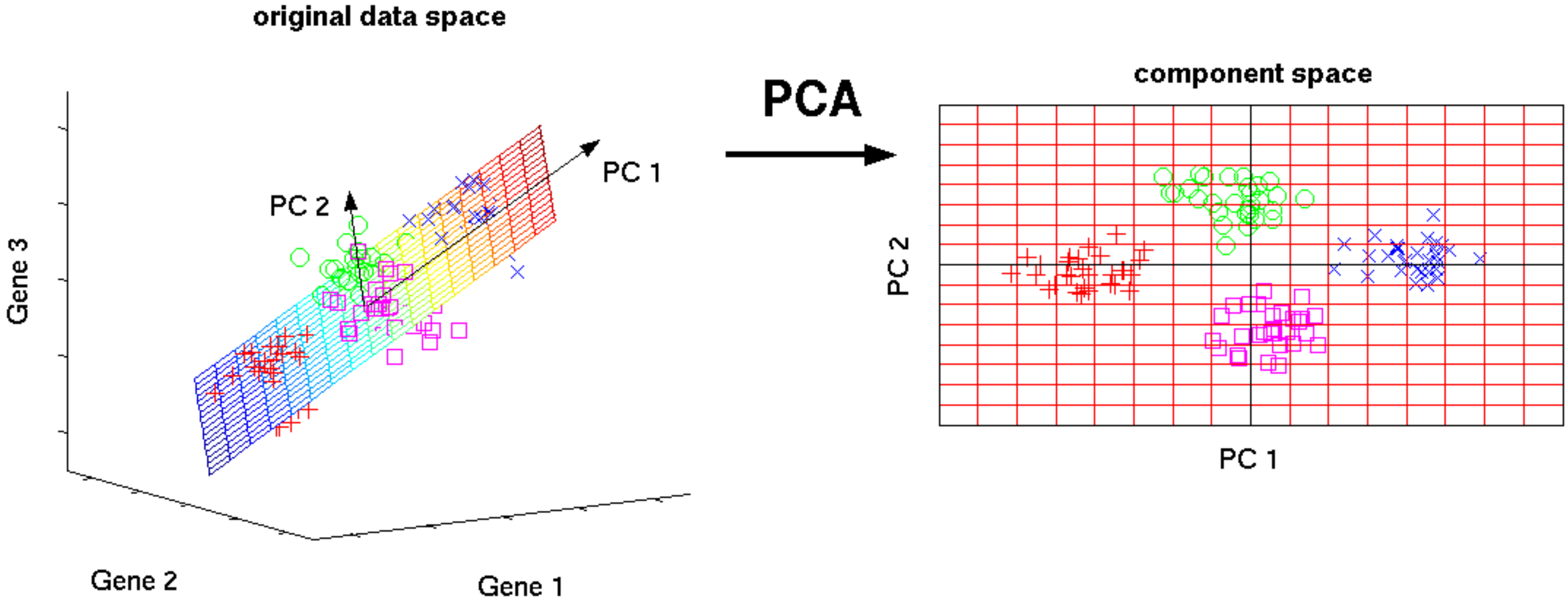
A Cartoon of PCA



A Cartoon of PCA

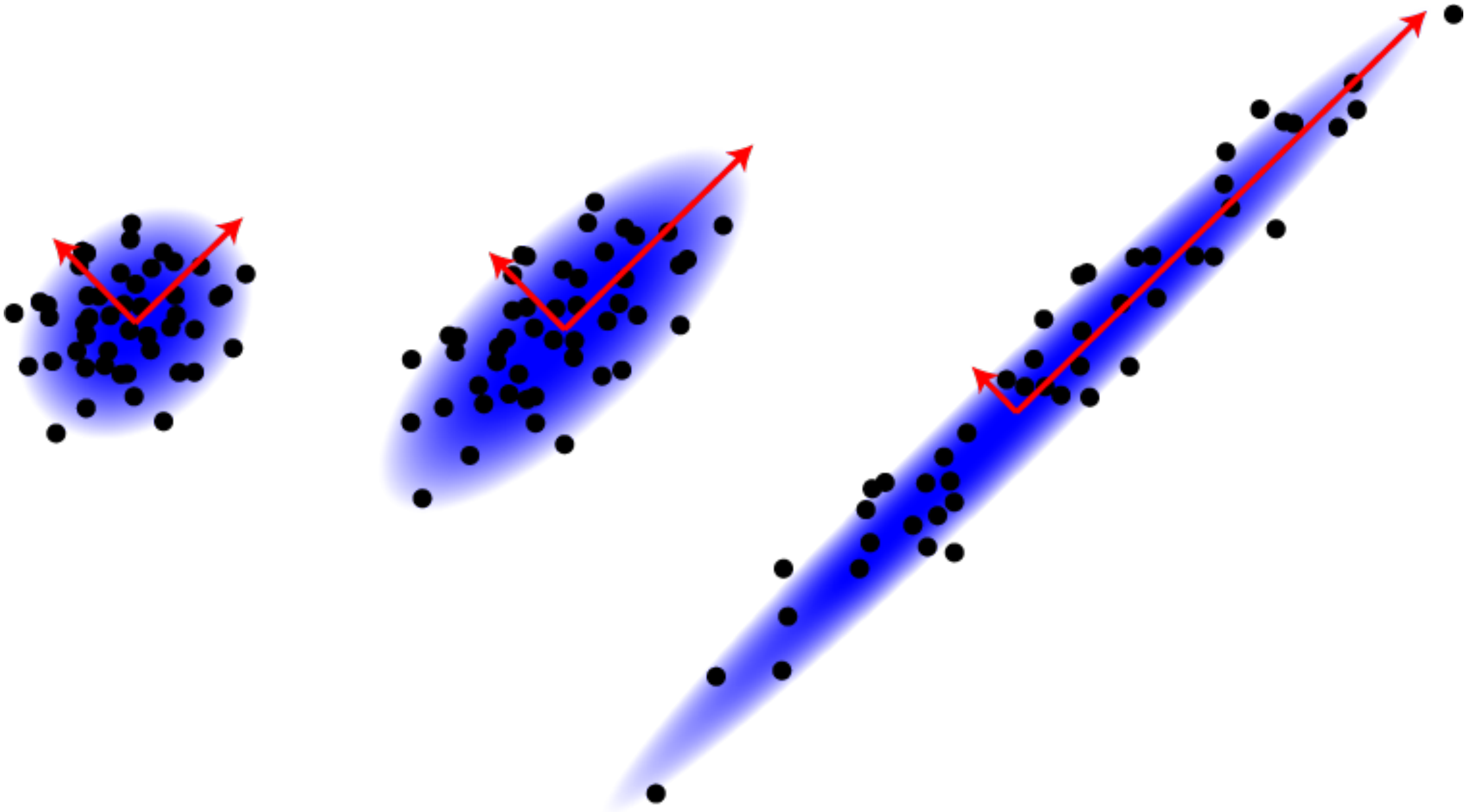


A 3D Cartoon of PCA

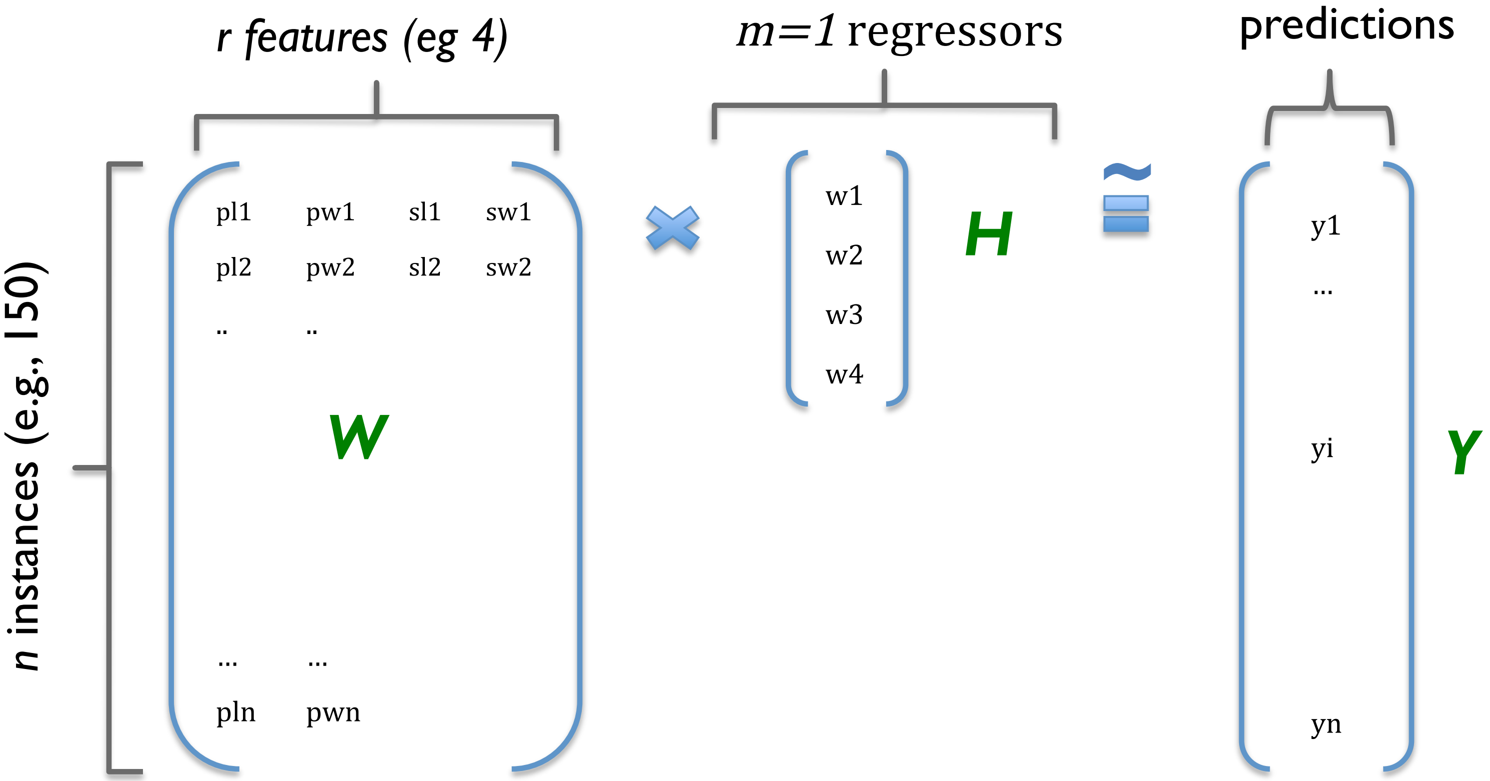


http://www.nlpca.org/

Some more cartoons

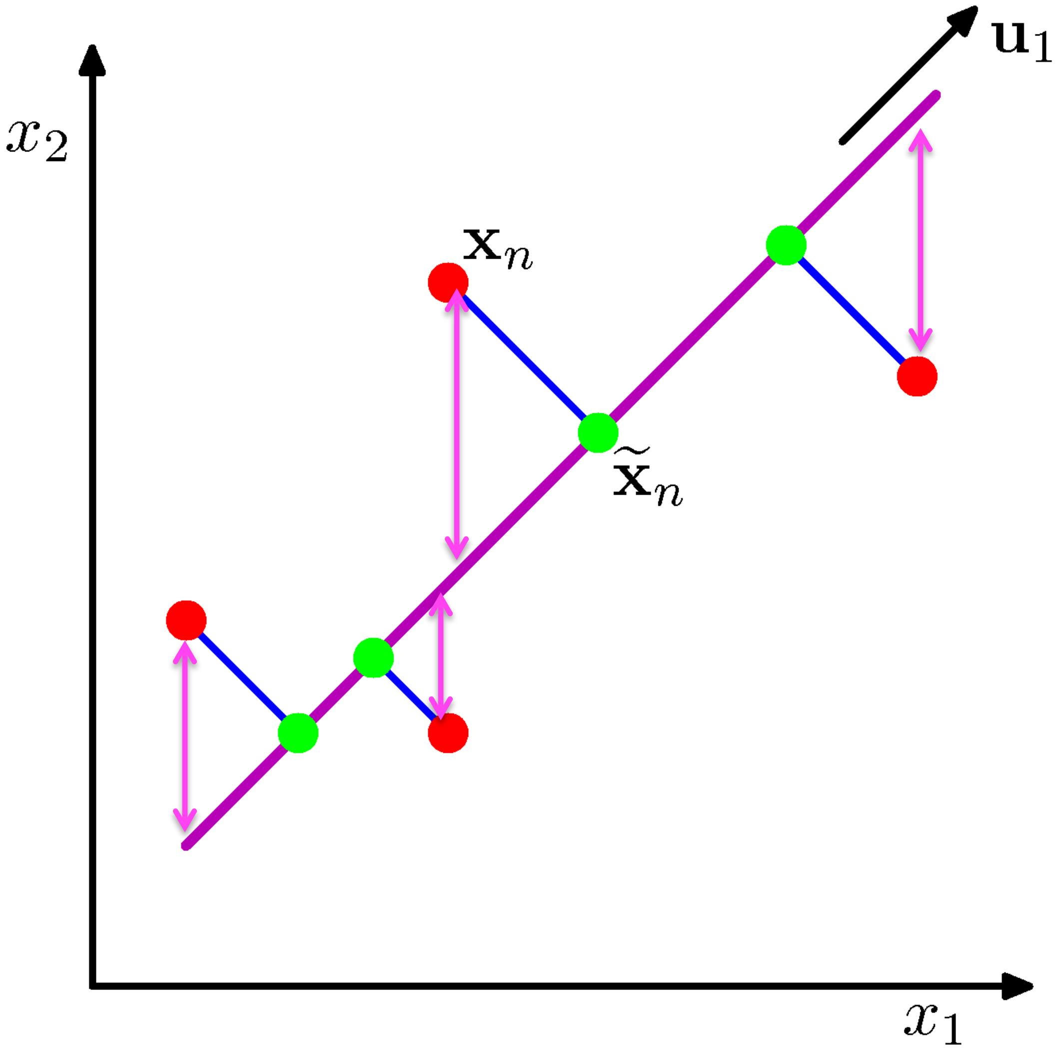


PCA *vs* Linear Regression



Y[i,1] = instance i's prediction

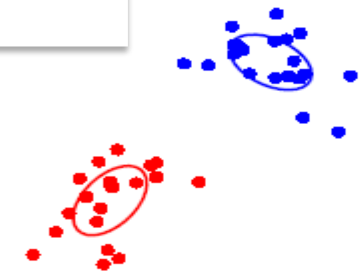
PCA vs Linear Regression



In contrast: in regression we'd minimize square error on *one* dimension (x_2) using a linear combination the *other dimensions*

