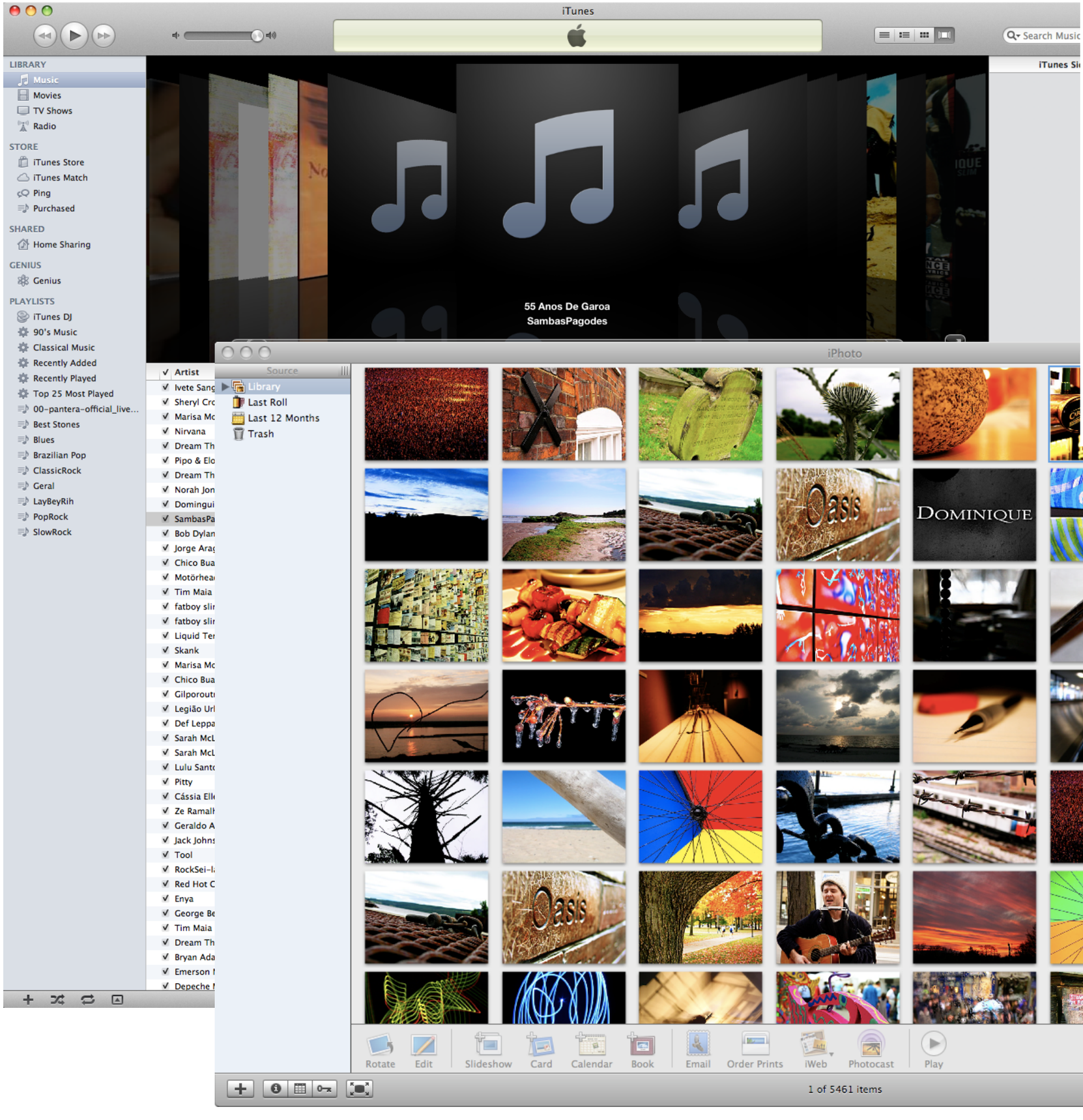
CS-GY-9223
Visualization for Machine Learning

Dimensionality Reduction

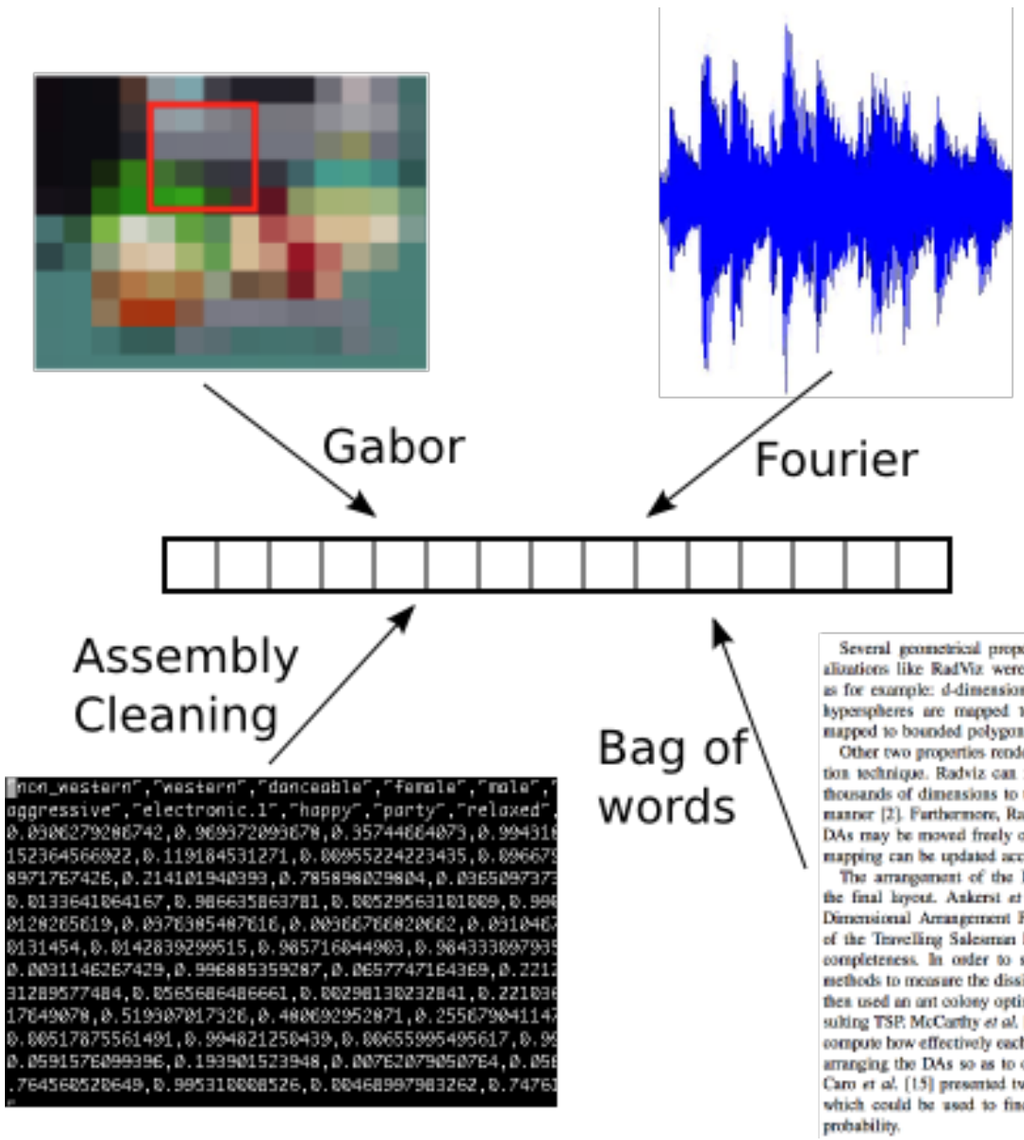
Course and Slides based on lectures by David Sontag, Carlos Guestrin and Luke Zettlemoyer, and Luis Gustavo Nonato

Dimensionality reduction

- Input data may have thousands or millions of dimensions!
  - e.g., text data has ???, images have ???
- **Dimensionality reduction**: represent data with fewer dimensions
  - easier learning – fewer parameters
  - visualization – show high dimensional data in 2D
  - discover "intrinsic dimensionality" of data
    - high dimensional data that is truly lower dimensional
    - noise reduction


iTunes
LIBRARY
Music
Movies
TV Shows
Radio
STORE
iTunes Store
iTunes Match
Ping
Purchased
SHARED
Home Sharing
GENIUS
Genius
PLAYLISTS
iTunes DJ
90's Music
Classical Music
Recently Added
Recently Played
Top 25 Most Played
00-pantera-official_live...
Best Stones
Blues
Brazilian Pop
ClassicRock
Geral
LayBeyRih
PopRock
SlowRock
Search Music
55 Anos De Garoa
SambasPagodes
Artist
Ivete Sang
Sheryl Cro
Marisa Mo
Nirvana
Dream Th
Pipo & Elo
Dream Th
Norah Jon
Domingui
SambasPa
Bob Dylan
Jorge Arag
Chico Bua
Motörhea
Tim Maia
fatboy slir
fatboy slir
Liquid Ter
Skank
Marisa Mo
Chico Bua
Gilporouti
Legião Url
Def Leppa
Sarah McL
Sarah McL
Lulu Santo
Pitty
Cássia Ell
Ze Ramalh
Geraldo A
Jack Johns
Tool
RockSei-l
Red Hot C
Enya
George Be
Tim Maia
Dream Th
Bryan Ada
Emerson
Depeche
iPhoto
Source
Library
Last Roll
Last 12 Months
Trash
Oasis
DOMINIQUE
PEOPLE SUCK
ROBOTS RULE!
Rotate
Edit
Slideshow
Card
Calendar
Book
Email
Order Prints
iWeb
Photocast
Play
Search
1 of 5461 items

Textual, image, sound, and tabular data become mathematically manageable when embedded into a n-dimensional Cartesian space.

