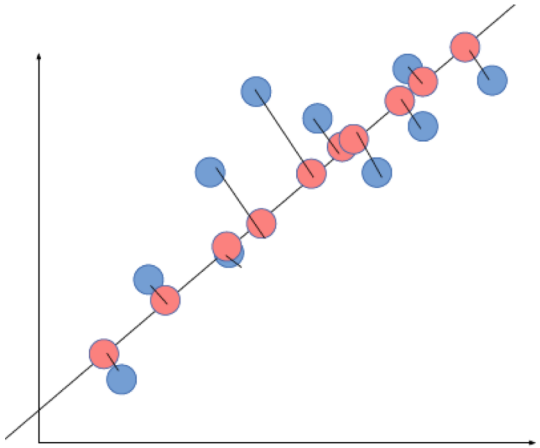
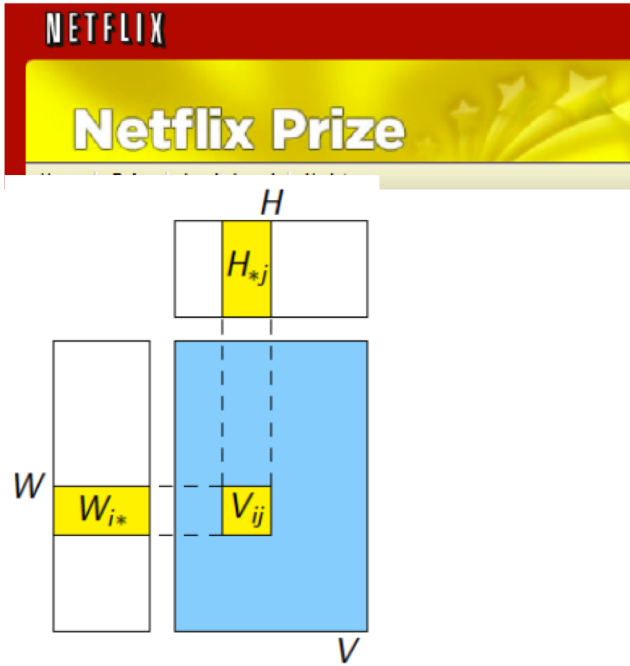
Tufts COMP 135: Introduction to Machine Learning

https://www.cs.tufts.edu/comp/135/2019s/

Dimensionality Reduction & Embedding





Prof. Mike Hughes

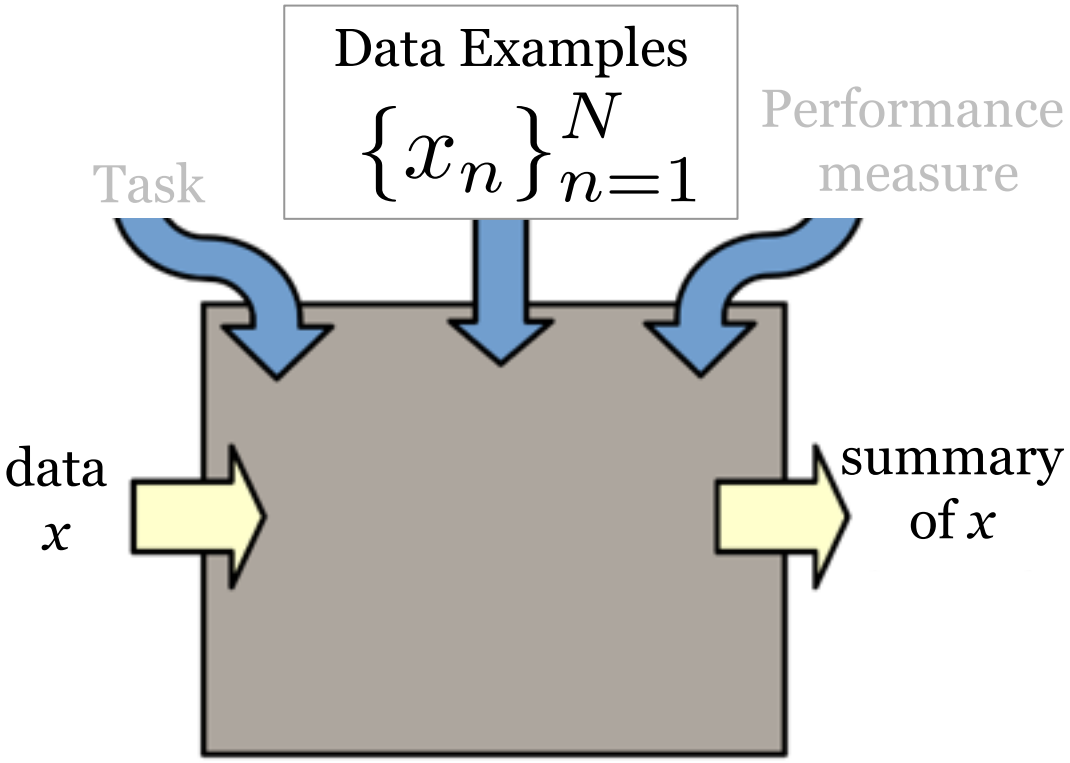
Many ideas/slides attributable to:
Liping Liu (Tufts), Emily Fox (UW)
Matt Gormley (CMU)

What will we learn?