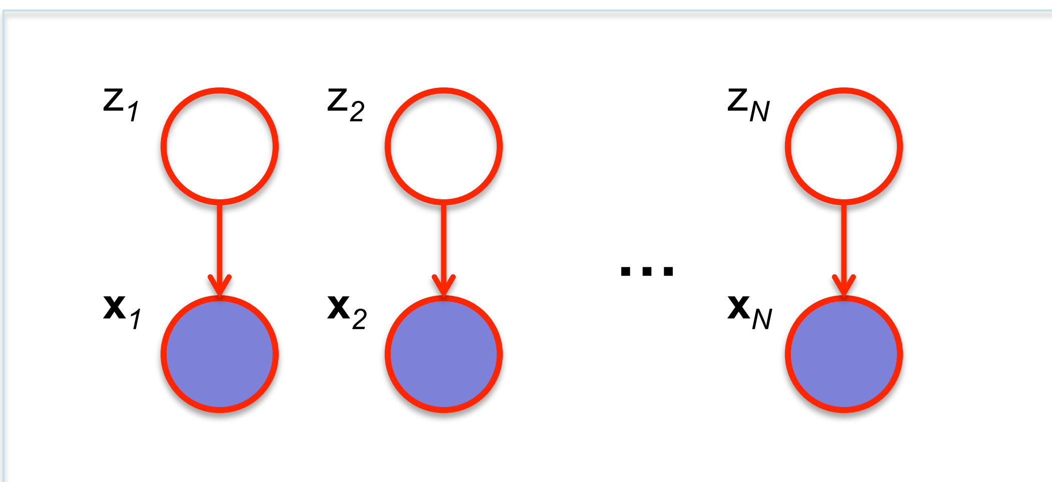
PCA vs mixture of Gaussians

Mixture of Gaussians



For each point:

- Pick the index of the (latent) Gaussian *Z=k*
- Pick the the point **x** from that the k-th Gaussian, **x** ~ $N(\mu_k,\Sigma_k)$



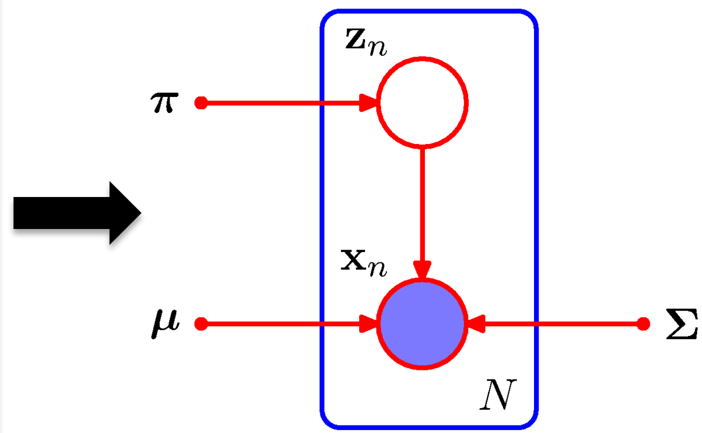
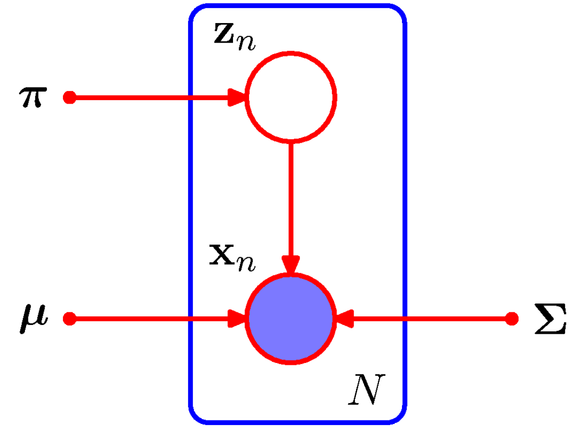


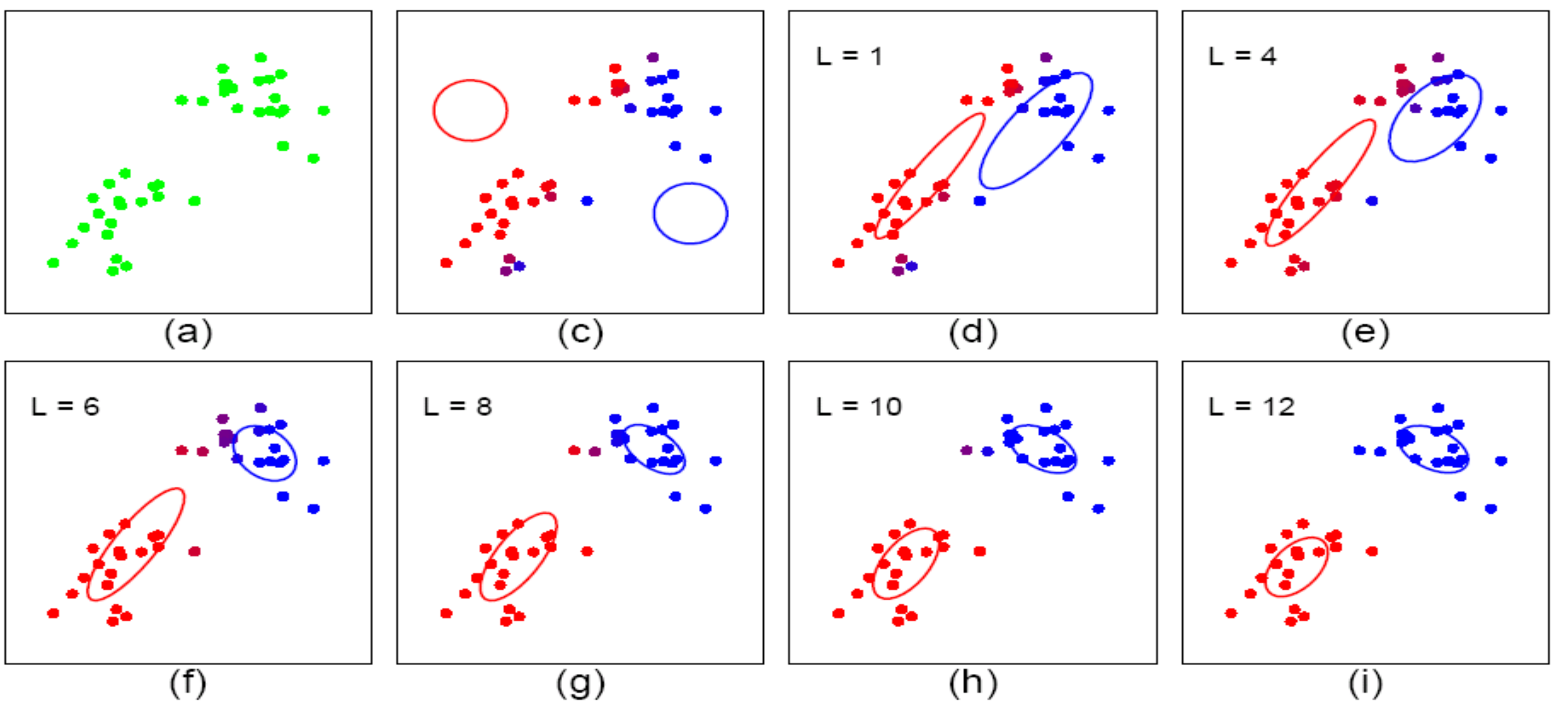
Plate notation

PCA vs mixture of Gaussians

Mixture of Gaussians

- Pick the index of the (latent) Gaussian *Z=k*
- Pick the the point **x** from that the k-th Gaussian, **x** $N(\mu_k,\Sigma_k)$

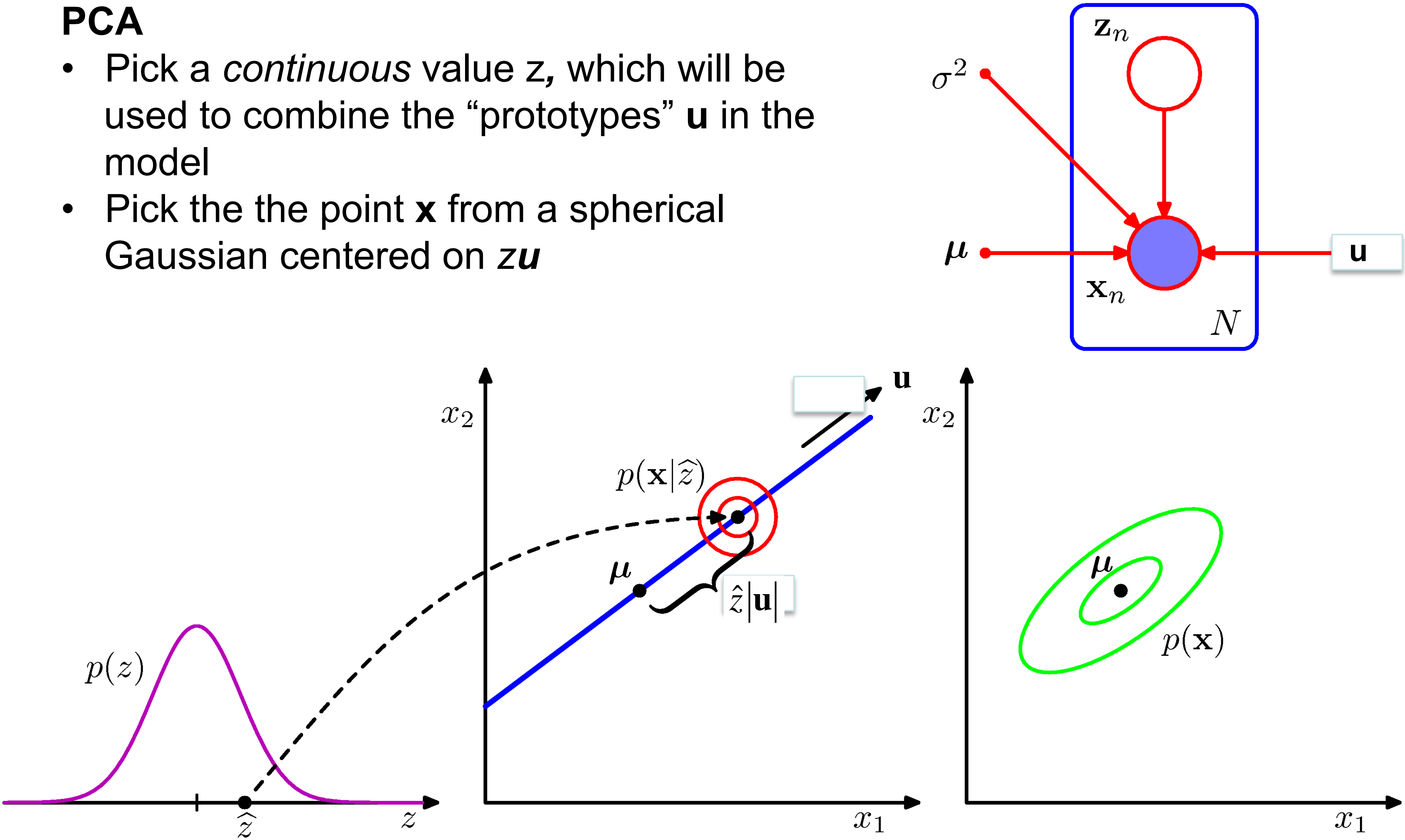




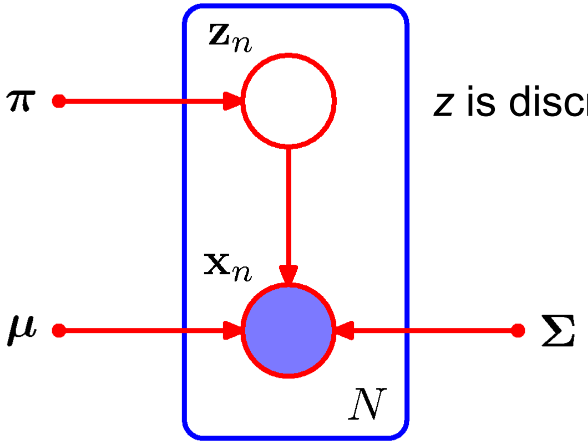
PCA vs mixture of Gaussians

PCA

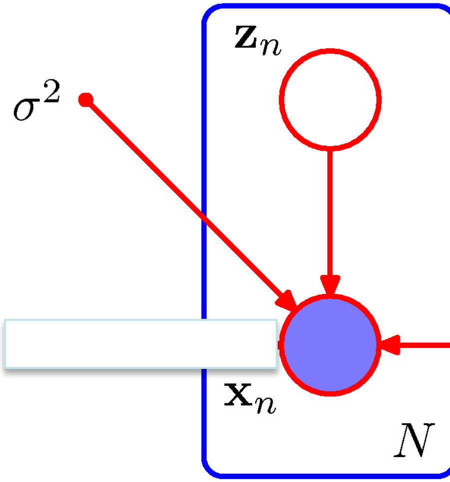
- Pick a *continuous* value z**,** which will be used to combine the "prototypes" **u** in the model
- Pick the the point **x** from a spherical Gaussian centered on ***zu***