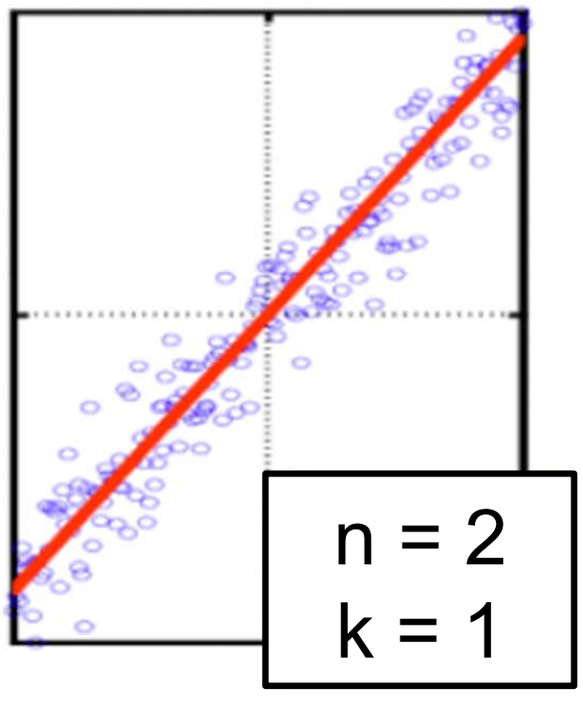


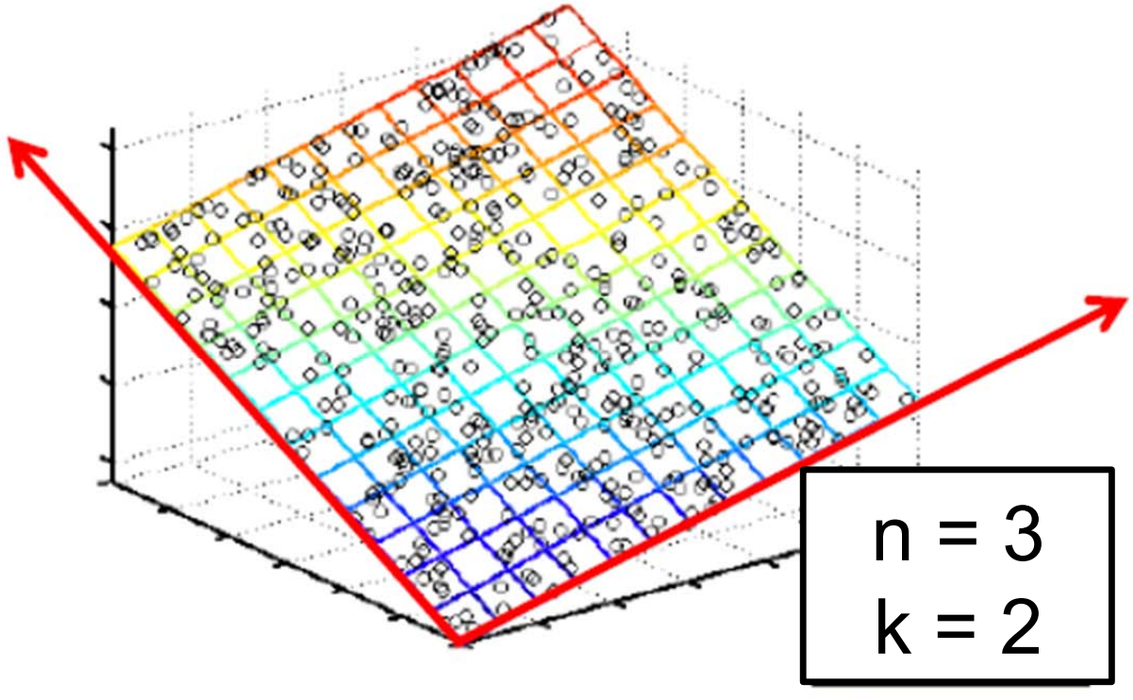
Given a multidimensional dimensional data set:

- How are the points spread?
- Are there well defined groups of similar instances?
- Which is the gist information contained in the data?
- Which are the relevant attributes in each group?

Dimension reduction

- Assumption: data (approximately) lies on a lower dimensional space
- Examples:

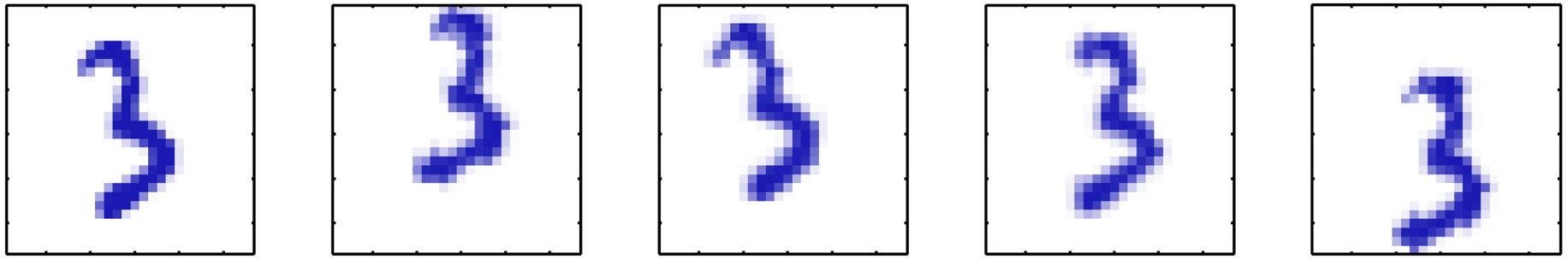




Slide from Yi Zhang

Example (from Bishop)

- Suppose we have a dataset of digits (“3”) perturbed in various ways:



- What operations did I perform? What is the data’s intrinsic dimensionality?
- Here the underlying manifold is *nonlinear*