Supervised Learning

Unsupervised Learning

Reinforcement Learning



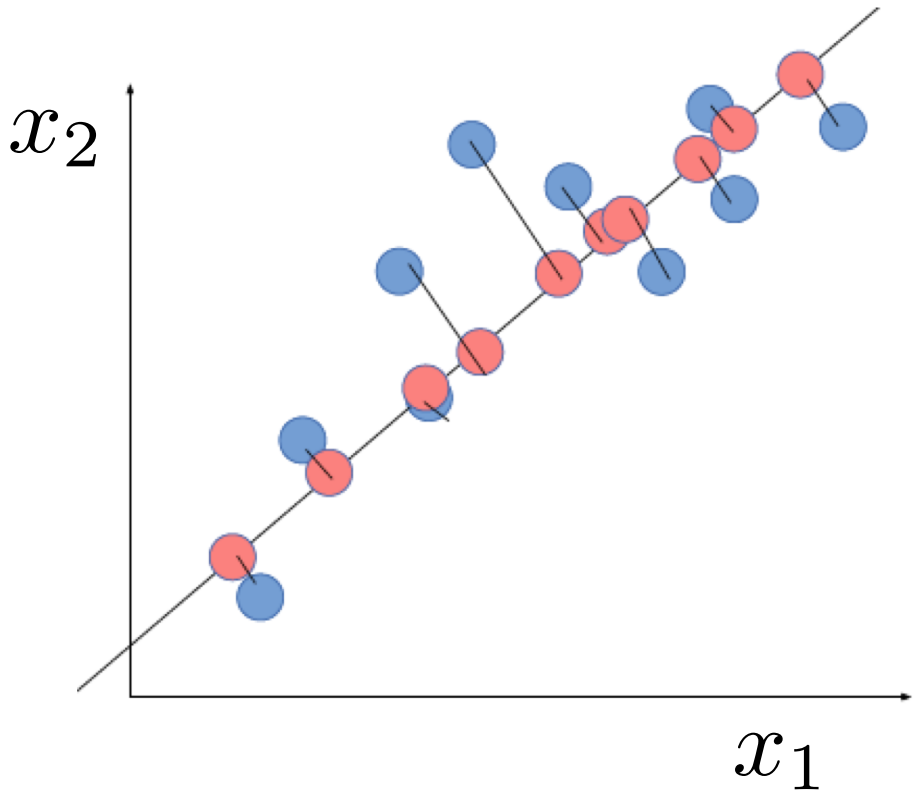
Task: Embedding

Supervised Learning

Unsupervised Learning

embedding

Reinforcement Learning



Dim. Reduction/Embedding
Unit Objectives

- Goals of dimensionality reduction
  - Reduce feature vector size (keep signal, discard noise)
  - "Interpret" features: visualize/explore/understand
- Common approaches
  - Principal Component Analysis (PCA)
  - t-SNE ("tee-snee")
  - word2vec and other neural embeddings
- Evaluation Metrics
  - Storage size
  - "Interpretability"

- Reconstruction error
- Prediction error

Example: 2D viz. of movies

Example Factors

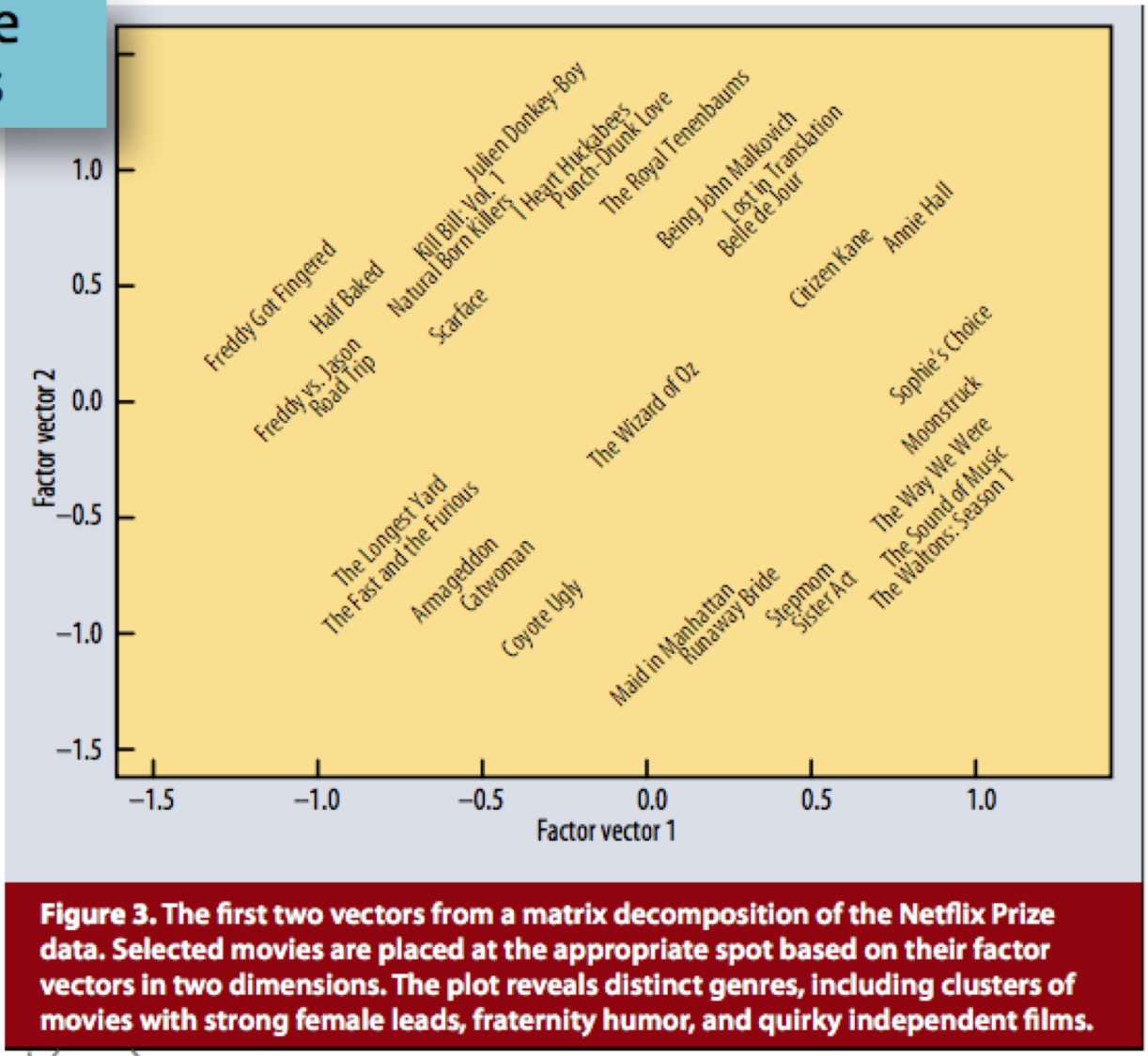
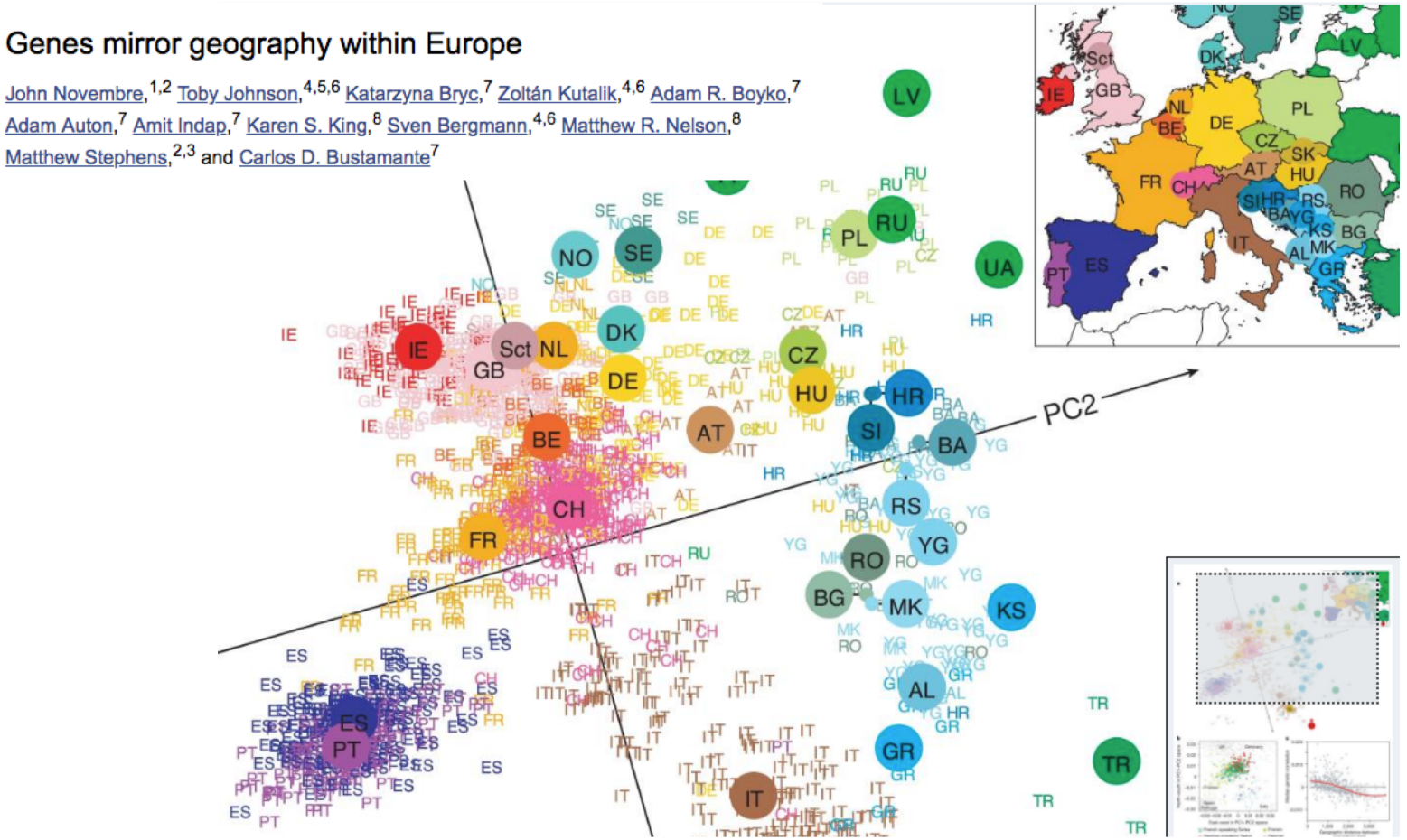