PCA vs mixture of Gaussians

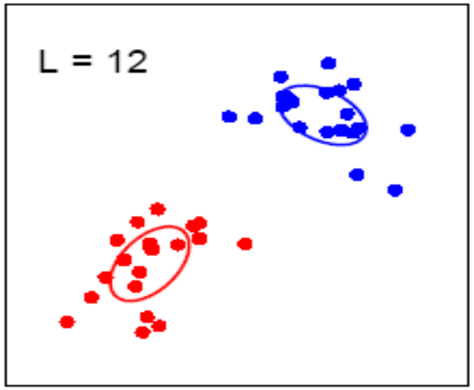


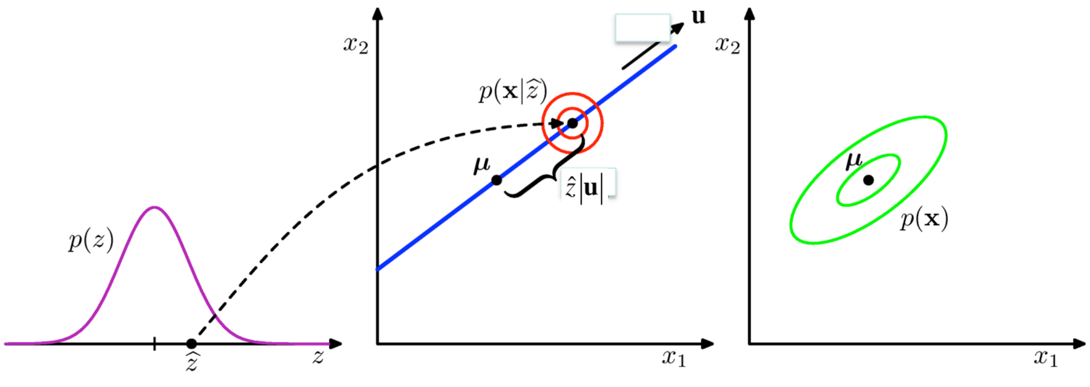
z is discrete



z is continuous

Comment: we can preprocess the data so that the mean is **0** to simplify the model





Finding the Principle Components

- There are different algorithms that can be used
  - EM (Roweis, NIPS 2007)
  - Can also be turned into an eigenvector computation (next)

Outline

- PCA
  - Example (Bishop, ch 12)
  - PCA as a mixture model variant
    - With a continuous latent variable
  - Breaking down PCA
    - Optimization problem
    - Solution
    - Intuition

The PCA Problem (vectors)

Start with a zero-mean dataset, where $\mathbf{x}^t$ is a the *t*-th instance:

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_m^1 \\ \vdots & \ddots & & \vdots \\ x_1^n & \dots & & x_m^n \end{bmatrix} = \begin{bmatrix} \vdots \\ - \mathbf{x}^t - \\ \vdots \end{bmatrix}$$

We want to find small number of *orthogonal* prototypes $\mathbf{u}_1, ..\mathbf{u}_k$
and *k* weights $z^t_1, \dots, z^t_k$ for each instance x^t so that if we approximate x^t by

$$\hat{\mathbf{x}}^t = \sum_{i=1}^{k} z_k^t \mathbf{u}_k$$

the approximation error will be small: we want to find **u**'s and ***z***'s to minimize

$$J = \frac{1}{N} \sum_{t=1}^{N} \|\mathbf{x}^t - \hat{\mathbf{x}}^t\|^2$$

The PCA Problem (matrices)

Given a zero-mean dataset
$$X = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_m^1 \\ \vdots & \ddots & & \vdots \\ x_1^n & \dots & & x_m^n \end{bmatrix} = \begin{bmatrix} \vdots \\ - \mathbf{x}^t - \\ \vdots \end{bmatrix}$$

Find factors U and Z so that X is approximately their outer product:

$$\begin{bmatrix} & | & \\ \dots & \mathbf{z}^t & \dots \\ & | & \end{bmatrix} \begin{bmatrix} & \vdots & \\ - - & \mathbf{u}_k & - - \\ & \vdots & \end{bmatrix} = \begin{bmatrix} \vdots \\ - \hat{\mathbf{x}}^t - \\ \vdots \end{bmatrix} = \hat{X}$$

$$\hat{\mathbf{x}}^t = \sum_{i=1}^{k} z_k^t \mathbf{u}_k$$

Specifically minimizing the square of the reconstruction error

$$J = \frac{1}{N} \sum_{t=1}^{N} \|\mathbf{x}^t - \hat{\mathbf{x}}^t\|^2$$

under the constraint that the rows of U are *orthogonal*.

A PCA Algorithm

Start with a zero-mean dataset, where

- $\mathbf{x}^t$ is a the *t*-th instance
- $\mathbf{f}_i$ is a column of feature values for the *i*-th feature.
- Compute the sample covariance matrix

$$C_X = X^T X$$

i.e.,

$$C_X(i,j) = \sum_t f_i^t f_j^t$$

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_m^1 \\ \vdots & \ddots & & \vdots \\ x_1^n & \dots & & x_m^n \end{bmatrix} = \begin{bmatrix} \vdots \\ - \mathbf{x}^t - \\ \vdots \end{bmatrix}$$

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_m^1 \\ \vdots & \ddots & & \vdots \\ x_1^n & \dots & & x_m^n \end{bmatrix} = \begin{bmatrix} & | & \\ \dots & \mathbf{f}_i & \dots \\ & | & \end{bmatrix}$$

- Find the largest *k* eigenvectors of C_X. These are the prototypes, U.
- Now find Z given X and U.

PCA Algorithm: Intuitions

Start with a zero-mean dataset, where

- $\mathbf{x}^t$ is a the *t*-th instance
- $\mathbf{f}_i$ is a column of feature values for the *i*-th feature.
- Compute the sample covariance matrix

$$C_X = X^T X$$

Some intuitions:

1. Suppose you wanted to predict feature *i* from feature *j*. Your best guess would be

$$f_i \text{ is predicted as } C_X(i,j) \cdot f_j$$

2. If you wanted to predict feature *i* from *all other* feature's *j,* a plausible guess is

$$f_i \text{ is predicted as } \frac{1}{n} \sum_{j \neq i} C_X(i,j) \cdot f_j$$

3. Any *eigenvector,* ***e,*** of C_X leads to an *internally consistent** set of predictions

* up to a multiplier

$$\exists \lambda : \lambda \mathbf{e} = C_X \mathbf{e} \Rightarrow \forall i, \lambda e_i = \frac{1}{n} \sum_j C_X(i,j) e_j$$

PCA: Eigenfaces

Turk and Pentland, 1991

PCA: Eigenfaces

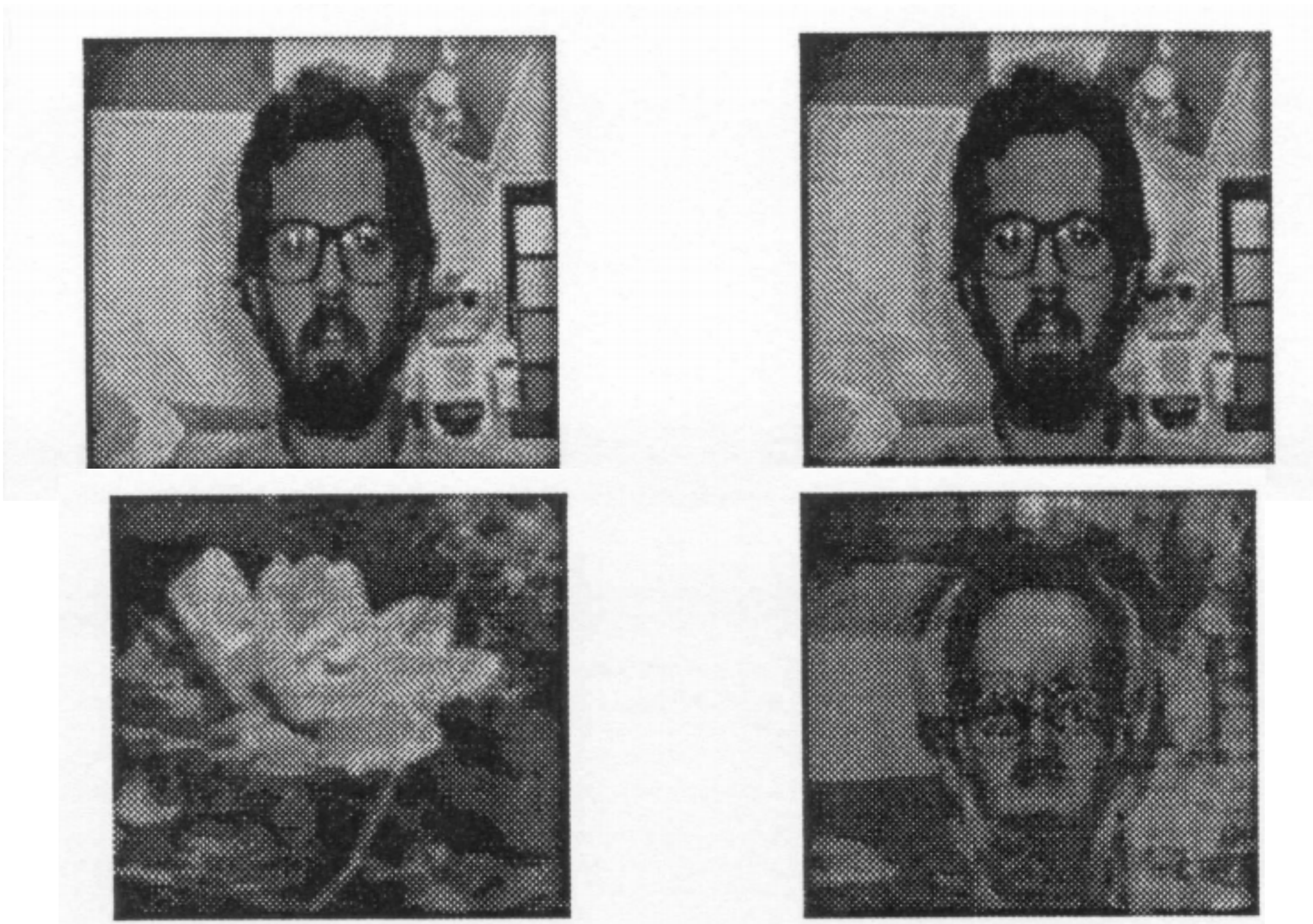
Turk and Pentland, 1991

Average face

Six eigenfaces (PC's)

PCA: Eigenfaces

Turk and Pentland, 1991

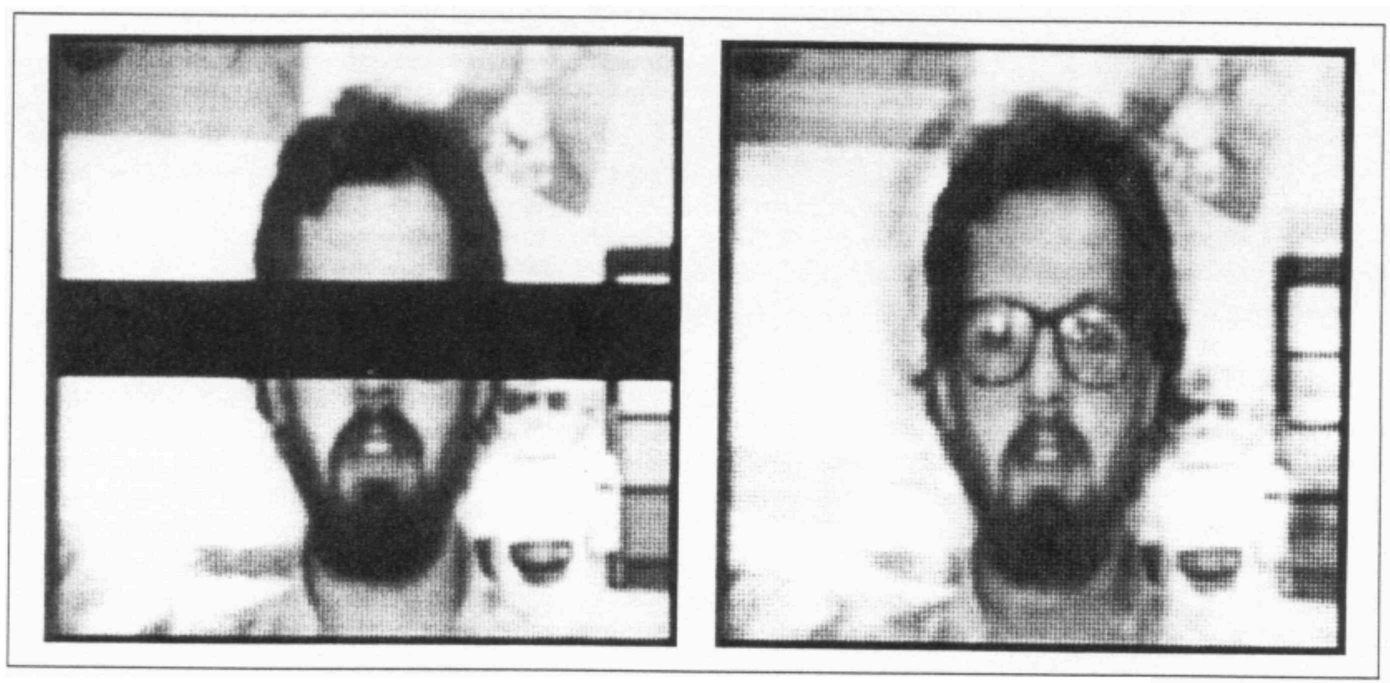


PCA: Eigenfaces

PCA: Eigenfaces

How is this done?