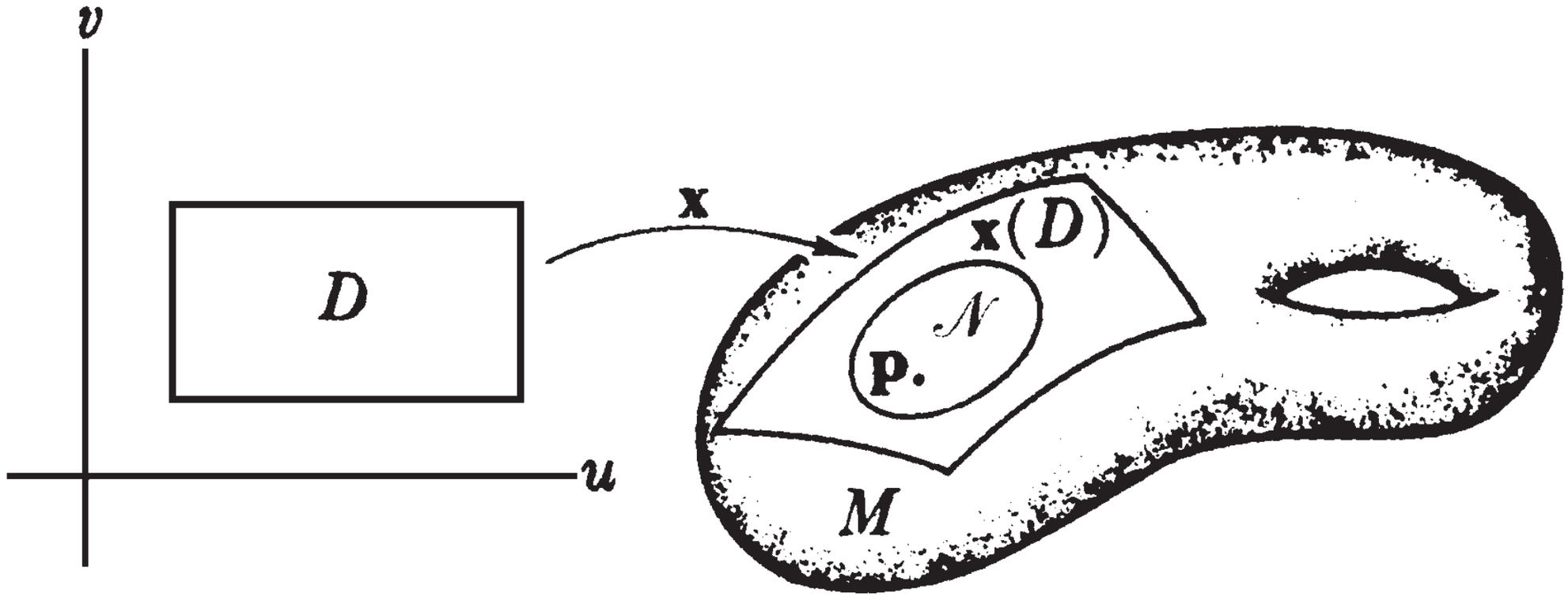

v
u
D
x
x(D)
N
p.
M

FIG. 4.2

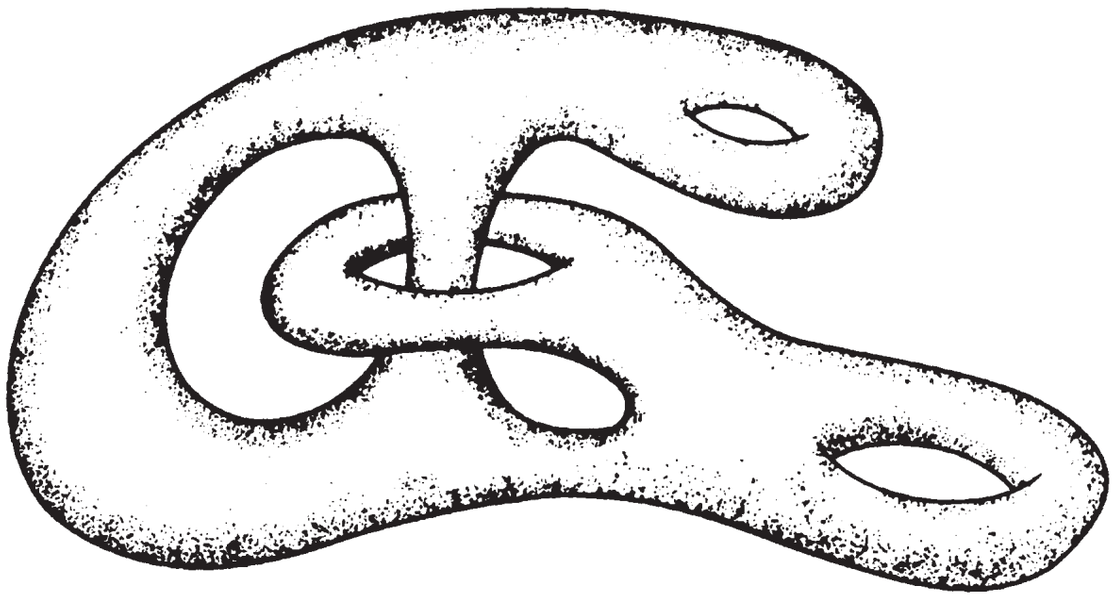
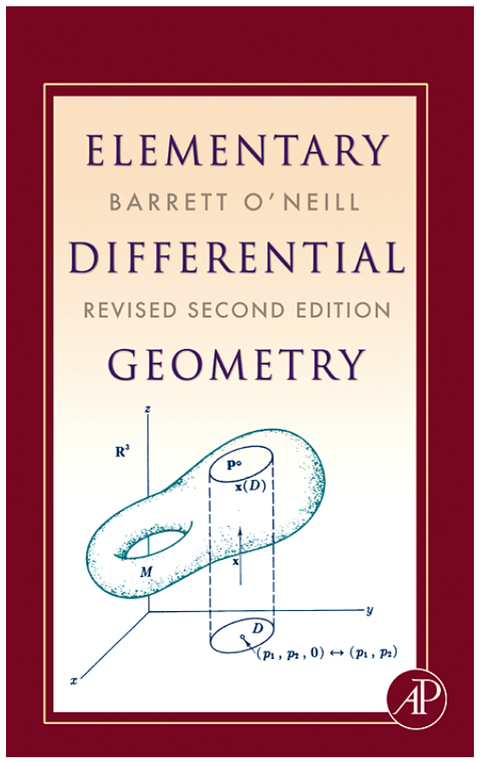


FIG. 4.8


ELEMENTARY
BARRETT O'NEILL
DIFFERENTIAL
REVISED SECOND EDITION
GEOMETRY
z
R³
P₀
x(D)
M
x
y
D
(p₁ , p₂ , 0) ↔ (p₁ , p₂)
x
AP

Lower dimensional projections

- Obtain new feature vector by transforming the original features $x_1 \ldots x_n$

$$z_1 = w_0^{(1)} + \sum_i w_i^{(1)} x_i$$

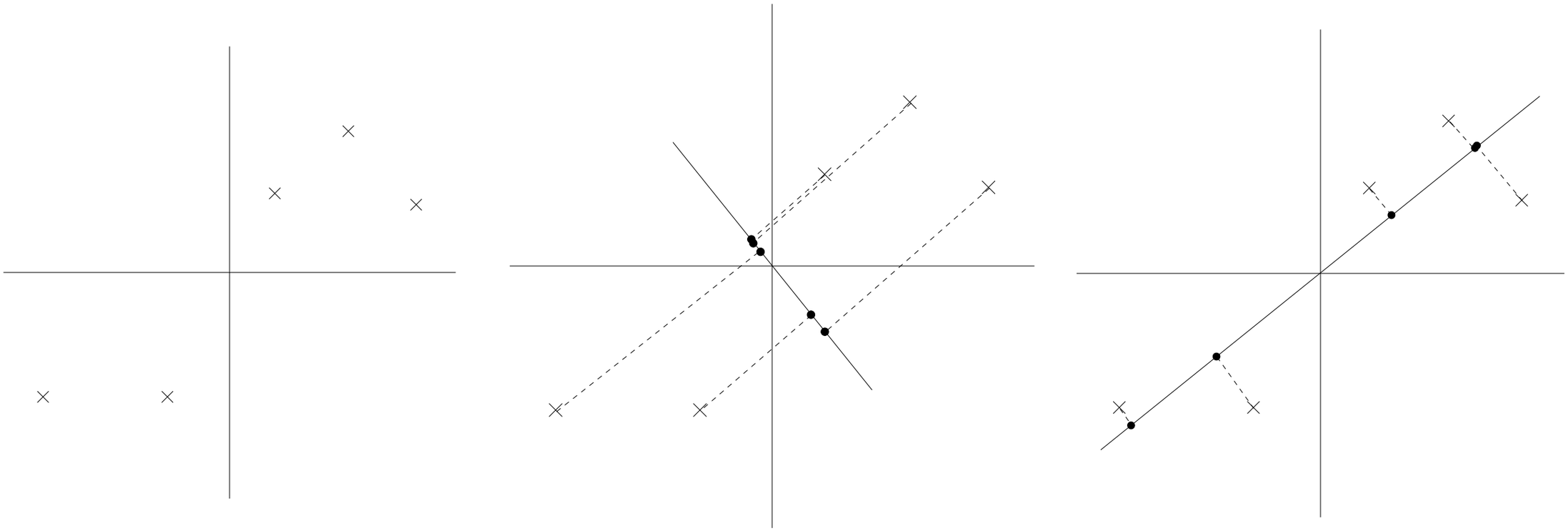
$$\cdots$$

$$z_k = w_0^{(k)} + \sum_i w_i^{(k)} x_i$$

In general will not be invertible – cannot go from *z* back to *x*

- New features are linear combinations of old ones
- Reduces dimension when k<n
- This is typically done in an unsupervised setting
  - just **X**, but no Y