Figure 3. The first two vectors from a matrix decomposition of the Netflix Prize data. Selected movies are placed at the appropriate spot based on their factor vectors in two dimensions. The plot reveals distinct genres, including clusters of movies with strong female leads, fraternity humor, and quirky independent films.

Figure from Koren et al. (2009)

Example: Genes vs. geography



Example: Eigen Clothing

A 2D Human Body Model Dressed in Eigen Clothing

Peng Guan* Oren Freifeld† Michael J. Black*

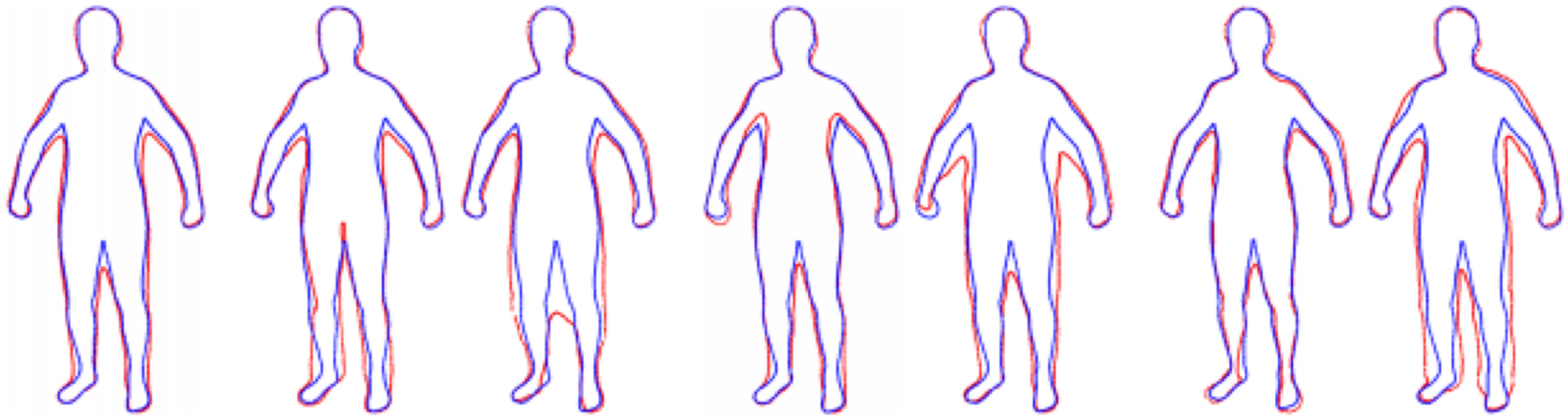
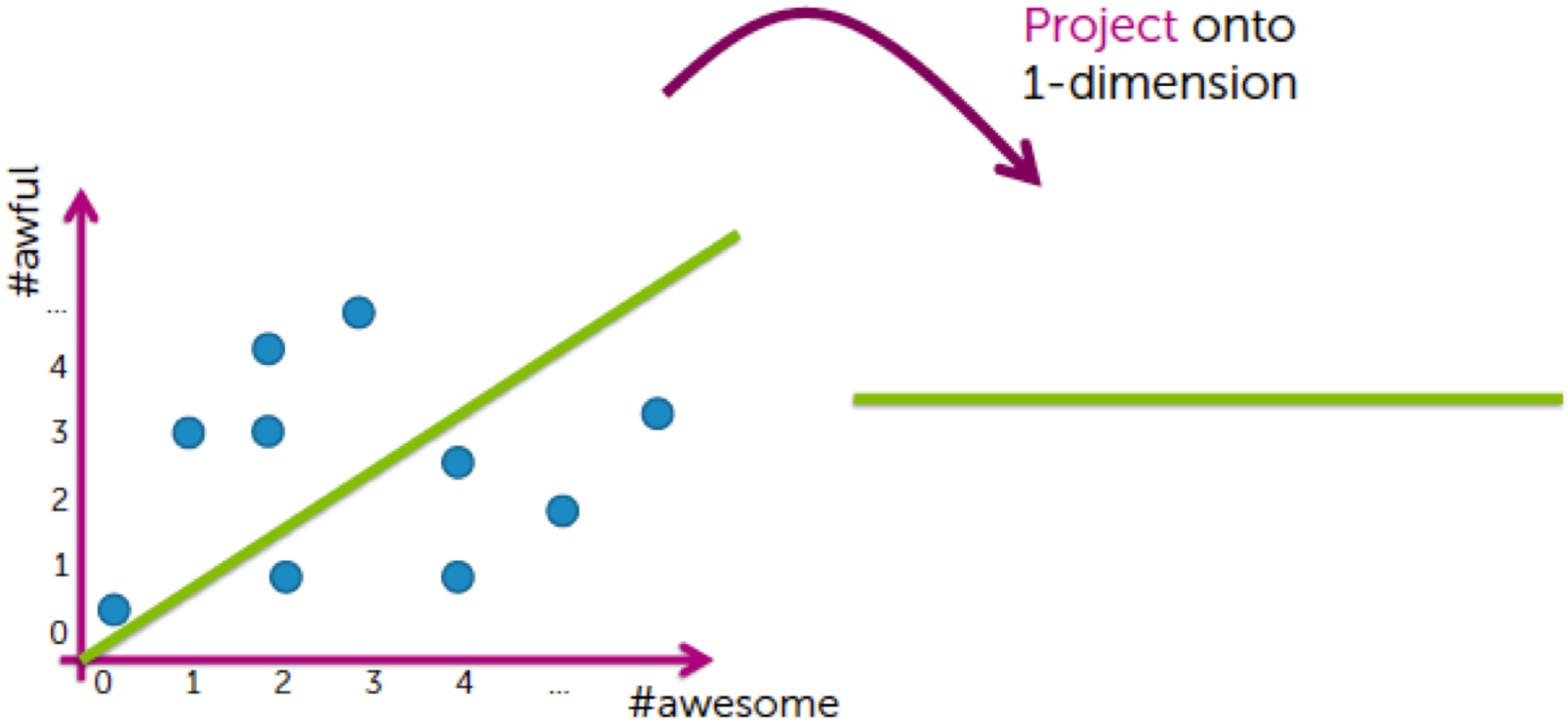