Simplest approach:
- Add the image with missing values to the data matrix
- Minimize reconstruction error over the non-missing values



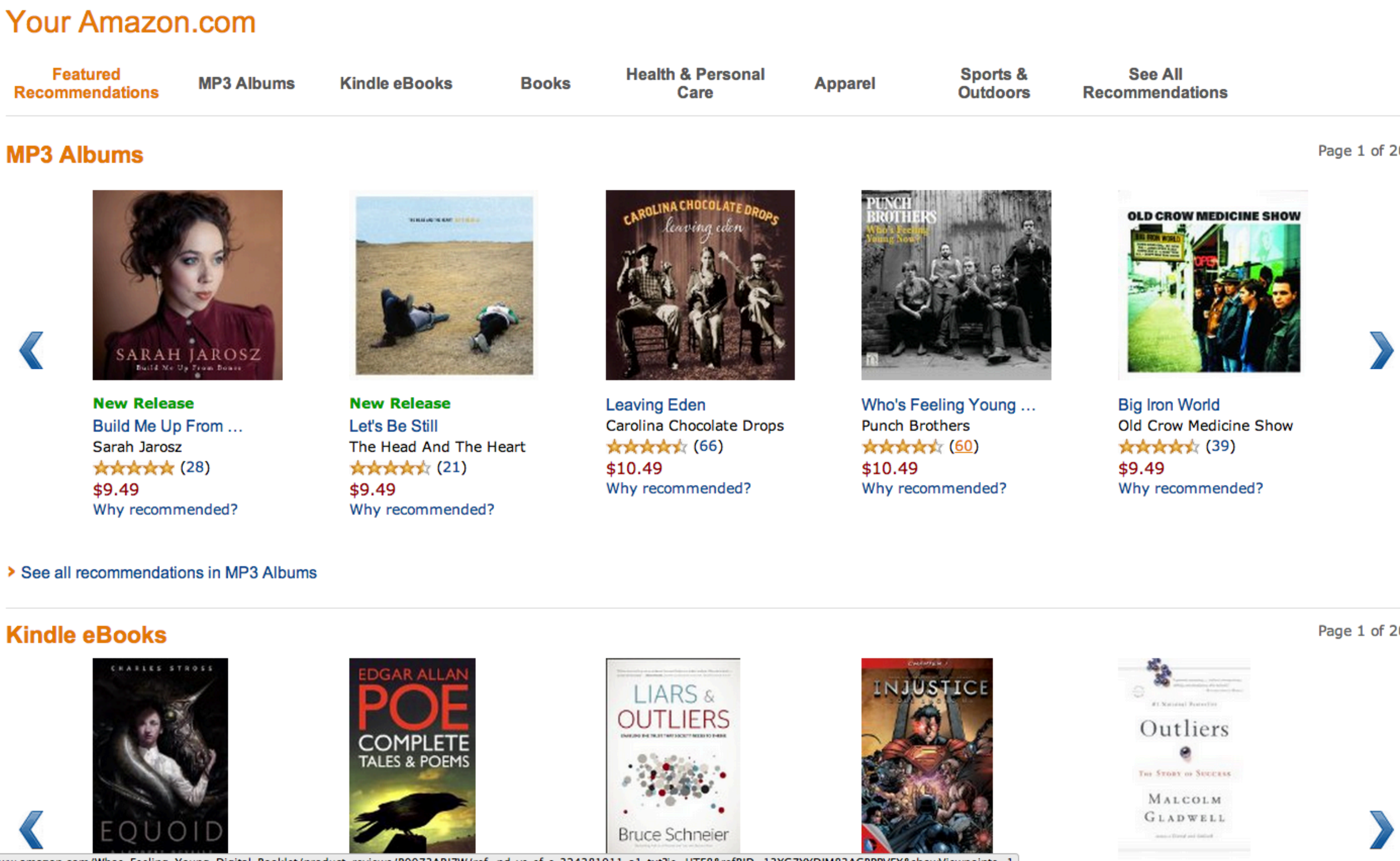
$$\begin{bmatrix} & | & \\ \cdots & \mathbf{z}^t & \cdots \\ & | & \end{bmatrix} \begin{bmatrix} & \vdots & \\ - \; - & \mathbf{u}_k & - \; - \\ & \vdots & \end{bmatrix} = \begin{bmatrix} \vdots \\ - \; \hat{\mathbf{x}}^t \; - \\ - \; ? \; - \\ \vdots \end{bmatrix} = \hat{X}$$

Matrix completion for image denoising

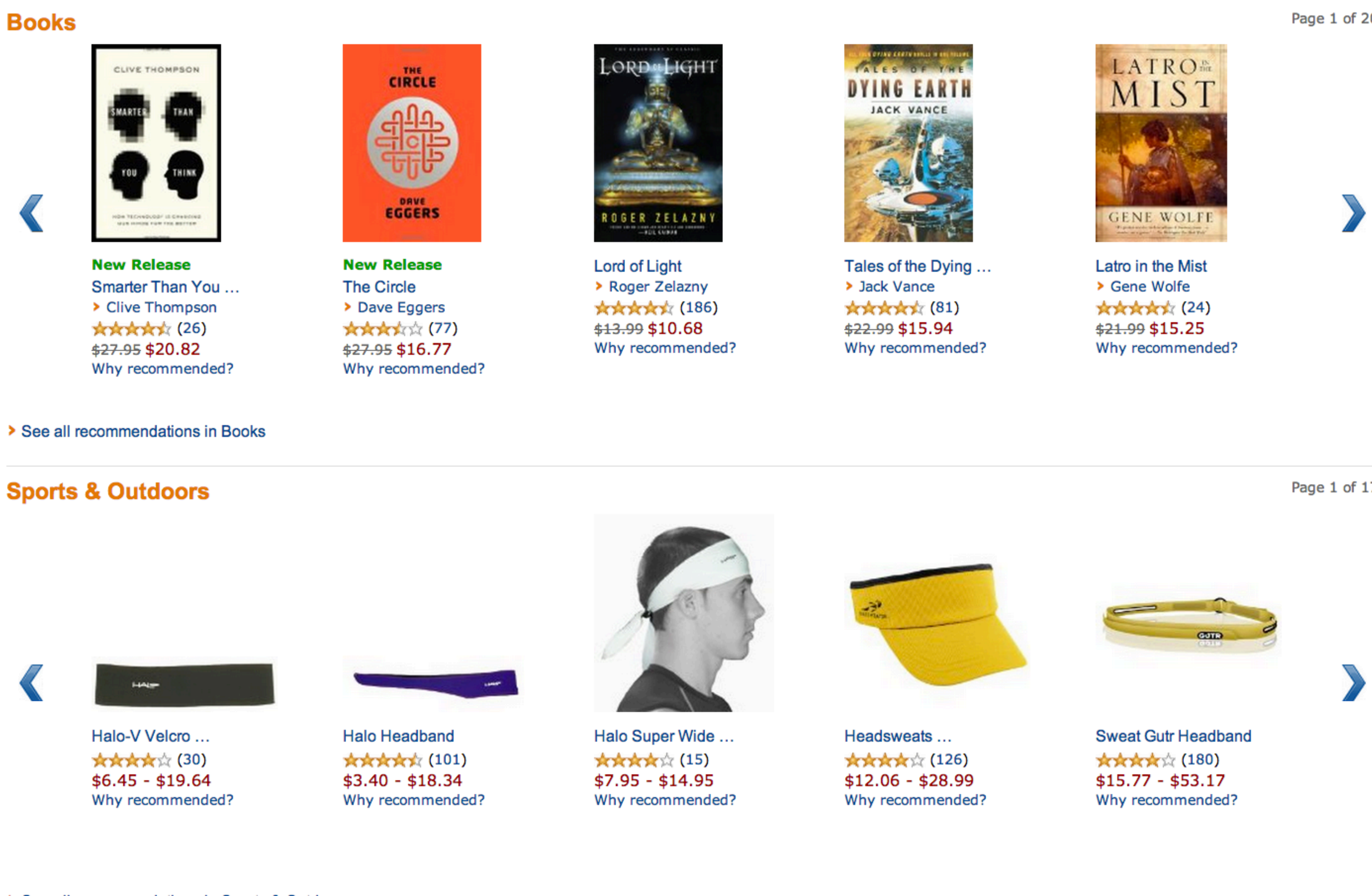
Outline

- Principle Compoments Analysis (PCA)
- Other types of/applications of matrix factorization
  - Collaborative filtering/recommendation
  - Matrix factorization for CF using gradient descent

What is collaborative filtering?



What is collaborative filtering?



What is collaborative filtering?

Your Amazon.com › **Improve Your Recommendations**
(If you're not William Cohen, click here.)

Help us make better recommendations. You can refine your recommendations by rating items or adjusting the checkboxes.

EDIT YOUR COLLECTION
- **Items you've purchased**
- Instant videos you've watched
- Items you've marked "I own it"
- Items you've rated
- Items you've liked
- Items you've marked "Not interested"
- Items you've marked as gifts

EDIT YOUR PREFERENCES
- [x] Show Amazon book recommendations as Kindle editions when possible.

Need Help?
Visit our help area to learn more.

Items you've purchased

Your Rating:

1.

Love Is Strange (A Paranormal Romance)
by Bruce Sterling
Your tags: [Add] (What's this?)
Click to Add: paranormal romance, nerd, futurist, science fiction romance, science fiction, technology, scifi, literature

- [ ] This was a gift
- [ ] Don't use for recommendations

2.

Mad Magazine #1
by Harvey Kurtzman
Your tags: [Add] (What's this?)
Click to Add: harvey kurtzman, dc

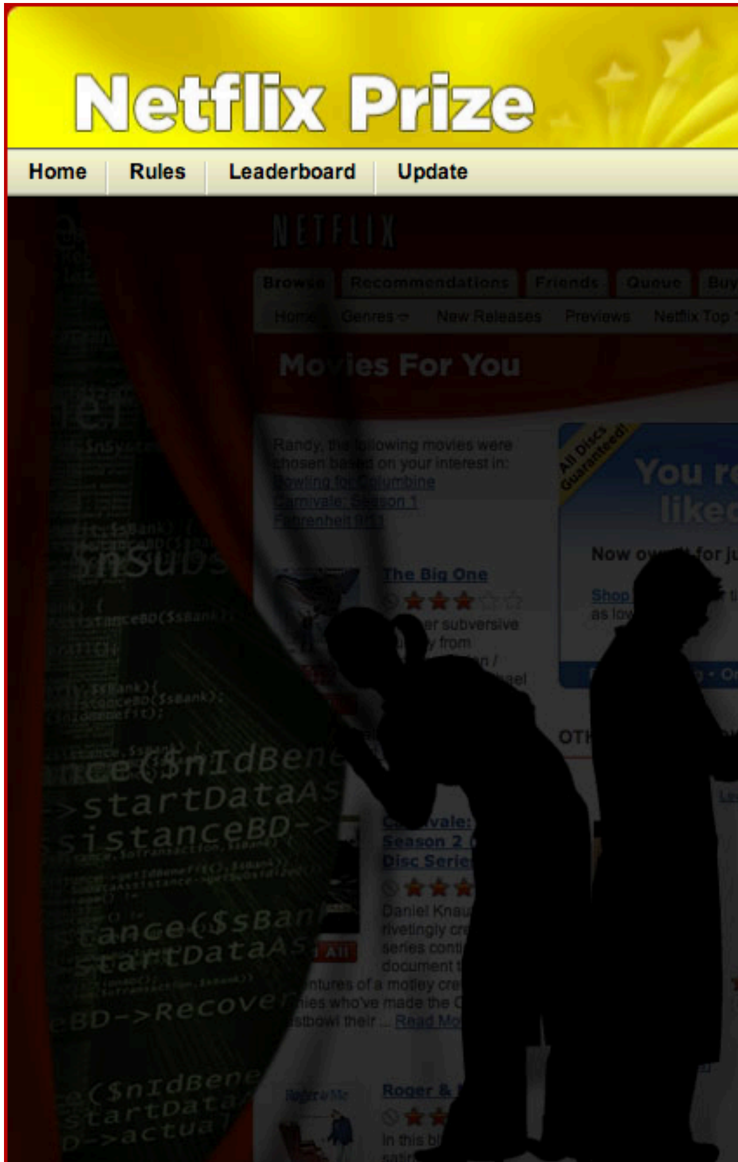
- [ ] This was a gift
- [ ] Don't use for recommendations

3.

Ahoy!
Punch Brothers | Format: MP3 Music
Your tags: [Add] (What's this?)
Click to Add: bluegrass, music, punch brothers, singer-songwriters

- [ ] This was a gift
- [ ] Don't use for recommendations

What is collabo



Congratulations!

The Netflix Prize sought to substantially improve the accuracy of predictions about how much someone is going to enjoy a movie based on their movie preferences.

On September 21, 2009 we awarded the $1M Grand Prize to team “BellKor’s Pragmatic Chaos”. Read about their algorithm, checkout team scores on the Leaderboard, and join the discussions on the Forum.

We applaud all the contributors to this quest, which improves our ability to connect people to the movies they love.

Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 **leaders.**

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8642	9.27	2009-07-15 14:53:22
14	Gravity	0.8643	9.26	2009-04-22 18:31:32
15	Ces	0.8651	9.18	2009-06-21 19:24:53
16	Invisible Ideas	0.8653	9.15	2009-07-15 15:53:04
17	Just a guy in a garage	0.8662	9.06	2009-05-24 10:02:54
18	J Dennis Su	0.8666	9.02	2009-03-07 17:16:17
19	Craig Carmichael	0.8666	9.02	2009-07-25 16:00:54
20	acmehill	0.8668	9.00	2009-03-21 16:20:50
Progress Prize 2007 - RMSE = 0.8723 - Winning Team: KorBell				
Cinematch score - RMSE = 0.9525				

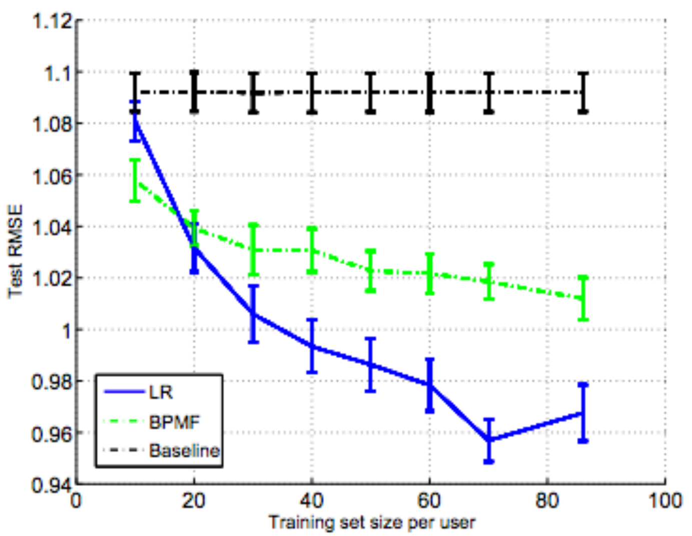
What is collaborative filtering?