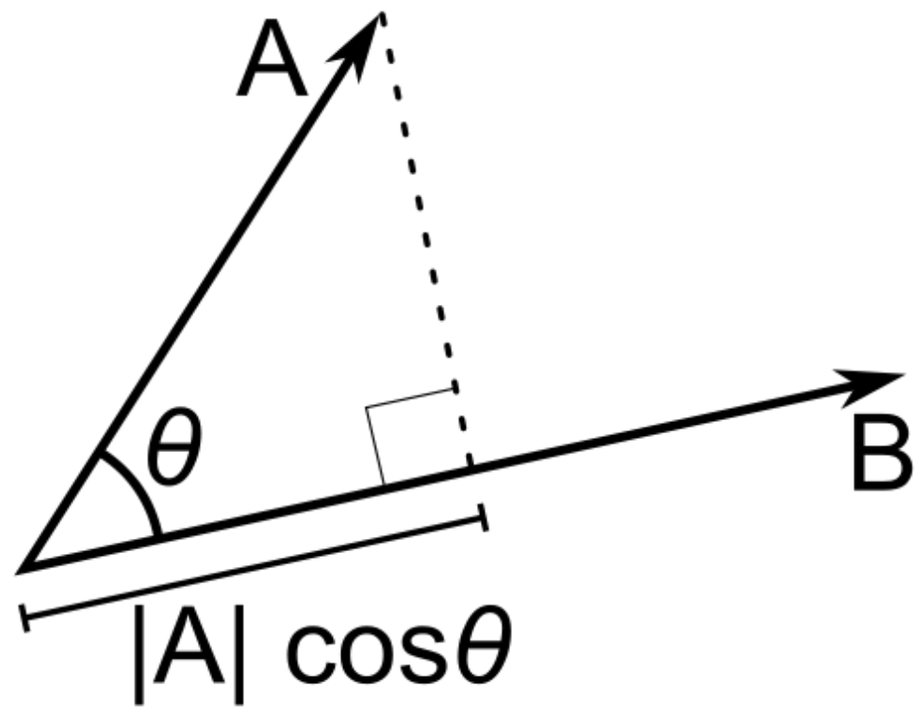
Which projection is better?



From notes by Andrew Ng

Reminder: Vector Projections

- Basic definitions:
  - $A.B = |A||B|\cos\theta$
  - $\cos\theta = |adj|/|hyp|$



- Assume $|B|=1$ (unit vector)
  - $A.B = |A|\cos\theta$
  - So, dot product is length of projection!!!

Using a new basis for the data

- 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:
      - $\mathbf{u}_j \bullet \mathbf{u}_j = 1$, and $\mathbf{u}_j \bullet \mathbf{u}_l = 0$ for $j \neq l$
  - **select a center** – $\bar{\mathbf{x}}$, defines offset of space
  - **best coordinates** in lower dimensional space defined by dot-products: $(z_1,\ldots,z_k)$, $z_j^i = (\mathbf{x}^i - \bar{\mathbf{x}}) \bullet \mathbf{u}_j$

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

Linear algebra review:

http://immersivemath.com/ila/index.html

Maximize variance of projection

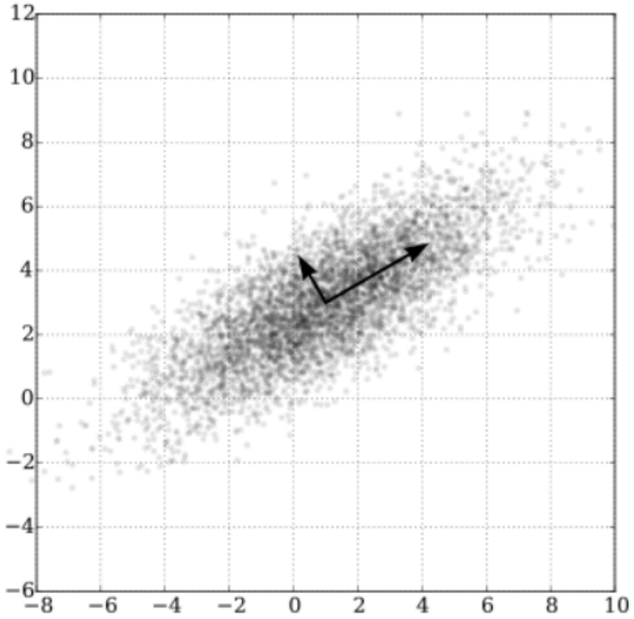
Let $x^{(i)}$ be the i^{th} data point minus the mean.

Choose unit-length u to maximize:

$$\frac{1}{m}\sum_{i=1}^{m}(x^{(i)T}u)^2 = \frac{1}{m}\sum_{i=1}^{m}u^T x^{(i)}x^{(i)T}u$$

Covariance matrix Σ

$$= u^T\left(\frac{1}{m}\sum_{i=1}^{m}x^{(i)}x^{(i)T}\right)u.$$

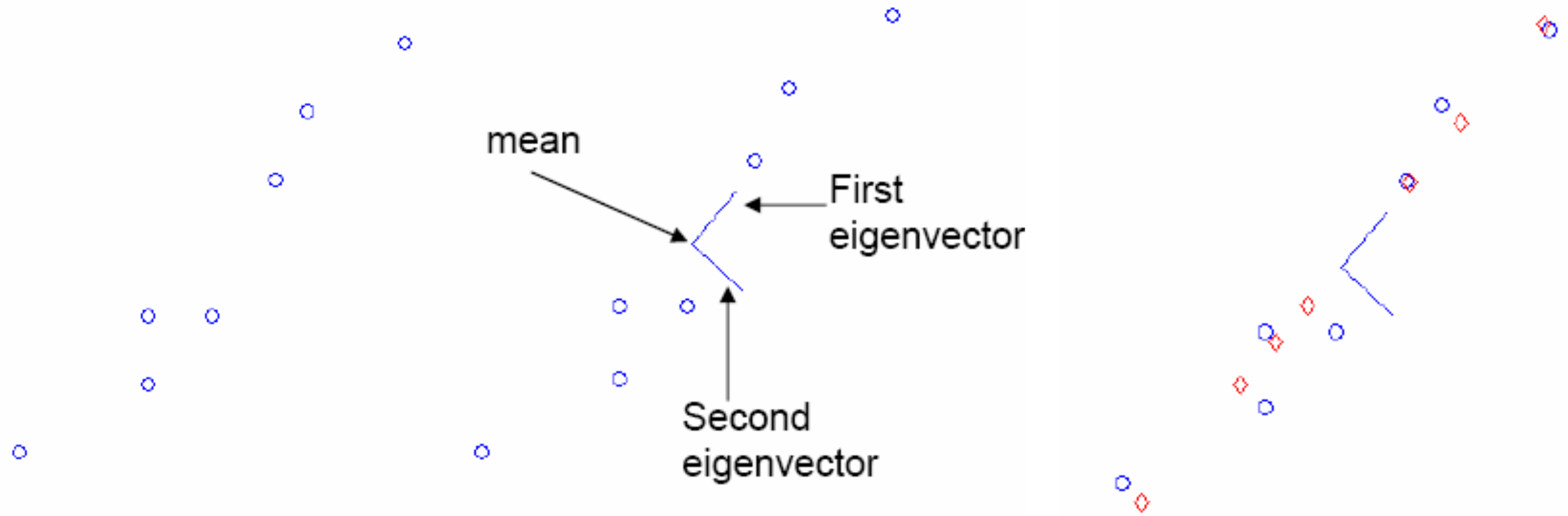


PCA of a multivariate Gaussian distribution centered at (1,3) with a standard deviation of 3 in roughly the (0.866, 0.5) direction and of 1 in the orthogonal direction. The vectors

PCA example

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

Data: Projection: Reconstruction:



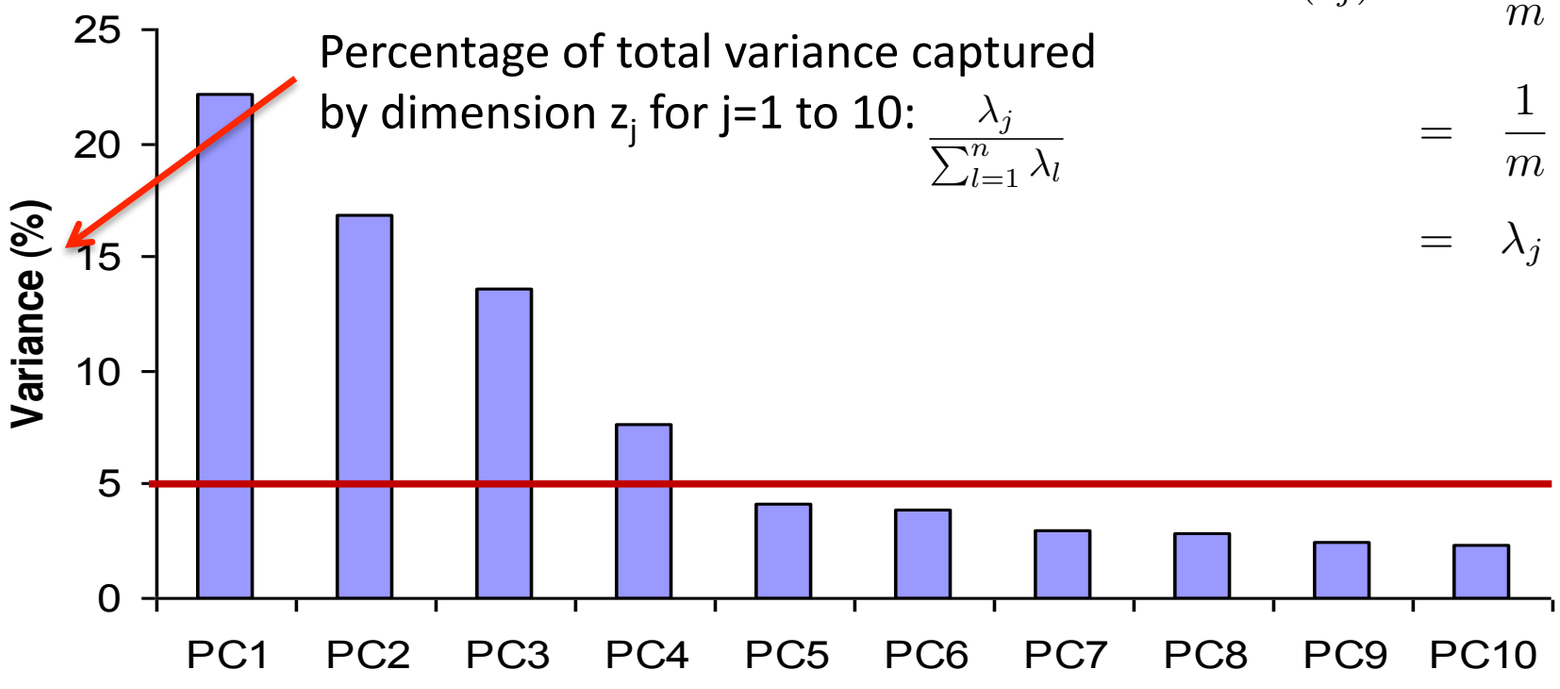
Dimensionality reduction with PCA

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.

$$\begin{aligned}\mathrm{var}(z_j) &= \frac{1}{m}\sum_{i=1}^{m}(z_j^i)^2 \\ &= \frac{1}{m}\sum_{i=1}^{m}(x^i \cdot u_j)^2 \\ &= \lambda_j\end{aligned}$$



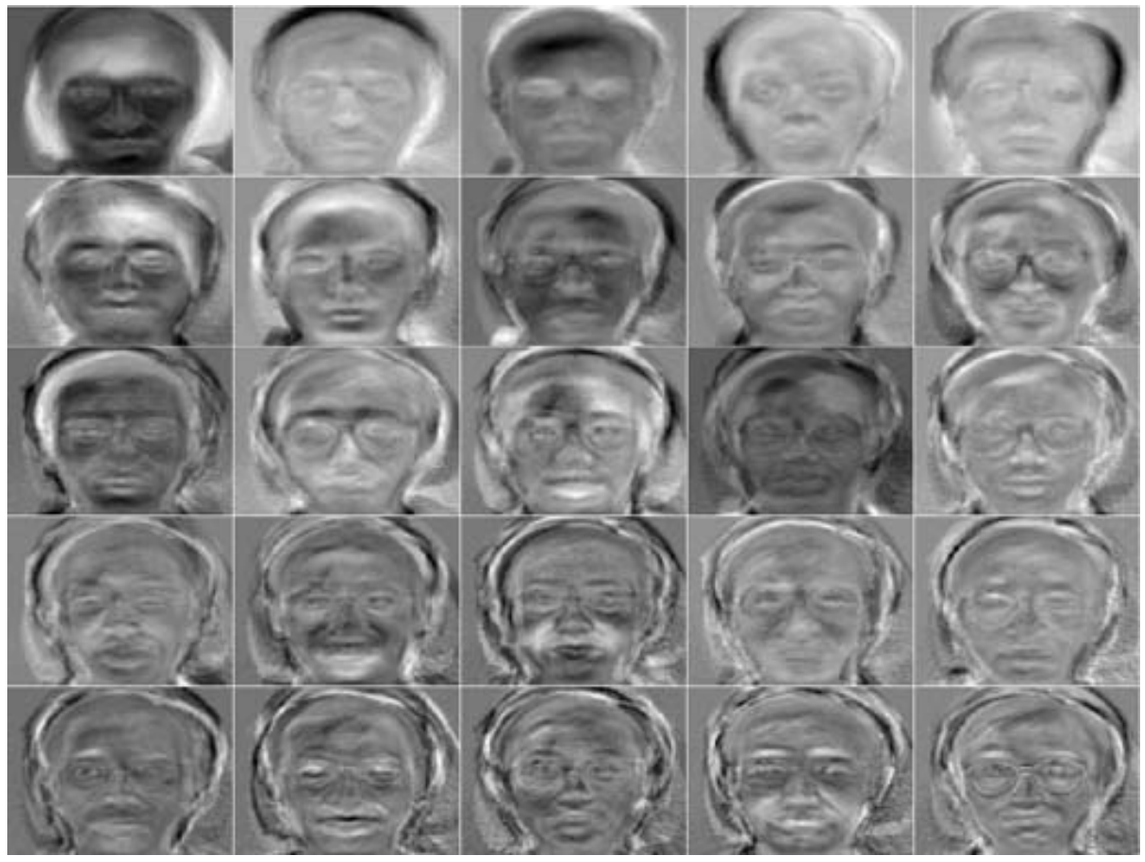
You might lose some information, but if the eigenvalues are small, you don't lose much

Slide from Aarti Singh

Eigenfaces [Turk, Pentland ’91]

- Input images:

- Principal components:



Eigenfaces reconstruction

- Each image corresponds to adding together (weighted versions of) the principal components:

See also: http://immersivemath.com/ila/ch10_eigen/ch10.html

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 k eigenvectors
  - great implementations available, e.g., Matlab svd

SVD

- Write $\mathbf{X} = \mathbf{Z}\,\mathbf{S}\,\mathbf{U}^T$
  - $\mathbf{X} \leftarrow$ data matrix, one row per datapoint
  - $\mathbf{S} \leftarrow$ singular value matrix, diagonal matrix with entries σ_i
    - Relationship between singular values of $\mathbf{X}$ and eigenvalues of Σ given by $\lambda_i = \sigma_i^2/m$
  - $\mathbf{Z} \leftarrow$ weight matrix, one row per datapoint
    - $\mathbf{Z}$ times $\mathbf{S}$ gives coordinate of x_i in eigenspace
  - $\mathbf{U}^T \leftarrow$ singular vector matrix
    - In our setting, each row is eigenvector $\mathbf{u}_j$