Fig. 4. Eigen clothing. The blue contour is always the same naked shape. The red contour shows the mean clothing contour (a) and ±3 std from the mean for several principal components (b)-(d).

principal components accounting for around 90% of the variance to define the eigen-clothing model. Figure 4 shows the mean and first few clothing eigenvectors for the real data set. This illustrates how the principal components can account for various garments such as long pants, skirts, baggy shirts, etc. Note that

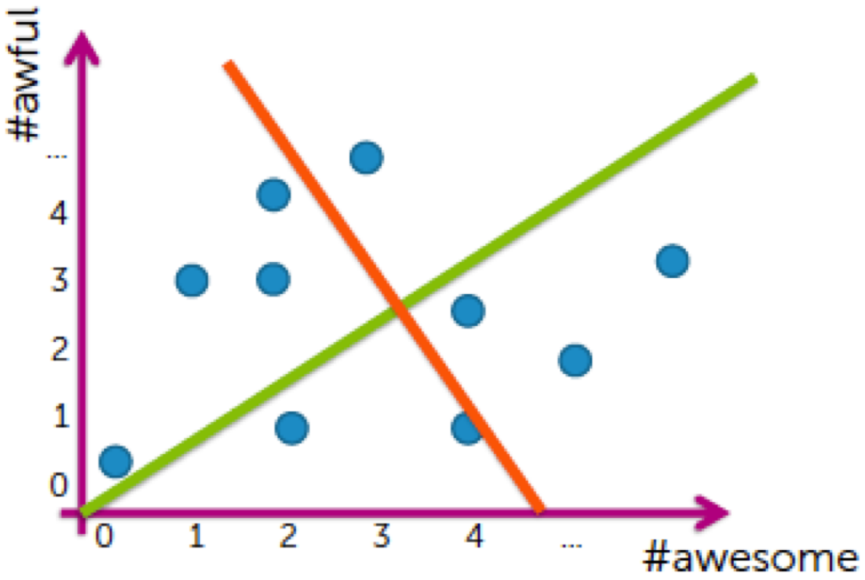
Principal Component Analysis

Linear Projection to 1D



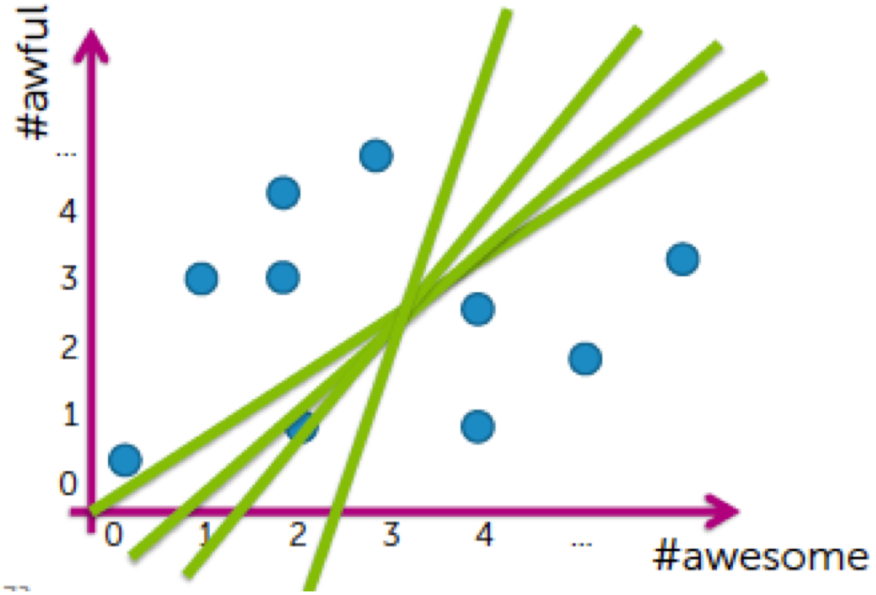
Reconstruction from 1D to 2D

2D Orthogonal Basis



Perfect reconstruction!

Which 1D projection is best?