A Framework for Optimizing Paper Matching

Laurent Charlin
Department of Computer Science
University of Toronto
Toronto, ON M5S 3H5
laurent@cs.toronto.edu

Richard Zemel
Department of Computer Science
University of Toronto
Toronto, ON M5S 3H5
zemel@cs.toronto.edu

Craig Boutilier
Department of Computer Science
University of Toronto
Toronto, ON M5S 3H5
cebly@cs.toronto.edu



Other examples of social filtering….

cmu

Web   Images   Maps   Shopping   News   More   Search tools

About 24,700,000 results (0.24 seconds)

Carnegie Mellon University | CMU
www.cmu.edu/
Carnegie Mellon University (**CMU**) is a global research university recognized for world-class arts and technology programs, collaboration across disciplines and ...
4.7 ★★★★★ 105 Google reviews · Write a review

5000 Forbes Ave Pittsburgh, PA 15213
(412) 268-2000

Careers
Connect to Careers@CarnegieMellon to view and ...

Prospective Students
Admissions - Graduate Education - International Students - ...

Academics
Schools & Colleges - Graduate Admission - College of Fine Arts

Contact Us
Skip navigation and jump directly to page content. Contact Us ...

Carnegie Mellon Athletics
Fitness & Recreation - Staff Directory - Men's Track and Field

Directory
Public information for all university computing accounts.

Search cmu.edu

News for cmu

CMU president Subra Suresh elected to Institute of Medicine
Pittsburgh Post Gazette - 1 day ago
Mr. Suresh, who became president of **CMU** in July, was elected to the Institute of Medicine today in recognition of his research into cell ...

CMU ex-trustee's money laundering trial begins
Pittsburgh Post Gazette - 1 day ago

Carnegie Mellon University - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Carnegie_Mellon**_University
Carnegie Mellon has seven colleges and independent schools: the Carnegie Institute of Technology (engineering), College of Fine Arts, Dietrich College of ...

Central Michigan University: Home
www.cmich.edu/
Students are offered educational experiences in the arts, humanities, and natural and social sciences, in addition to educational depth in at least one academic ...

Complete Music Update
www.completemusicupdate.com/
An UnLimited Media Website, **Complete Music Update** | ThisWeek London | ThreeWeeks Edinburgh ... NEXT **CMU** INSIGHTS COURSE | NOW BOOKING

People also search for

University of Pittsburgh — Pittsburgh
Cornell University — Ithaca
Massach… Institute of Technology — Cambridge
Carnegie Mellon School of… — Pittsburgh
Stanford University — Stanford

Other examples of social filtering….

COMBINED PRINT & E-BOOK FICTION

1. **STORM FRONT,** by John Sandford
2. **DOING HARD TIME,** by Stuart Woods
3. **DOCTOR SLEEP,** by Stephen King
4. **THE HUSBAND'S SECRET,** by Liane Moriarty
5. **THE LONGEST RIDE,** by Nicholas Sparks

Complete List »

COMBINED PRINT & E-BOOK NONFICTION

1. **KILLING JESUS,** by Bill O'Reilly and Martin Dugard
2. **MY STORY,** by Elizabeth Smart with Chris Stewart
3. **DAVID AND GOLIATH,** by Malcolm Gladwell
4. **I AM MALALA,** by Malala Yousafzai with Christina Lamb
5. **THE REASON I JUMP,** by Naoki Higashida

Complete List »

HARDCOVER FICTION

1. **STORM FRONT,** by John Sandford
2. **DOCTOR SLEEP,** by Stephen King
3. **THE LONGEST RIDE,** by Nicholas Sparks
4. **GONE,** by James Patterson and Michael Ledwidge
5. **DOG SONGS,** by Mary Oliver

Complete List »

HARDCOVER NONFICTION

1. **KILLING JESUS,** by Bill O'Reilly and Martin Dugard
2. **DAVID AND GOLIATH,** by Malcolm Gladwell
3. **THE REASON I JUMP,** by Naoki Higashida
4. **I AM MALALA,** by Malala Yousafzai with Christina Lamb
5. **MY STORY,** by Elizabeth Smart with Chris Stewart

October 26, 2013

80.916 k 19.815 k 1.1 k

The Hot 100

The week's most popular current songs across …

1 – 10 11 – 20 21 – 30 31 – 40 … 91 – 100 Next chart Play entire chc

1 –

Royals
Lorde
Love Club

1 1 15

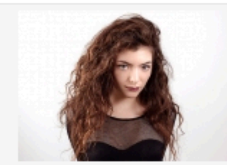
Listen
Watch
Buy
Send Ringtone

0 0 0 0

Lorde's 'Royals' Rules Hot 100 For Third Week

Lorde's "Royals" commands the Billboard Hot 100 for a third week, Ylvis' "The Fox" darts to the Digital Songs top 10 and Justin Bieber and Eminem soar into the Hot 100's top 20.

As we do every Wednesday, let's break down the Hot 100's top 10 and more.



2

Wrecking Ball
Miley Cyrus
Bangerz

Listen
Watch
Buy

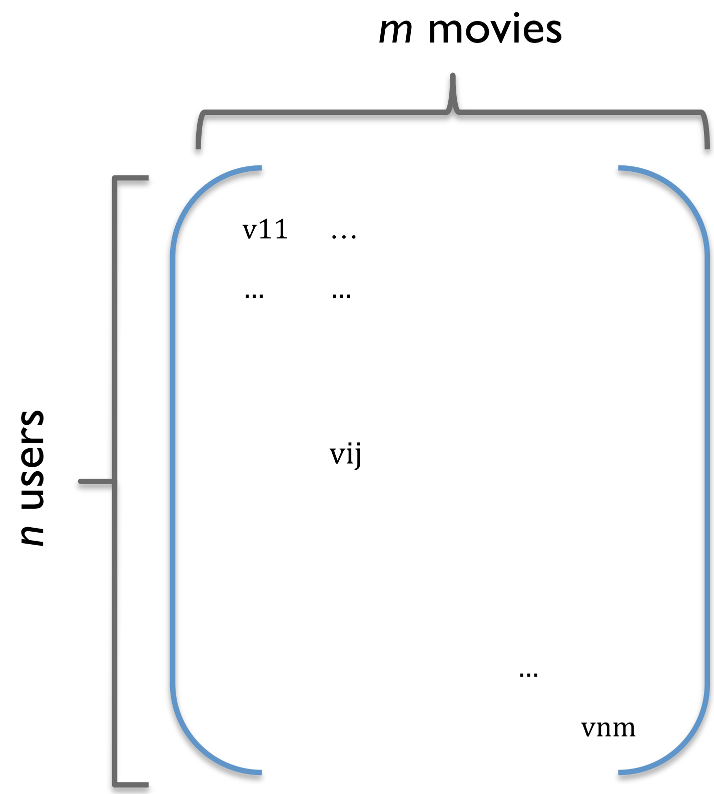
Everyday Examples of Collaborative Filtering...

- Bestseller lists
- Top 40 music lists
- The “recent returns” shelf at the library
- Unmarked but well-used paths thru the woods
- The printer room at work
- “Read any good books lately?”
- ....
- **Common insight**: personal tastes are *correlated*:
  - If Alice and Bob both like X and Alice likes Y then Bob is more likely to like Y
  - especially (perhaps) if Bob knows Alice

Outline

- Principle Compoments Analysis (PCA)
- Other types of/applications of matrix factorization
  – Collaborative filtering/recommendation
  – Algorithms:
    - K-NN type methods
    - Classification-base methods
    - …
  – Matrix factorization

Recovering latent factors in a matrix



V[i,j] = user i's rating of movie j

Recovering latent factors in a matrix

$$\left.\begin{matrix} \\ \\ n\text{ users} \\ \\ \\ \end{matrix}\right\{ \begin{pmatrix} x1 & y1 \\ x2 & y2 \\ .. & .. \\ \\ \\ ... & ... \\ xn & yn \end{pmatrix} \times \overbrace{\begin{pmatrix} a1 & a2 & .. & ... & am \\ b1 & b2 & ... & ... & bm \end{pmatrix}}^{m\text{ movies}} \approx \overbrace{\begin{pmatrix} v11 & ... & & & \\ ... & ... & & & \\ & & & & \\ & vij & & & \\ & & & & \\ & & & ... & \\ & & & & vnm \end{pmatrix}}^{m\text{ movies}}$$

Minimize squared error reconstruction error and force the "prototype" users to be orthogonal ➔ PCA

V[i,j] = user i's rating of movie j

KDD 2011

Large-Scale Matrix Factorization with Distributed Stochastic Gradient Descent

Rainer Gemulla

talk pilfered from →

Peter J. Haas   Yannis Sismanis   Erik Nijkamp

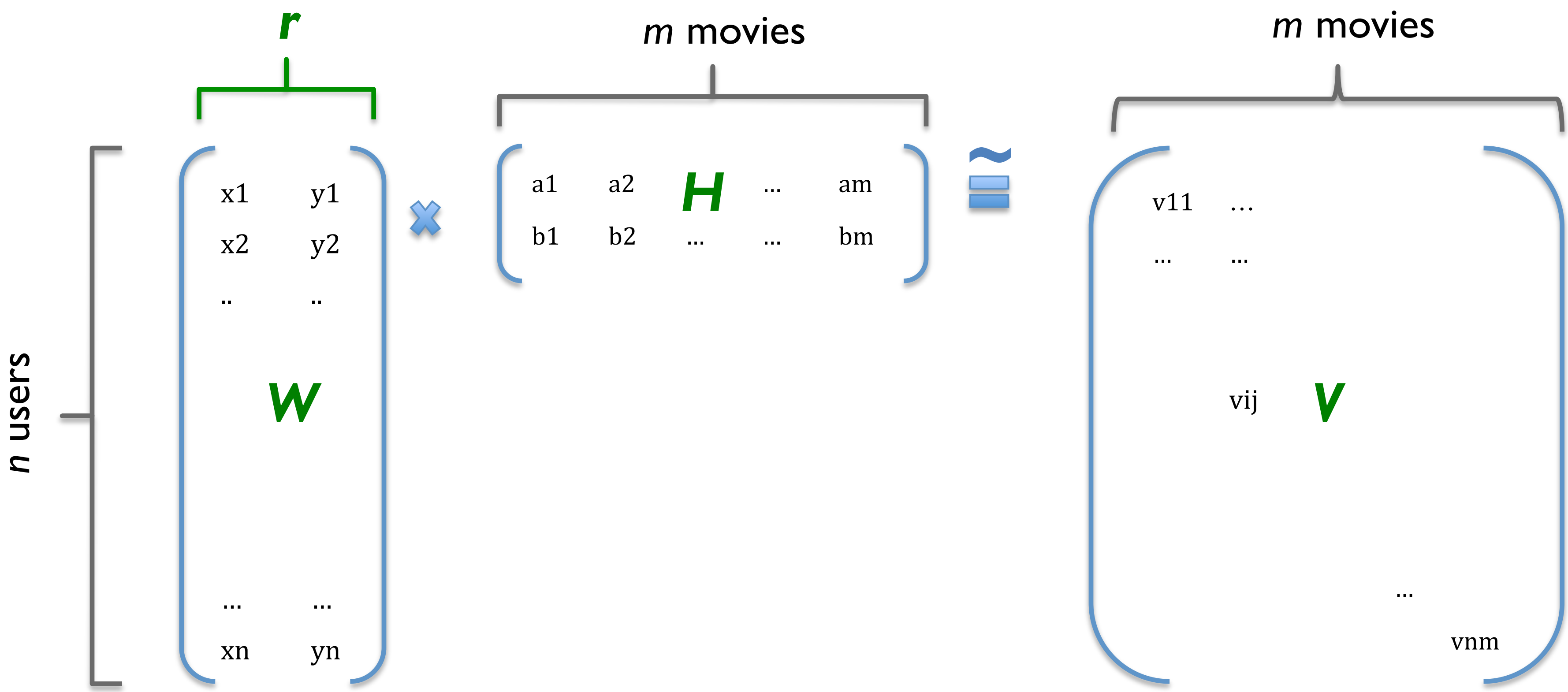
Collaborative Filtering

- Problem
  - Set of users
  - Set of items (movies, books, jokes, products, stories, ...)
  - Feedback (ratings, purchase, click-through, tags, ...)
- Predict additional items a user may like
  - Assumption: Similar feedback $\Longrightarrow$ Similar taste
- Example

$$\begin{array}{cc} & \begin{array}{ccc} \textit{Avatar} & \textit{The Matrix} & \textit{Up} \end{array} \\ \begin{array}{l} \textit{Alice} \\ \textit{Bob} \\ \textit{Charlie} \end{array} & \left(\begin{array}{ccc} ? & 4 & 2 \\ 3 & 2 & ? \\ 5 & ? & 3 \end{array}\right) \end{array}$$