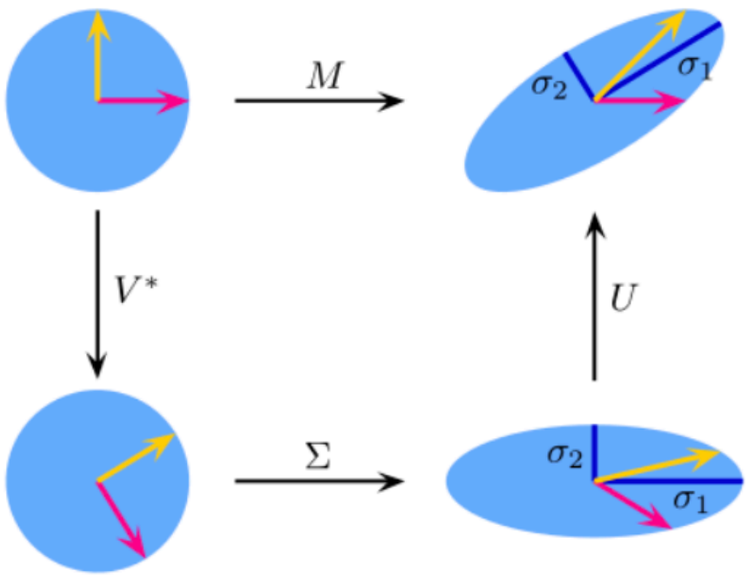


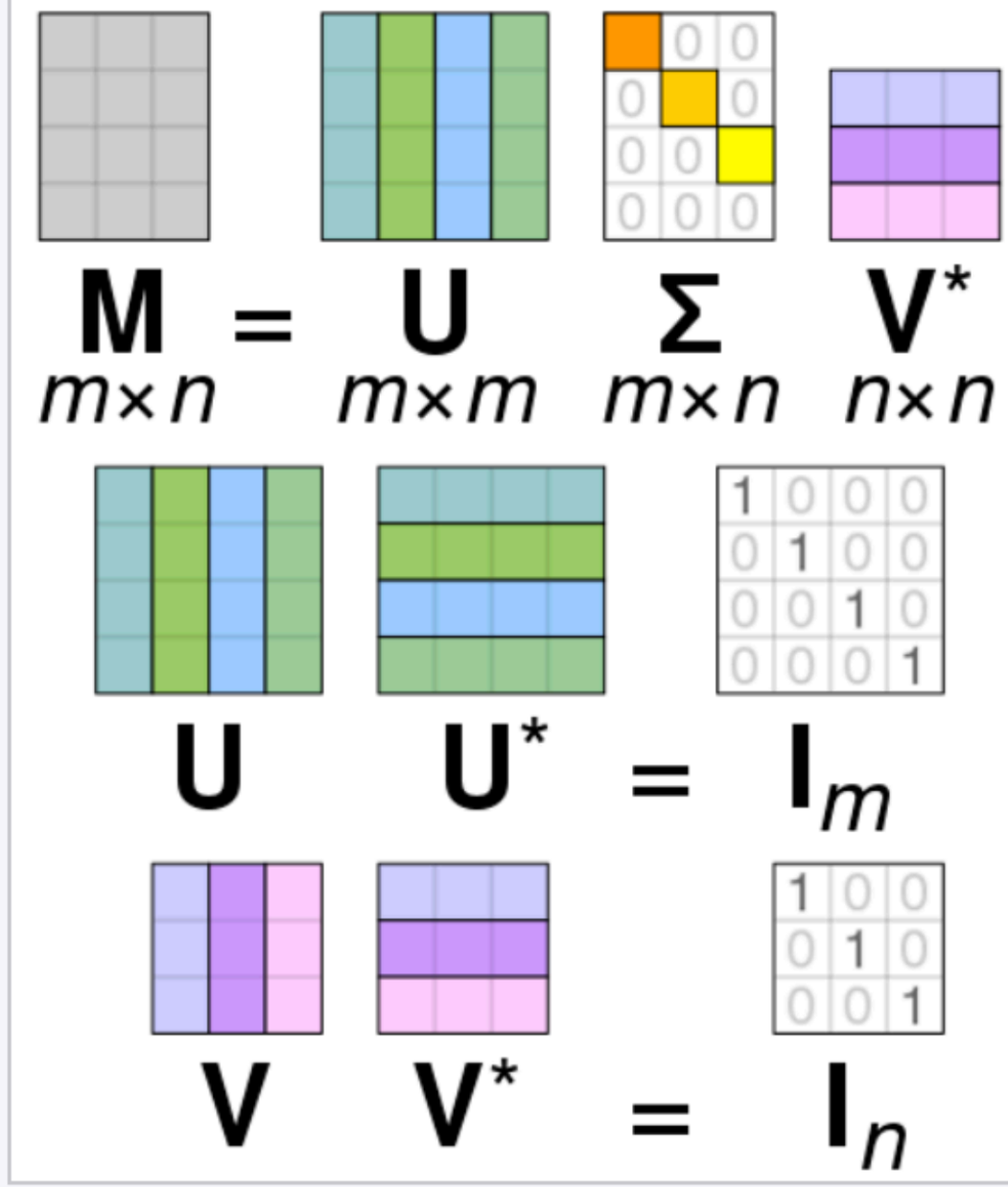
Illustration of the singular value decomposition **UΣV*** of a real 2×2 matrix **M**.

Top: The action of **M**, indicated by its effect on the unit disc *D* and the two canonical unit vectors e_1 and e_2.

Left: The action of **V***, a rotation, on *D*, e_1, and e_2.

Bottom: The action of **Σ**, a scaling by the singular values σ_1 horizontally and σ_2 vertically.

Right: The action of **U**, another rotation.



Visualisation of the matrix multiplications in singular value decomposition

https://en.wikipedia.org/wiki/Singular_value_decomposition

Instances are represented as column vectors:

$$A = \begin{bmatrix} \vdots & & \vdots \\ \cdots & a^i & \cdots \\ \vdots & & \vdots \end{bmatrix}$$

n-dimensional instance

$$A = U\Sigma V^\top$$ SVD decomposition

$$\begin{bmatrix} \vdots & \vdots & & \vdots \\ u^1 & u^2 & \cdots & u^n \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & & & \\ 0 & \sigma_2 & 0 & \cdots & & \\ & & \ddots & & & \\ \cdots & 0 & \sigma_p & 0 & \cdots & \\ & & \cdots & 0 & \cdots & \\ & & & & \ddots & \\ & & & & \cdots & 0 \end{bmatrix} \begin{bmatrix} \vdots & \vdots & & \vdots \\ v^1 & v^2 & \cdots & v^m \\ \vdots & \vdots & & \vdots \end{bmatrix}^\top$$

Orthonormal basis corresponding to the principal directions of the data

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 $\mathbf{U}^T$)
  - **Coefficients:** project each point onto the new vectors

Maximize variance of projection

Let $x^{(i)}$ be the i^{th} data point minus the mean.

Choose unit-length u to maximize:

$$\frac{1}{m}\sum_{i=1}^{m}(x^{(i)T}u)^2 = \frac{1}{m}\sum_{i=1}^{m}u^T x^{(i)}x^{(i)T}u$$

Covariance matrix Σ

$$= u^T\left(\frac{1}{m}\sum_{i=1}^{m}x^{(i)}x^{(i)T}\right)u.$$

Let ||u||=1 and maximize. Using the method of Lagrange multipliers, can show that the solution is given by the principal eigenvector of the covariance matrix! **(shown on board)**

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
- Non-linear dimensionality reduction

Contents [hide]