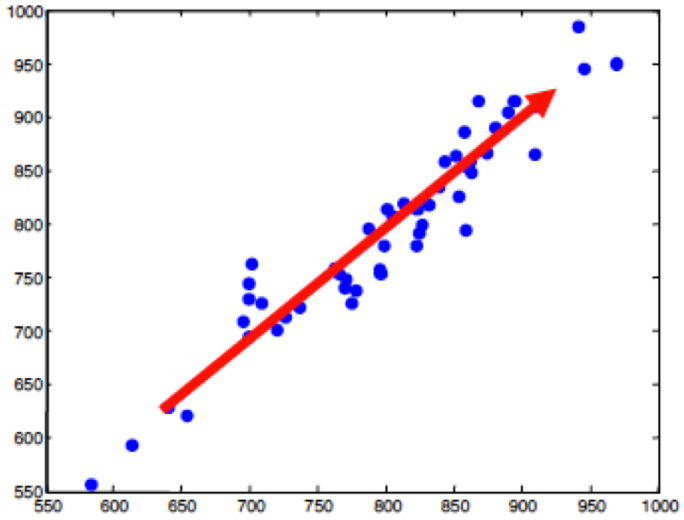


PCA Principles

- Minimize **reconstruction error**
  - Should be able to recreate x from z

- Equivalent to **maximizing variance**
  - Want z to retain maximum information

Best Direction related to Eigenvalues of Data Covariance



Project X to v: $z = \tilde{X} \cdot v$

Variance of projected points:

$$\sum_i (z^{(i)})^2 = z^T z = v^T \tilde{X}^T \tilde{X} v$$

Best "direction" v:

$$\max_v \; v^T \tilde{X}^T \tilde{X} v \quad s.t. \; ||v|| = 1$$

$\Rightarrow$ largest eigenvector of $X^T X$

Principal Component Analysis

Training step: .fit()

- Input:
  - X : training data, N x F
    - N high-dim. example vectors
  - K : int, number of dimensions to discover
    - Satisfies 1 <= K <= F

- Output:
  - m : mean vector, size F
  - V : learned eigenvector basis, K x F
    - One F-dimensional vector for each component
    - Each of the K vectors is orthogonal to every other

Principal Component Analysis

Transformation step: .transform()

- Input:
  - X : training data, N x F
    - N high-dim. example vectors
  - Trained PCA "model"
    - m : mean vector, size F
    - V : learned eigenvector basis, K x F
      - One F-dimensional vector for each component
      - Each of the K vectors is orthogonal to every other
- Output:
  - Z : projected data, N x K

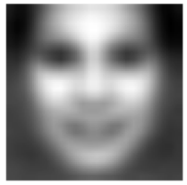
PCA Demo

- http://setosa.io/ev/principal-component-analysis/

Example: EigenFaces

Ex: Viola Jones data set

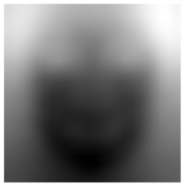
- 24x24 images of faces = 576 dimensional measurements
- Take first K PCA components


Mean


Dir 1


Dir 2


Dir 3


Dir 4 …

Projecting data onto first k dimensions


Xi


k=5


k=10


k=50 ….

PCA: How to Select K?

- 1) Use downstream supervised task metric
  - Regression error
- 2) Use memory constraints of task
  - Can't store more than 50 dims for 1M examples? Take K=50
- 3) Plot cumulative "variance explained"
  - Take K that seems to capture 90% or all variance

PCA Summary

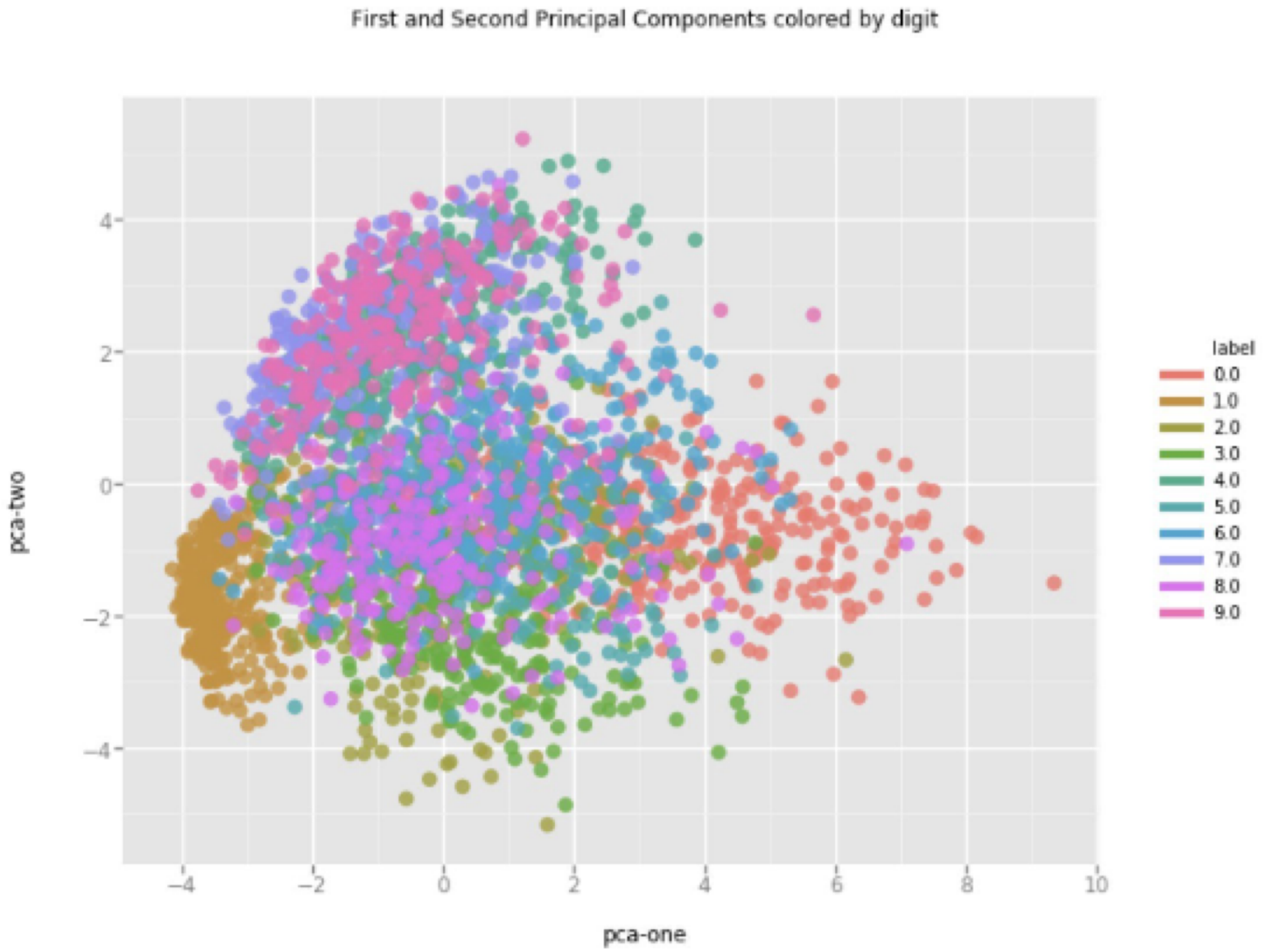
PRO

- Usually, fast to train, fast to test
  - Slow only if finding K eigenvectors of an F x F matrix is slow
- Nested model
  - PCA with K=5 has subset of params equal to PCA with K=4