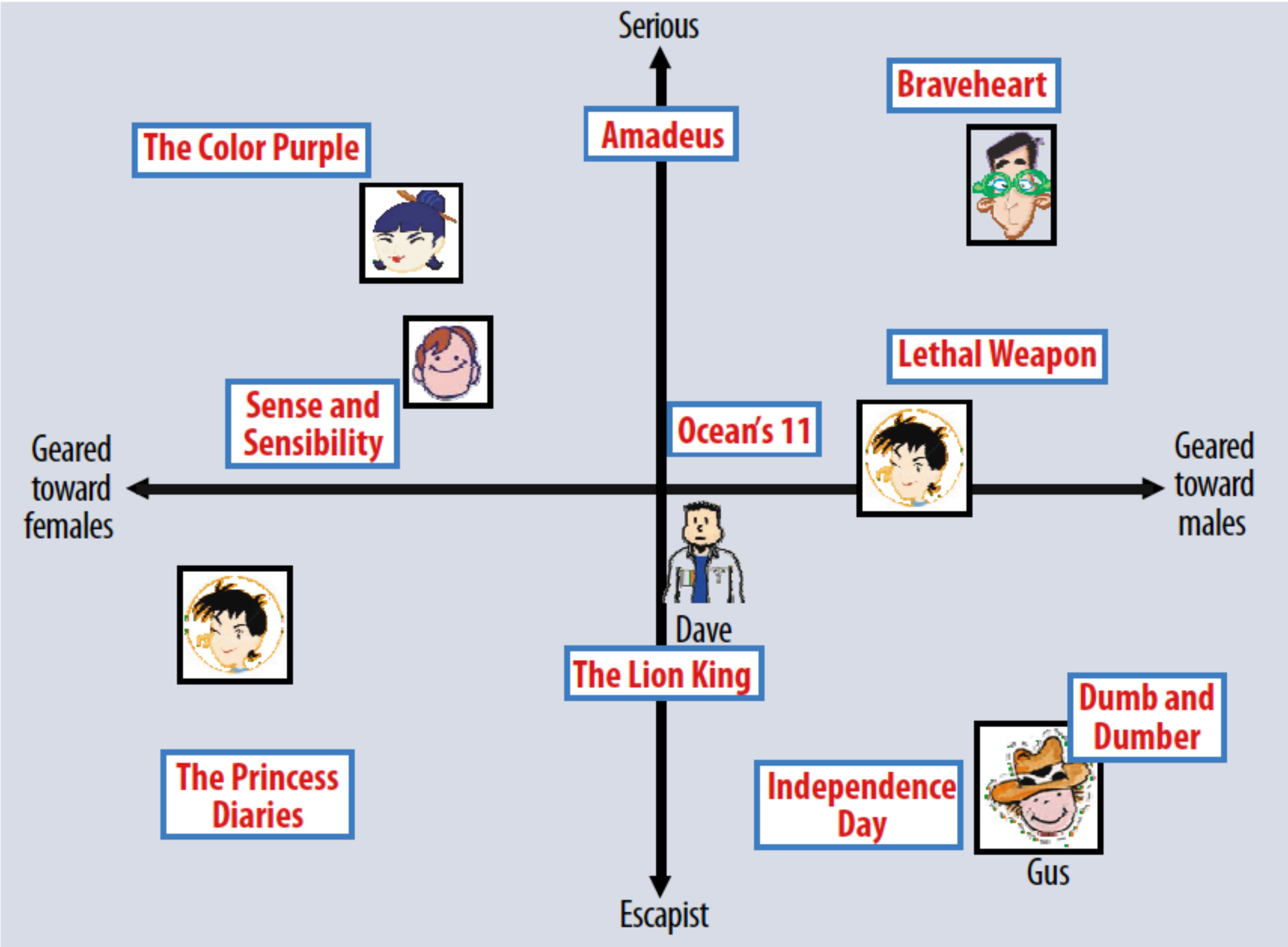
- Netflix competition: 500k users, 20k movies, 100M movie ratings, 3M question marks

Recovering latent factors in a matrix



V[i,j] = user i's rating of movie j

Semantic Factors (Koren et al., 2009)



Latent Factor Models

- Discover latent factors ($r = 1$)

	Avatar (2.24)	**The Matrix** (1.92)	**Up** (1.18)
Alice (1.98)	? (4.4)	**4** (3.8)	**2** (2.3)
Bob (1.21)	**3** (2.7)	**2** (2.3)	? (1.4)
Charlie (2.30)	**5** (5.2)	? (4.4)	**3** (2.7)

- Minimum loss

$$\min_{\mathbf{W},\mathbf{H},\mathbf{u},\mathbf{m}} \sum_{(i,j)\in Z} (\mathbf{V}_{ij} - \mu - \mathbf{u}_i - \mathbf{m}_j - [\mathbf{WH}]_{ij})^2$$

$$+ \lambda \left(\|\mathbf{W}\| + \|\mathbf{H}\| + \|\mathbf{u}\| + \|\mathbf{m}\| \right)$$

user-specific bias term

movie-specific bias term

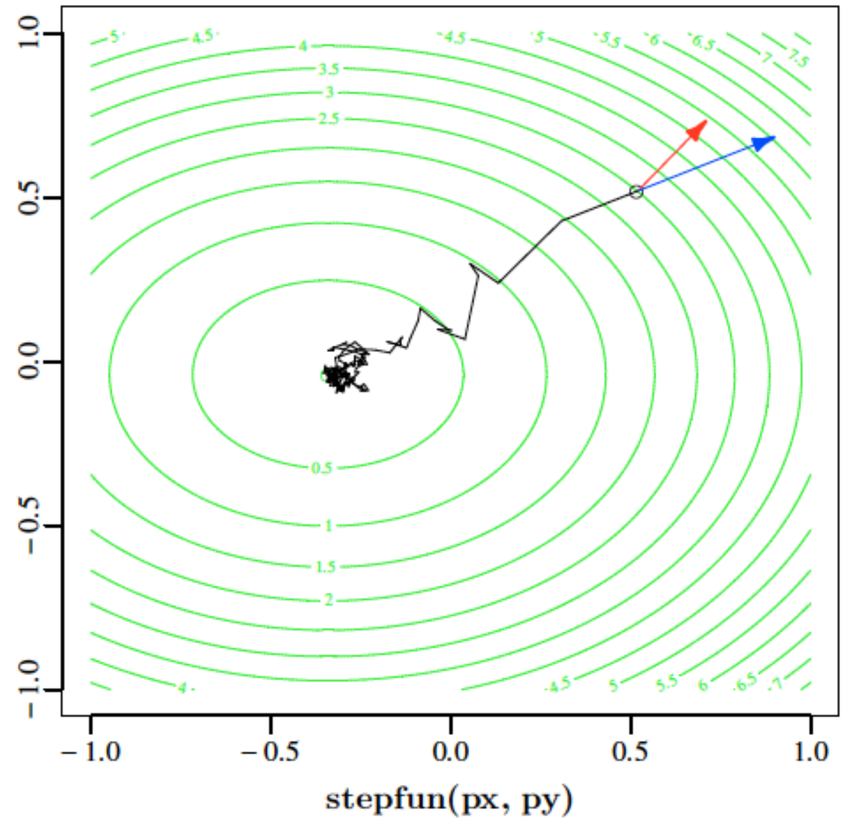
- Bias, regularization

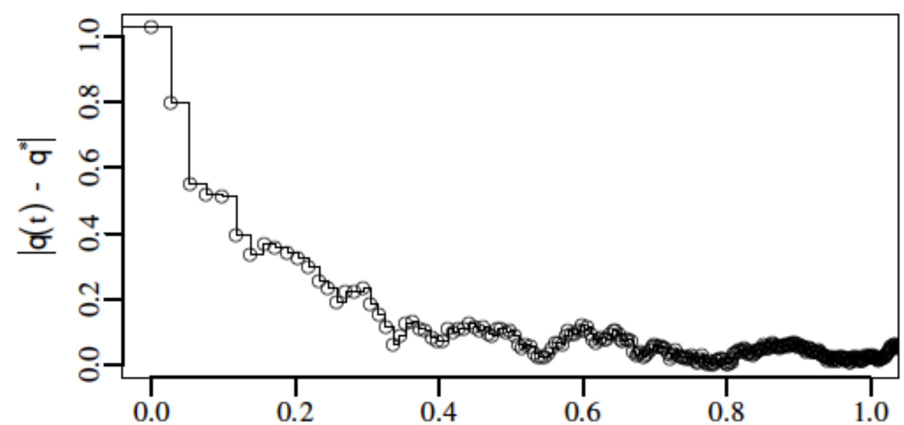
Stochastic Gradient Descent

- Find minimum θ^* of function L
- Pick a starting point θ_0
- Approximate gradient $\hat{L}'(\theta_0)$
- Jump "approximately" downhill
- Stochastic difference equation

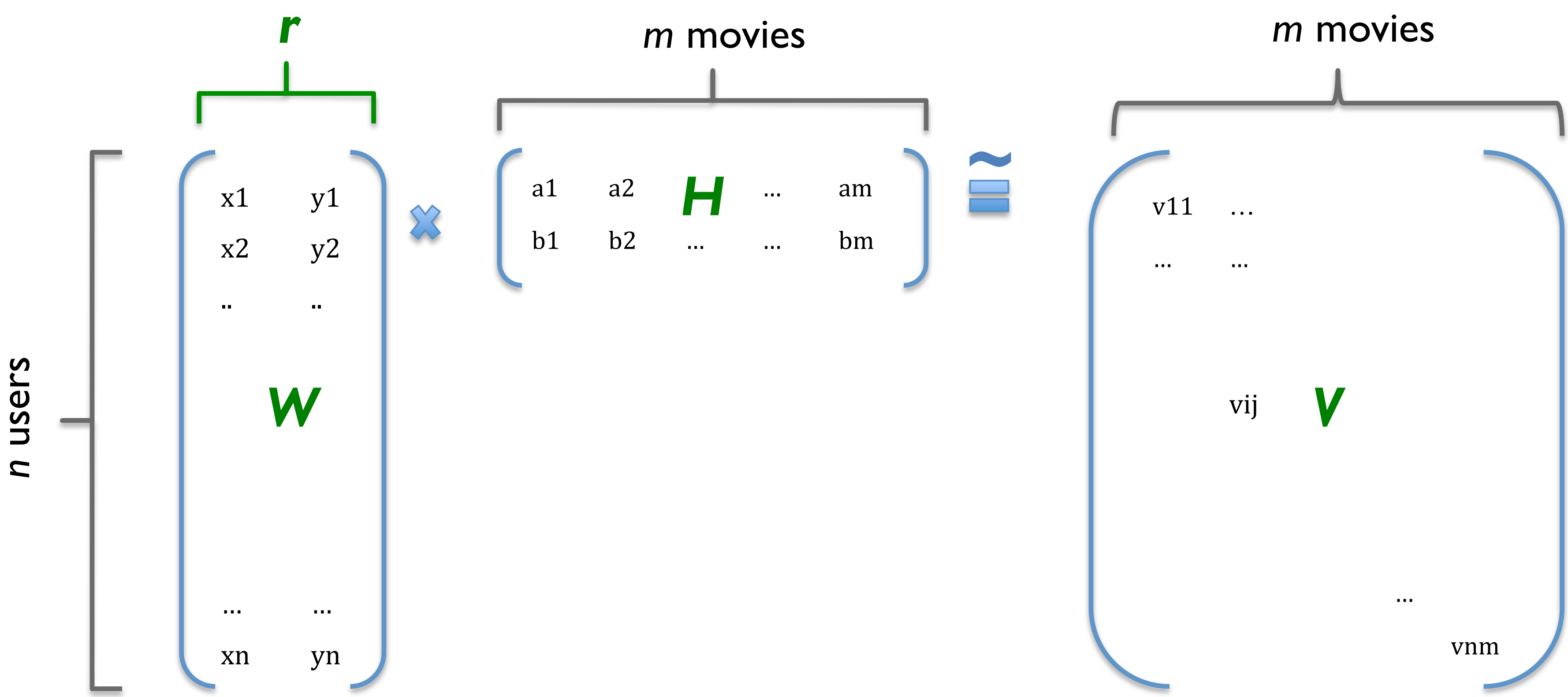
$$\theta_{n+1} = \theta_n - \epsilon_n \hat{L}'(\theta_n)$$

- Under certain conditions, asymptotically approximates (continuous) gradient descent





Recovering latent factors in a matrix



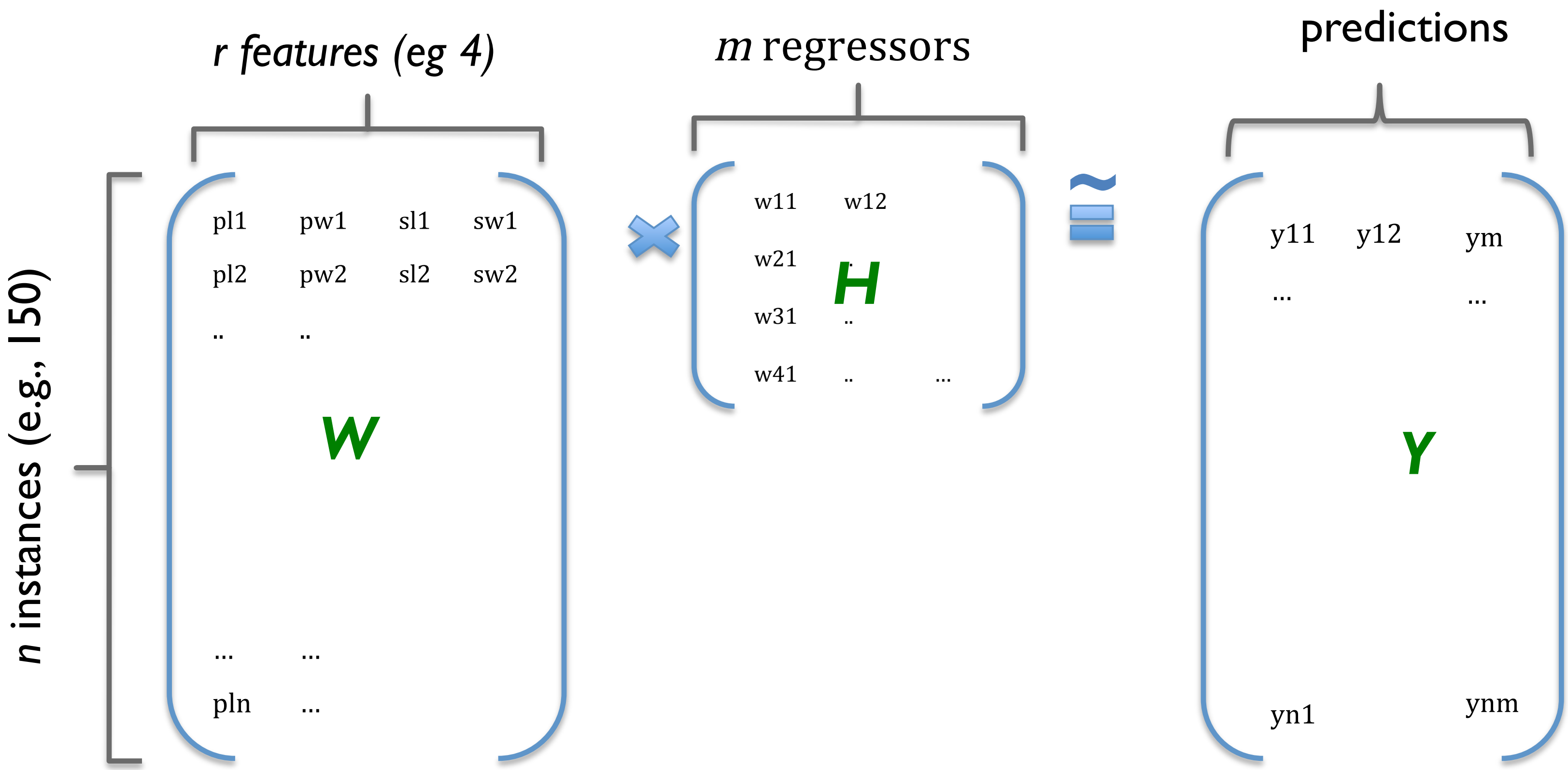
V[i,j] = user i's rating of movie j

… is like Linear Regression ….