https://en.wikipedia.org/wiki/Dimensionality_reduction

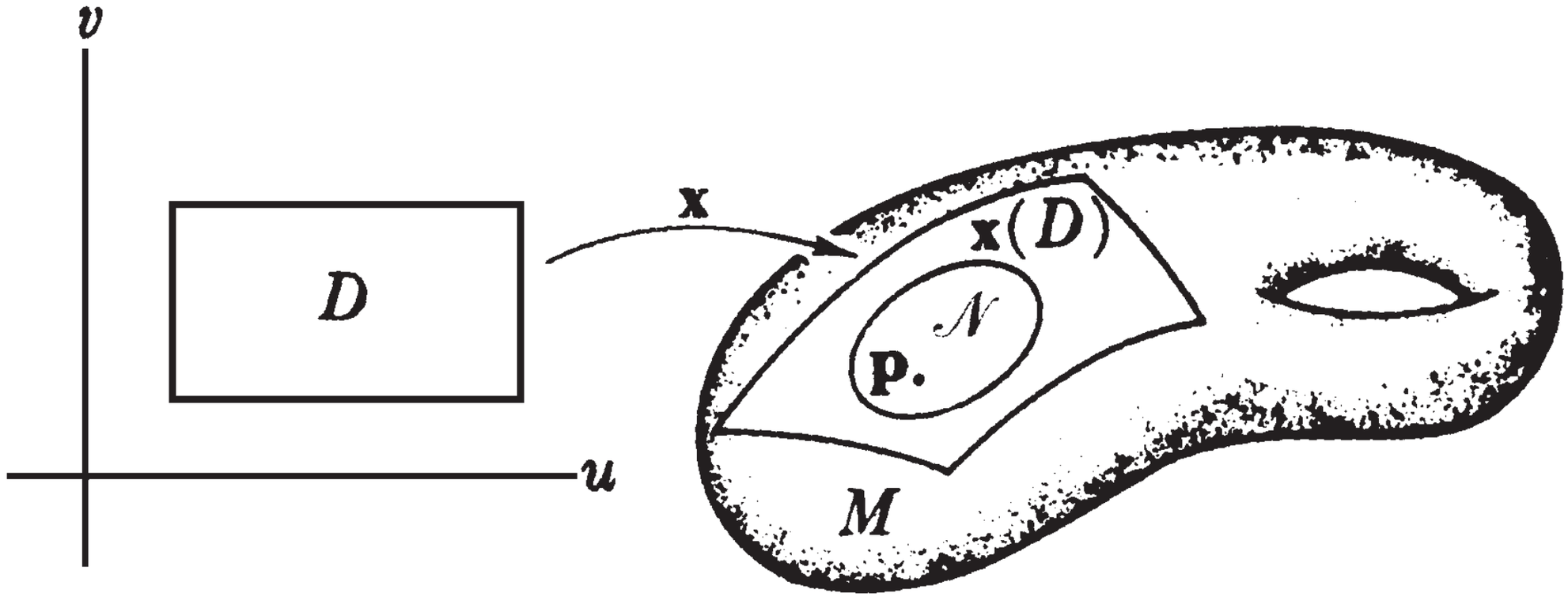

v
u
D
x
x(D)
N
p.
M

FIG. 4.2

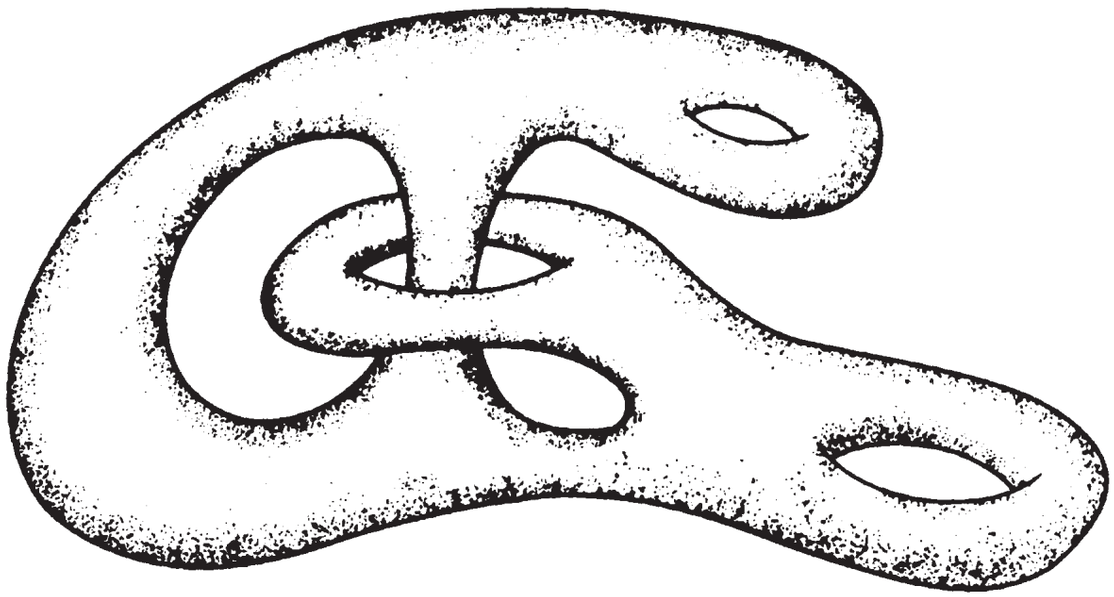
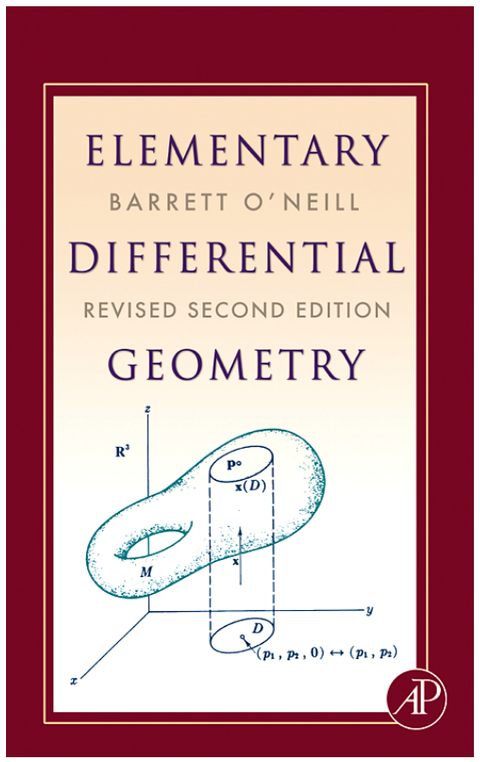
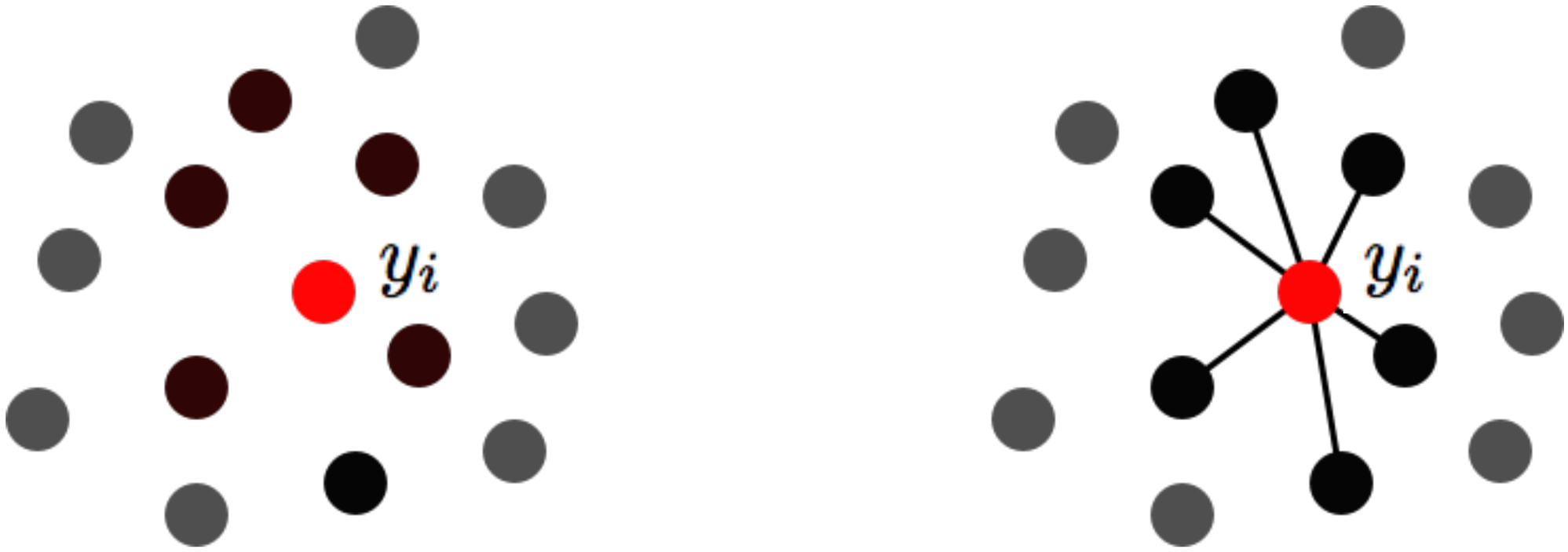


FIG. 4.8


ELEMENTARY
BARRETT O'NEILL
DIFFERENTIAL
REVISED SECOND EDITION
GEOMETRY
z
R³
P₀
x(D)
M
x
y
D
(p₁ , p₂ , 0) ↔ (p₁ , p₂)
x
AP

Multidimensional Space



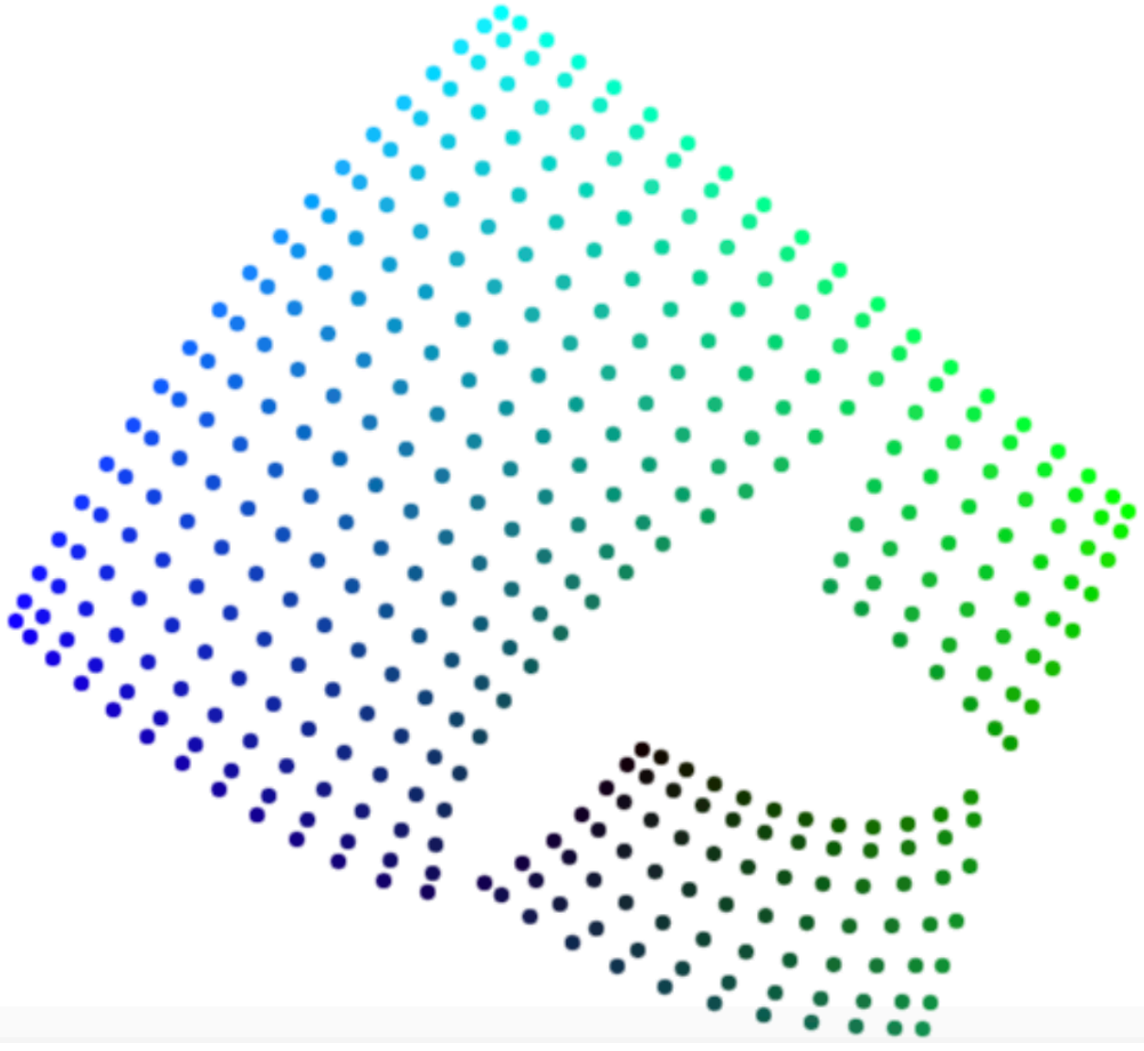
$$y_i = \sum_{j \in N_i} \alpha_{ij} y_j \qquad \alpha_{ij} = \frac{1}{|N_i|}$$