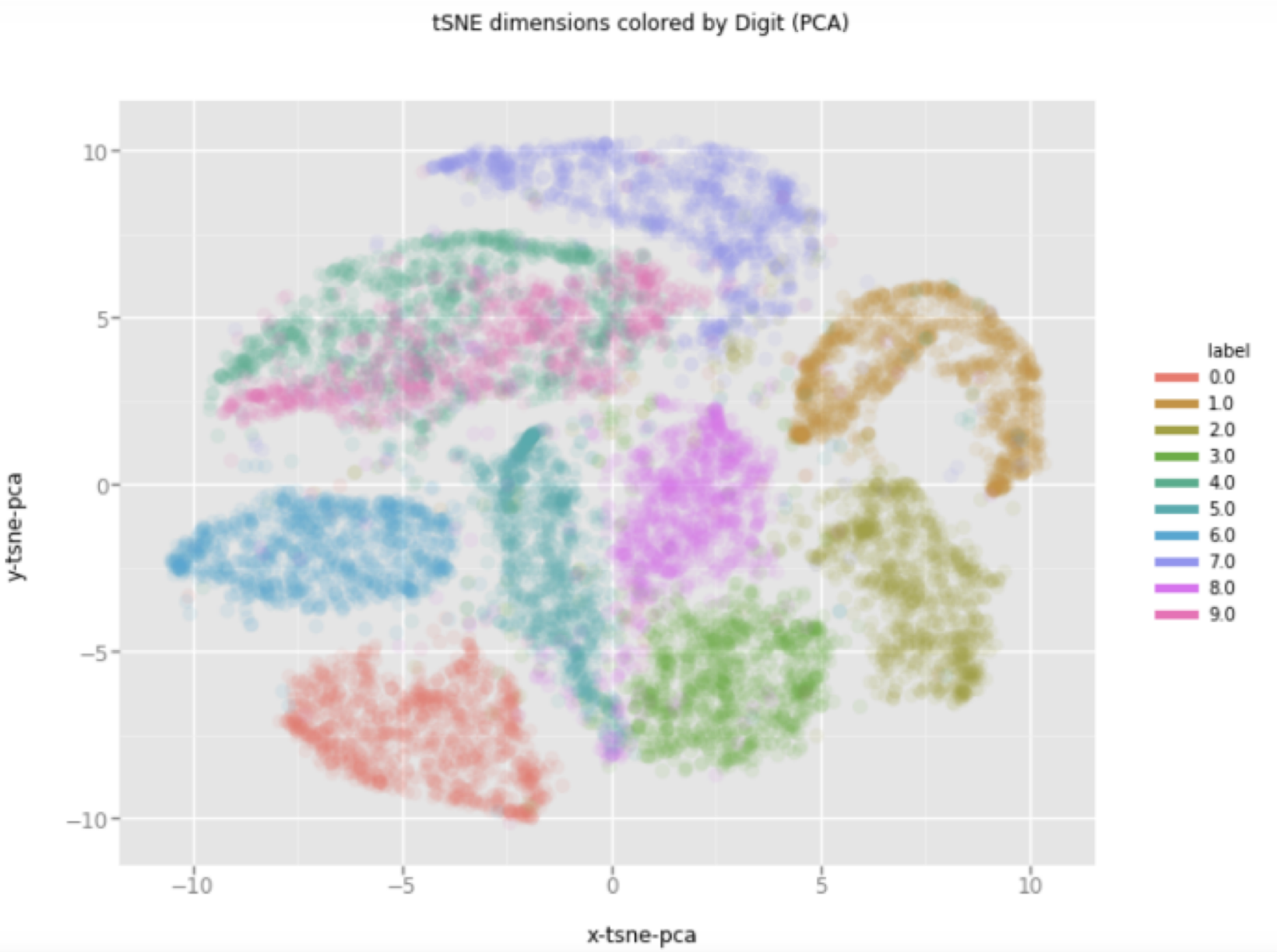
CON

- Learned basis known only up to +/- scaling
- Not often best for supervised tasks

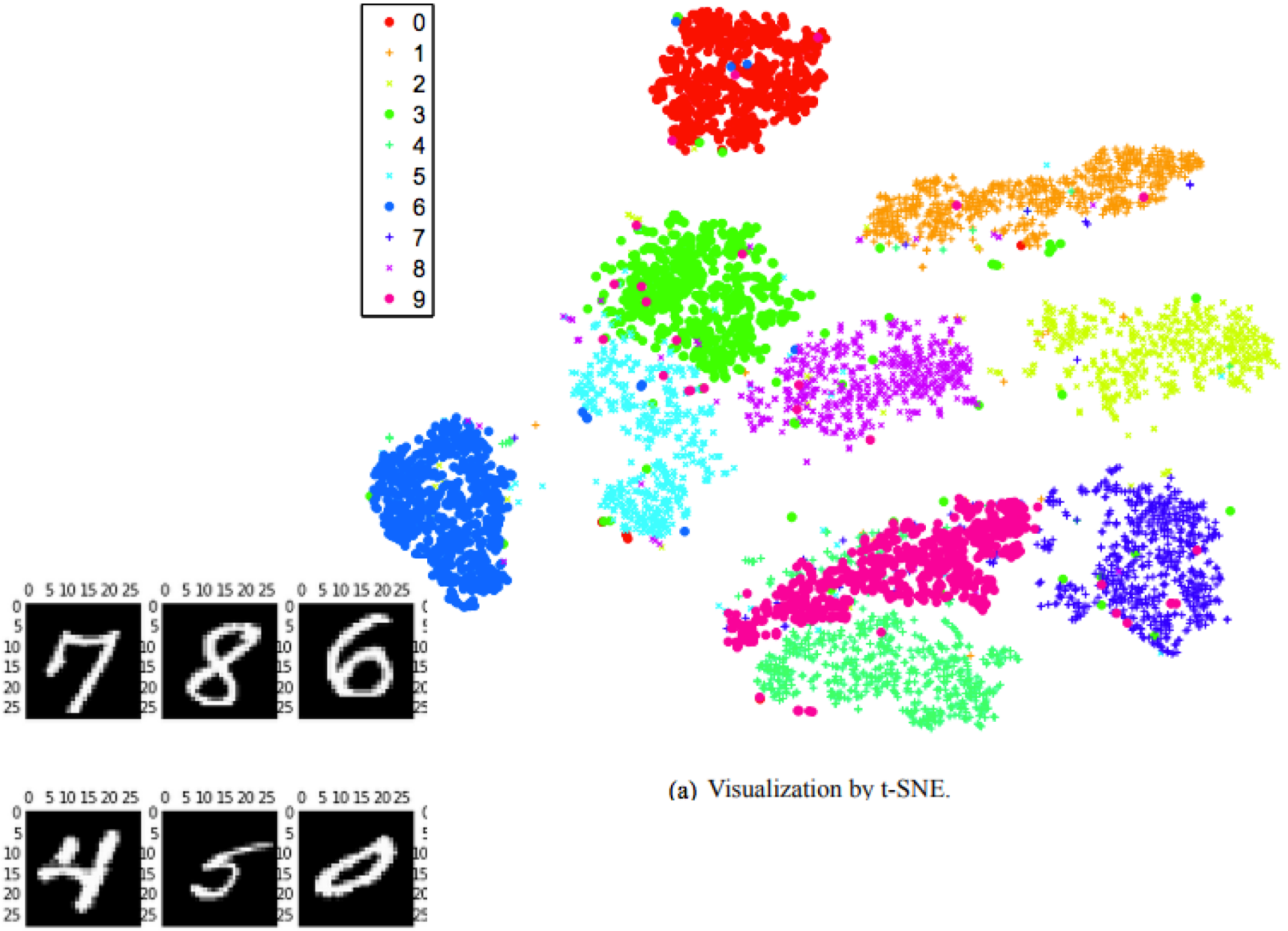
Visualization with t-SNE



Credit: Luuk Derksen (https://medium.com/@luckylwk/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b)