$$
\begin{array}{c}
\\
n \text{ instances (e.g., 150)}
\end{array}
\overbrace{\begin{pmatrix}
pl1 & pw1 & sl1 & sw1 \\
pl2 & pw2 & sl2 & sw2 \\
.. & .. & & \\
 & W & & \\
... & ... & & \\
pln & pwn & &
\end{pmatrix}}^{r \text{ features (eg 4)}}
\times
\overbrace{\begin{pmatrix}
w1 \\ w2 \\ w3 \\ w4
\end{pmatrix}}^{m=1 \text{ regressors}}
H
\approx
\overbrace{\begin{pmatrix}
y1 \\ ... \\ yi \\ yn
\end{pmatrix}}^{\text{predictions}}
Y
$$

Y[i,1] = instance i's prediction

.. for many outputs at once….



Y[I,j] = instance i's prediction for regression task j

… where we also have to find the dataset!

Matrix factorization as SGD

require that the loss can be written as

$$L = \sum_{(i,j)\in Z} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j})$$

Algorithm 1 SGD for Matrix Factorization

Require: A training set Z, initial values $\boldsymbol{W}_0$ and $\boldsymbol{H}_0$

while not converged **do** {step}

Select a training point $(i, j) \in Z$ uniformly at random.

$\boldsymbol{W}'_{i*} \leftarrow \boldsymbol{W}_{i*} - \epsilon_n N \frac{\partial}{\partial \boldsymbol{W}_{i*}} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j})$

$\boldsymbol{H}_{*j} \leftarrow \boldsymbol{H}_{*j} - \epsilon_n N \frac{\partial}{\partial \boldsymbol{H}_{*j}} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j})$

$\boldsymbol{W}_{i*} \leftarrow \boldsymbol{W}'_{i*}$

step size

end while

Matrix factorization as SGD - why does this work?

require that the loss can be written as

$$L = \sum_{(i,j)\in Z} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j})$$

Algorithm 1 SGD for Matrix Factorization

Require: A training set Z, initial values $\boldsymbol{W}_0$ and $\boldsymbol{H}_0$

while not converged **do** {step}

Select a training point $(i, j) \in Z$ uniformly at random.

$\boldsymbol{W}'_{i*} \leftarrow \boldsymbol{W}_{i*} - \epsilon_n N \frac{\partial}{\partial \boldsymbol{W}_{i*}} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j})$

$\boldsymbol{H}_{*j} \leftarrow \boldsymbol{H}_{*j} - \epsilon_n N \frac{\partial}{\partial \boldsymbol{H}_{*j}} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j})$

$\boldsymbol{W}_{i*} \leftarrow \boldsymbol{W}'_{i*}$

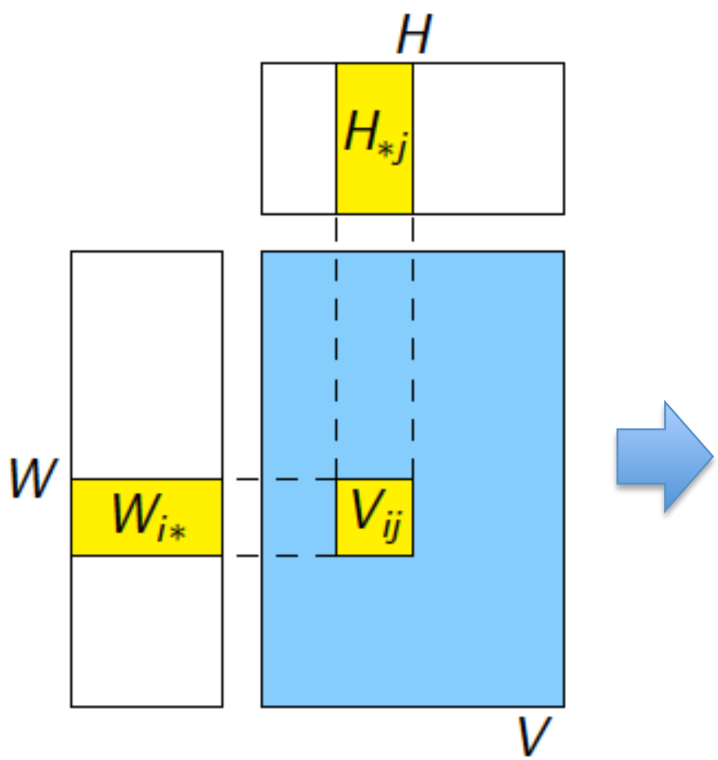
end while

step size

Matrix factorization as SGD - why does this work? Here's the key claim:

require that the loss can be written as

$$L = \sum_{(i,j)\in Z} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j})$$



$$\frac{\partial}{\partial \boldsymbol{W}_{i'k}} L_{ij}(\boldsymbol{W}, \boldsymbol{H}) = \begin{cases} 0 & \text{if } i \neq i' \\ \frac{\partial}{\partial \boldsymbol{W}_{ik}} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j}) & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial \boldsymbol{H}_{kj'}} L_{ij}(\boldsymbol{W}, \boldsymbol{H}) = \begin{cases} 0 & \text{if } j \neq j' \\ \frac{\partial}{\partial \boldsymbol{H}_{kj}} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j}) & \text{otherwise} \end{cases}$$

Checking the claim

$$\frac{\partial}{\partial \boldsymbol{W}_{i*}} L(\boldsymbol{W}, \boldsymbol{H}) = \frac{\partial}{\partial \boldsymbol{W}_{i*}} \sum_{(i',j) \in Z} L_{i'j}(\boldsymbol{W}_{i'*}, \boldsymbol{H}_{*j}) = \sum_{j \in Z_{i*}} \frac{\partial}{\partial \boldsymbol{W}_{i*}} L_{ij}(\boldsymbol{W}_{i*}, \boldsymbol{H}_{*j}),$$

where $Z_{i*} = \{\, j\colon (i,j) \in Z \,\}$.

$$\frac{\partial}{\partial \boldsymbol{H}_{*j}} L(\boldsymbol{W}, \boldsymbol{H}) = \sum_{i \in Z_{*j}} \frac{\partial}{\partial \boldsymbol{W}_{*j}} L_{ij}(\boldsymbol{W}_{i*}, \boldsymbol{H}_{*j}),$$

where $Z_{*j} = \{\, i\colon (i,j) \in Z \,\}$.

Think for SGD for logistic regression

- LR loss = compare *y* and $\hat{y}$ = dot(w,x)
- similar but now update w (user weights) and x (movie weight)

What loss functions are possible?

N1, N2 - diagonal matrixes, sort of like IDF factors for the users/ movies

$$L_{\mathrm{NZSL}} = \sum_{(i,j)\in Z} (\boldsymbol{V}_{ij} - [\boldsymbol{W}\boldsymbol{H}]_{ij})^2$$

$$L_{\mathrm{L2}} = L_{\mathrm{NZSL}} + \lambda\left(\|\boldsymbol{W}\|_{\mathrm{F}}^2 + \|\boldsymbol{H}\|_{\mathrm{F}}^2\right)$$

$$L_{\mathrm{NZL2}} = L_{\mathrm{NZSL}} + \lambda\left(\|\boldsymbol{N}_1\boldsymbol{W}\|_{\mathrm{F}}^2 + \|\boldsymbol{H}\boldsymbol{N}_2\|_{\mathrm{F}}^2\right)$$

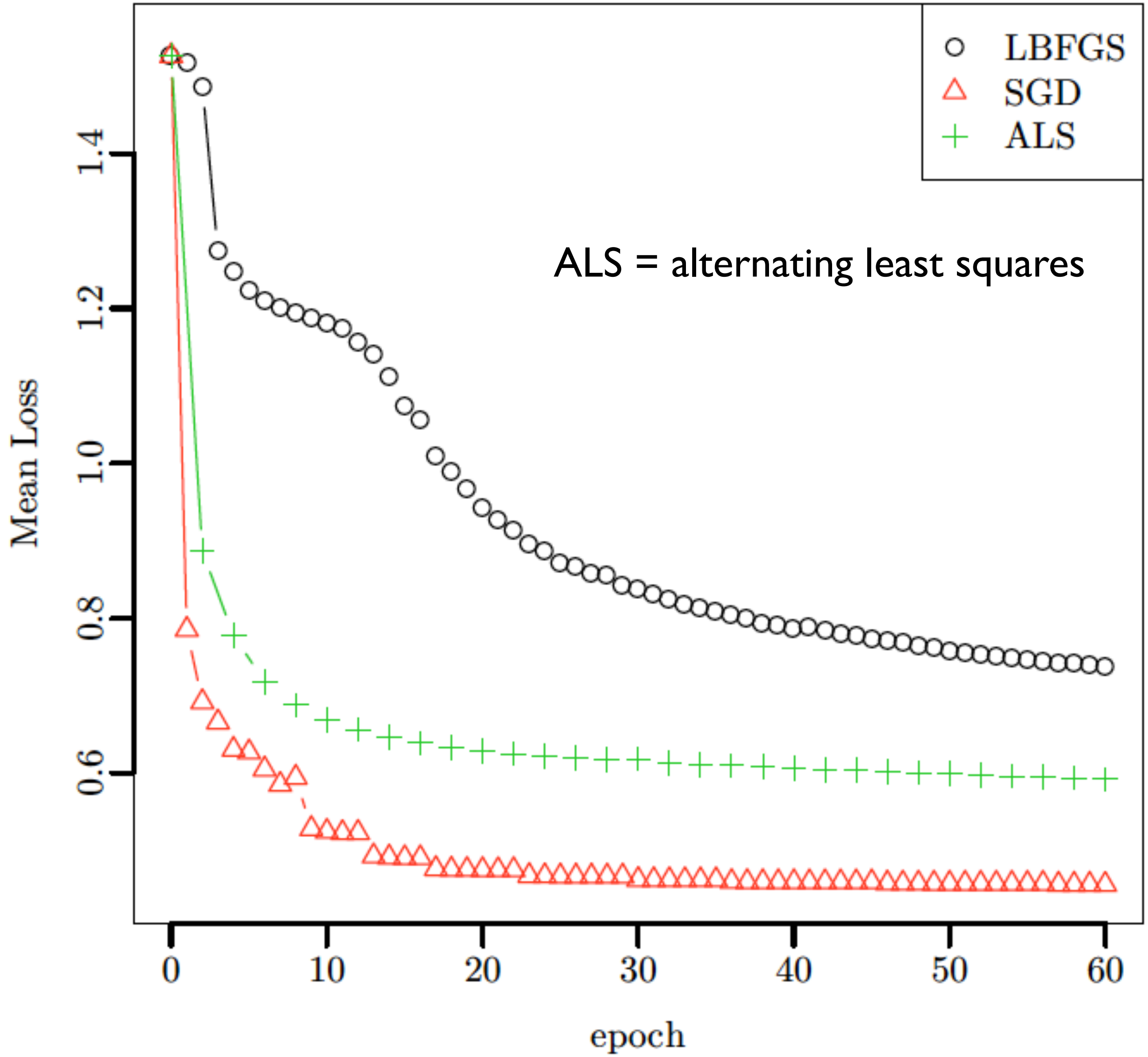
What loss functions are possible?

Loss Function	Definition and Derivatives
L_{NZSL}	$L_{\text{NZSL}} = \sum_{(i,j)\in Z} (\boldsymbol{V}_{ij} - [\boldsymbol{W}\boldsymbol{H}]_{ij})^2$
	$\frac{\partial}{\partial \boldsymbol{W}_{ik}} L_{ij} = -2(\boldsymbol{V}_{ij} - [\boldsymbol{W}\boldsymbol{H}]_{ij})\boldsymbol{H}_{kj}$ $\frac{\partial}{\partial \boldsymbol{H}_{kj}} L_{ij} = -2(\boldsymbol{V}_{ij} - [\boldsymbol{W}\boldsymbol{H}]_{ij})\boldsymbol{W}_{ik}$

What loss functions are possible?

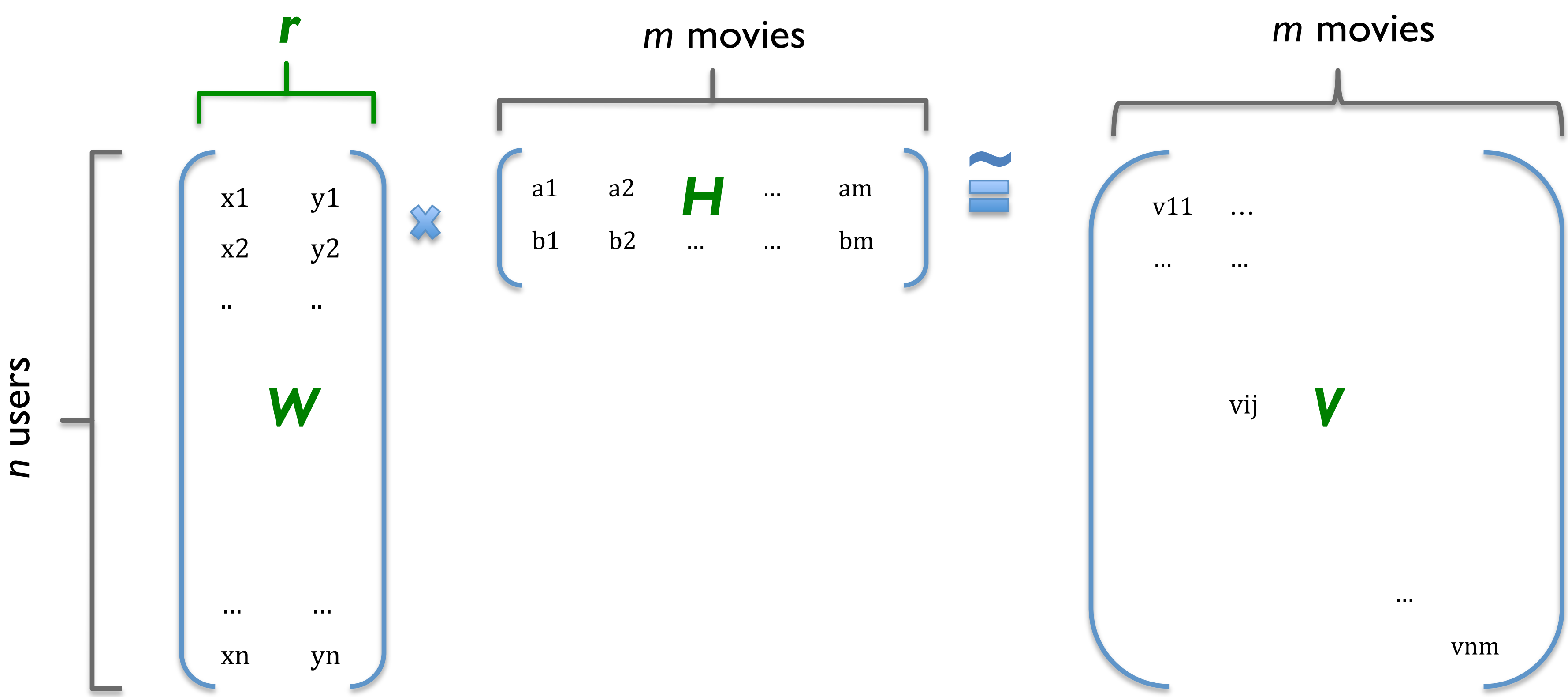
Loss Function	Definition and Derivatives
L_{L2}	$L_{\text{L2}} = L_{\text{NZSL}} + \lambda \left(\lVert \boldsymbol{W} \rVert_{\text{F}}^2 + \lVert \boldsymbol{H} \rVert_{\text{F}}^2 \right)$ $= \sum_{(i,j)\in Z} \left[(\boldsymbol{V}_{ij} - [\boldsymbol{W}\boldsymbol{H}]_{ij})^2 + \lambda \left(\frac{\lVert \boldsymbol{W}_{i*} \rVert_{\text{F}}^2}{N_{i*}} + \frac{\lVert \boldsymbol{H}_{*j} \rVert_{\text{F}}^2}{N_{*j}} \right) \right]$ $\frac{\partial}{\partial \boldsymbol{W}_{ik}} L_{ij} = -2(\boldsymbol{V}_{ij} - [\boldsymbol{W}\boldsymbol{H}]_{ij})\boldsymbol{H}_{kj} + 2\lambda \frac{\boldsymbol{W}_{ik}}{N_{i*}}$ $\frac{\partial}{\partial \boldsymbol{H}_{kj}} L_{ij} = -2(\boldsymbol{V}_{ij} - [\boldsymbol{W}\boldsymbol{H}]_{ij})\boldsymbol{W}_{ik} + 2\lambda \frac{\boldsymbol{H}_{kj}}{N_{*j}}$

Stochastic Gradient Descent on Netflix Data



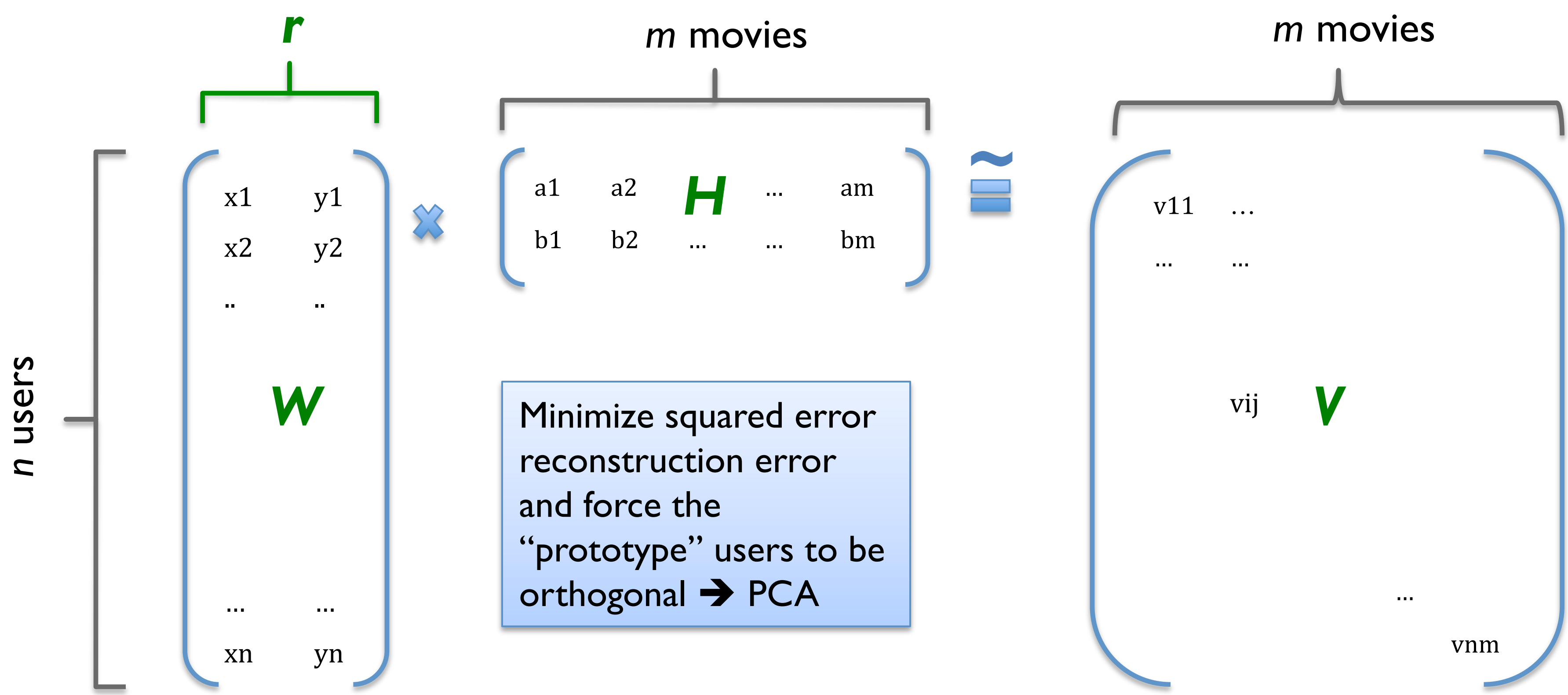
Wrapup: Matrix Multiplications in Machine Learning

Recovering latent factors in a matrix



V[i,j] = user i's rating of movie j

… vs PCA



Minimize squared error reconstruction error and force the "prototype" users to be orthogonal ➔ PCA

V[i,j] = user i's rating of movie j

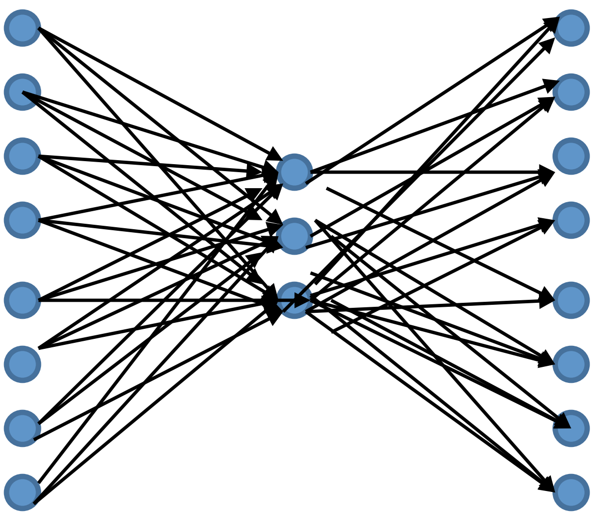
Flashback to NN lecture…..

… vs autoencoders & nonlinear PCA

- Assume we would like to learn the following (trivial?) output function:

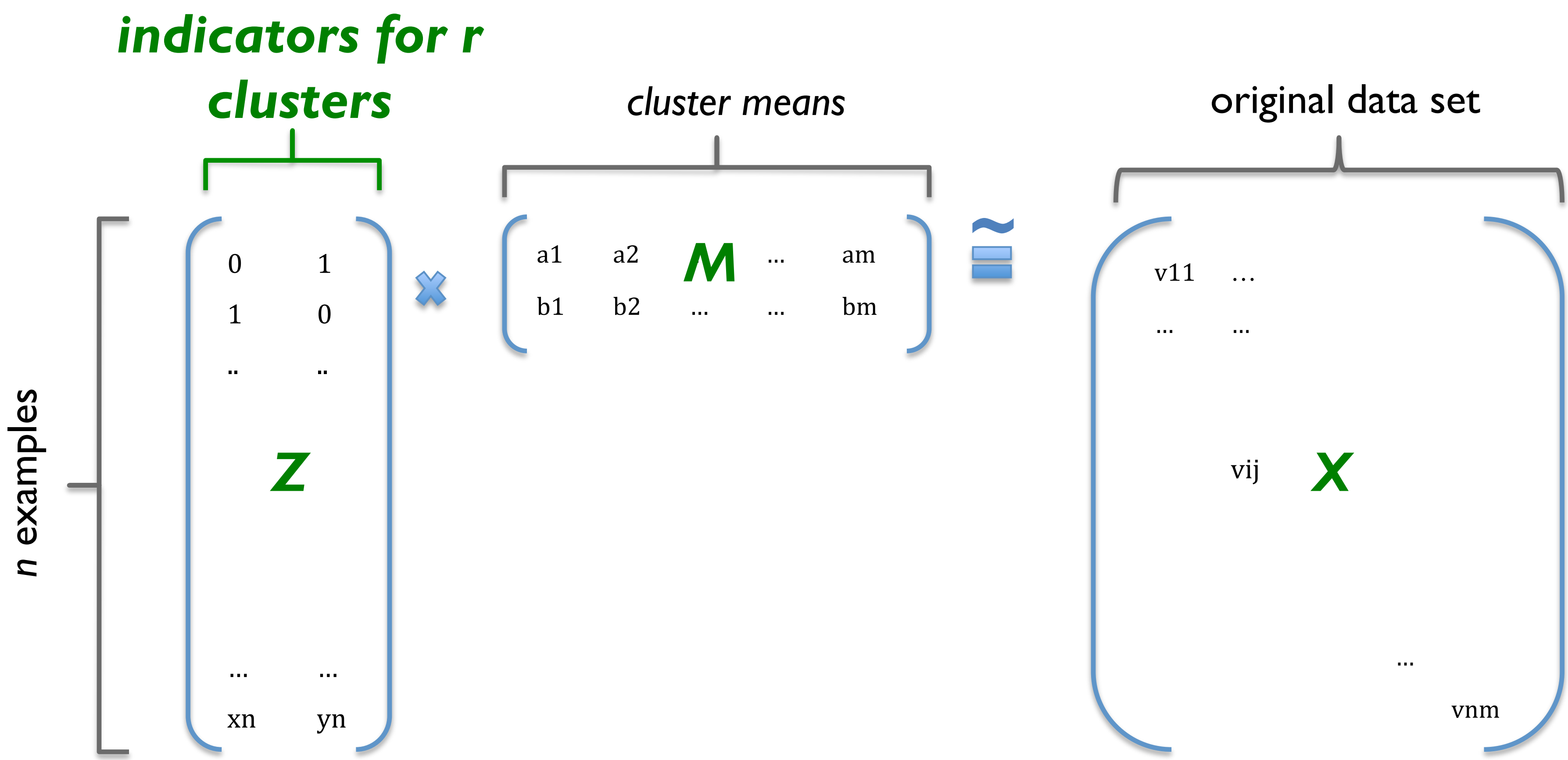
Input	Output
00000001	00000001
00000010	00000010
00000101	00000100
00001000	00001000
00010000	00010000
00100000	00100000
01000000	01000000
10000000	10000000

- Using the following network:



- With *linear* hidden units, how do the weights match up to W and H?

….. vs k-means


indicators for r clusters
cluster means
original data set
n examples
0 1
1 0
.. ..
Z
… …
xn yn
a1 a2 M … am
b1 b2 … … bm
v11 …
… …
vij X
…
vnm