Credit: Luuk Derksen (https://medium.com/@luckylwk/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b)



(a) Visualization by t-SNE.

Practical Tips for t-SNE

- If dim is very high, preprocess with PCA to ~30 dims, then apply t-SNE
- Beware: Non-convex cost function

How to Use t-SNE Effectively

https://distill.pub/2016/misread-tsne/

Matrix Factorization as Learned "Embedding"

Matrix Factorization (MF)

- User i represented by vector $\boldsymbol{u}_i \in R^k$
- Item j represented by vector $\boldsymbol{v}_j \in R^k$
- Inner product $\boldsymbol{u}_i^T \boldsymbol{v}_j$ approximates the utility M_{ij}
- Intuition:
  - Two items with similar vectors get similar utility scores from the same user;
  - Two users with similar vectors give similar utility scores to the same item

Example Factors

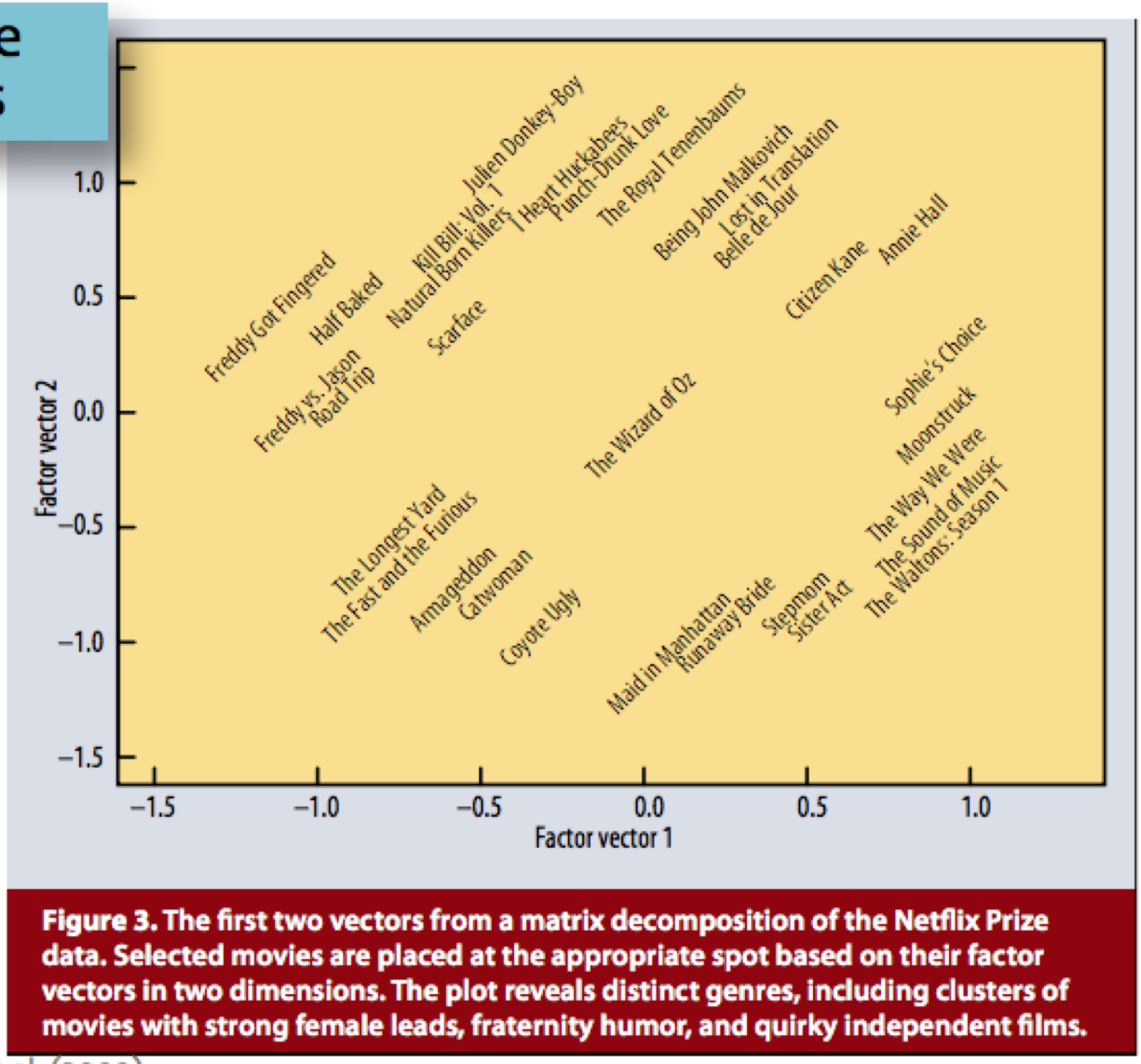


Figure 3. The first two vectors from a matrix decomposition of the Netflix Prize data. Selected movies are placed at the appropriate spot based on their factor vectors in two dimensions. The plot reveals distinct genres, including clusters of movies with strong female leads, fraternity humor, and quirky independent films.

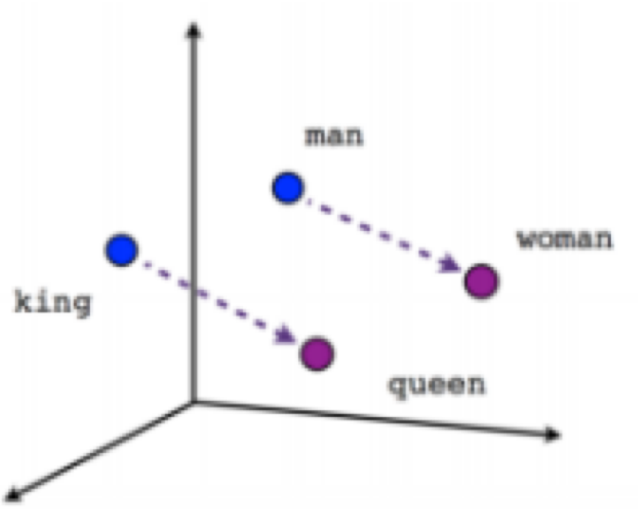
Figure from Koren et al. (2009)

Word Embeddings

Word Embeddings (word2vec)

Goal: map each word in vocabulary to an embedding vector

- Preserve semantic meaning in this new vector space



Male-Female

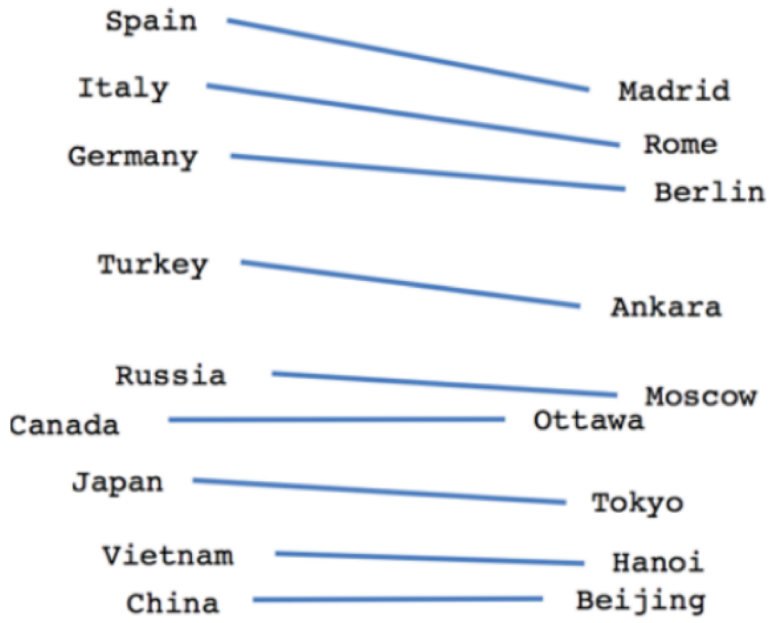
Verb tense

vec(swimming) – vec(swim) + vec(walk) = vec(walking)

Word Embeddings (word2vec)

Goal: map each word in vocabulary to an embedding vector

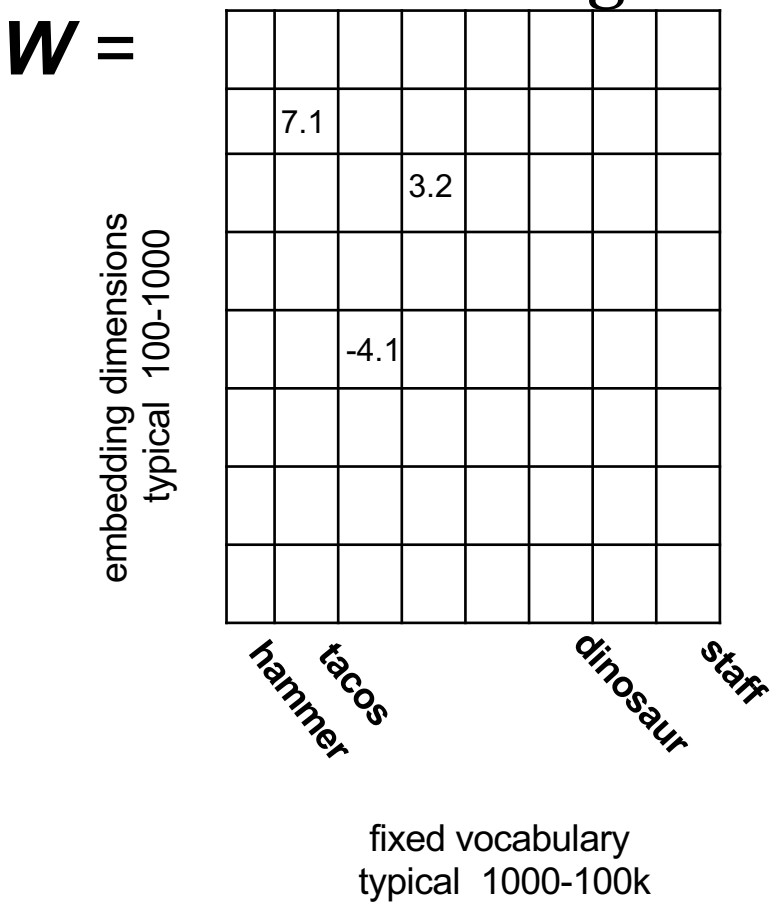
- Preserve semantic meaning in this new vector space



Country-Capital

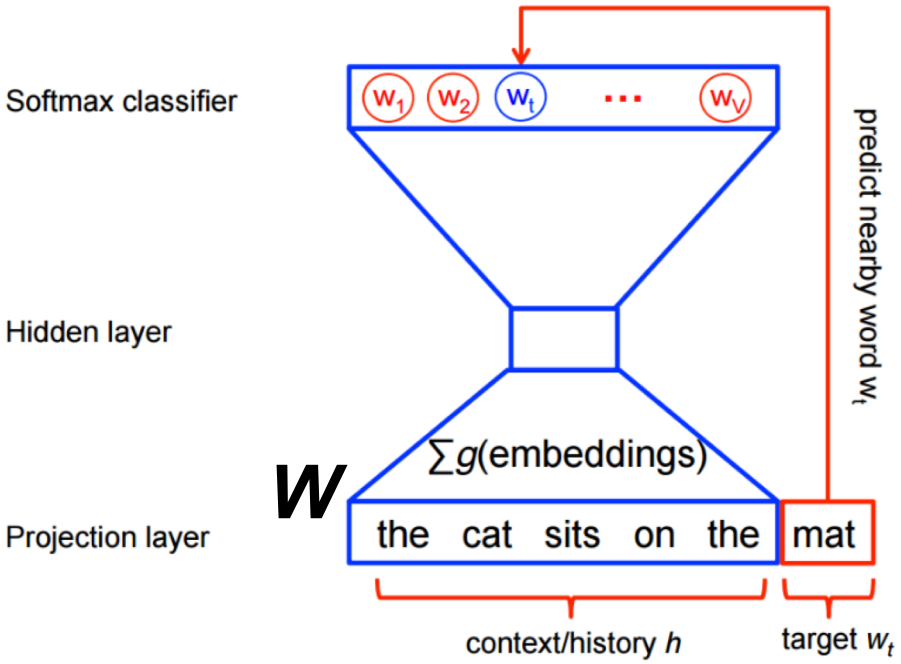
How to embed?

Goal: learn weights



Training

Reward embeddings that predict nearby words in the sentence.



Credit:
https://www.tensorflow.org/tutorials/representation/word2vec