We will watch 20 min of this excellent talk (approx: minutes 4-24):

https://www.youtube.com/watch?v=RJVL80Gg3lA&list=UUtXKDgv1AVoG88PLl8nGXmw

How to Use t-SNE Effectively

Although extremely useful for visualizing high-dimensional data, t-SNE plots can sometimes be mysterious or misleading. By exploring how it behaves in simple cases, we can learn to use it more effectively.



Step
470

A square grid with equal spacing between points. Try convergence at different sizes.

Points Per Side 20

Perplexity 10

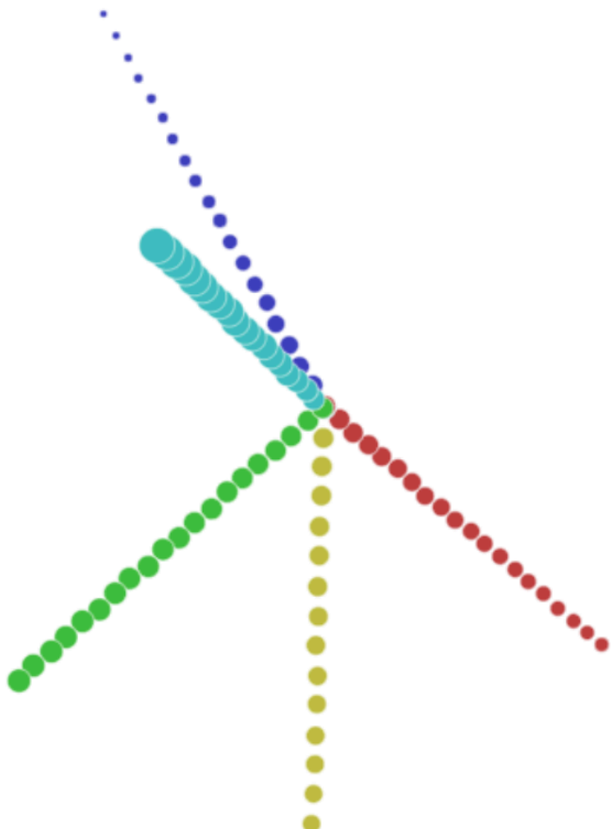
Epsilon 5

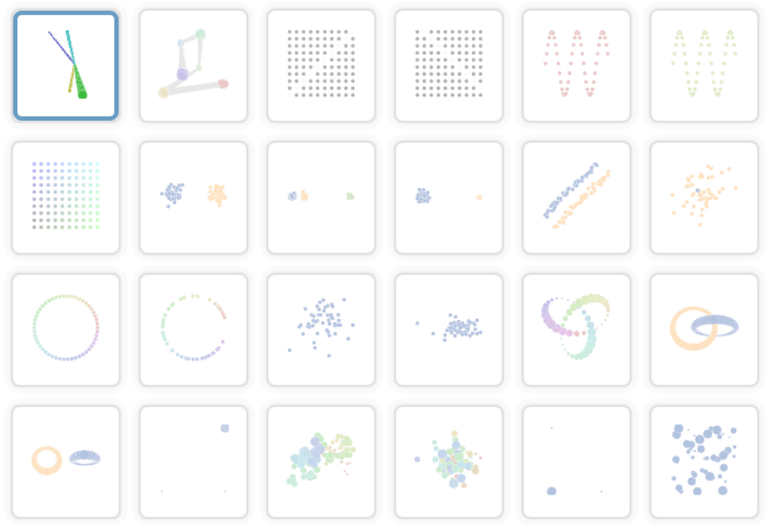
Understanding UMAP

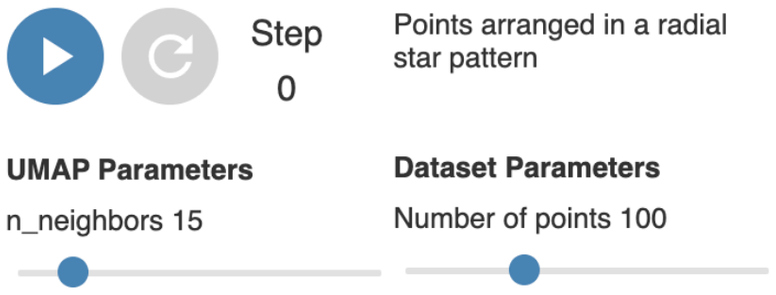
Andy Coenen, Adam Pearce | *Google PAIR*

Dimensionality reduction is a powerful tool for machine learning practitioners to visualize and understand large, high dimensional datasets. One of the most widely used techniques for visualization is t-SNE, but its performance suffers with large datasets and using it correctly can be challenging.

UMAP is a new technique by McInnes et al. that offers a number of advantages over t-SNE, most notably increased speed and better preservation of the data's global structure. In this article, we'll take a look at the theory behind UMAP in order to better understand how the algorithm works, how to use it effectively, and how its performance compares with t-SNE.







TopoMap: A 0-dimensional Homology Preserving Projection of High-Dimensional Data

Harish Doraiswamy, Julien Tierny, Paulo J. S. Silva, Luis Gustavo Nonato, and Claudio Silva

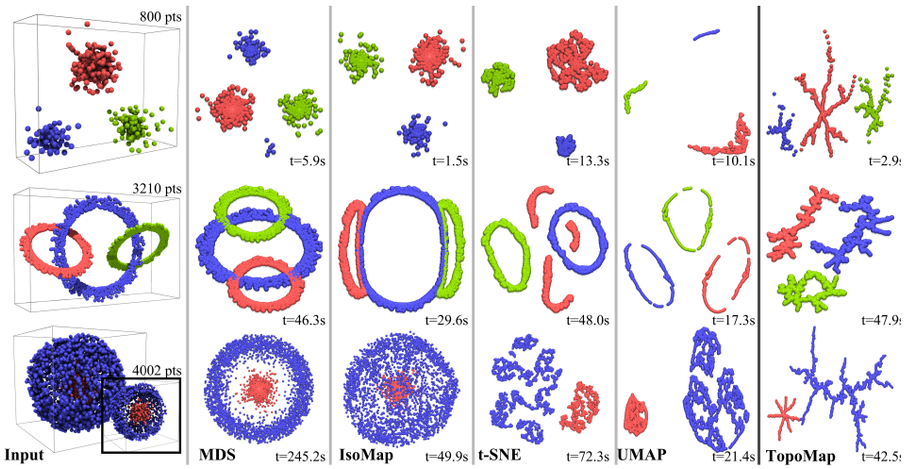


Fig. 1. The result of mapping three dimensional data to a 2D space using geometry preserving projections: Classical MDS, I tSNE, UMAP; and the proposed topology preserving TopoMap method. While geometry preserving methods tend either connected components or mix them up, TopoMap is guaranteed to preserve them, leveraging more reliable analysis.

Abstract— Multidimensional Projection is a fundamental tool for high-dimensional data analytics and visualization. With v exceptions, projection techniques are designed to map data from a high-dimensional space to a visual space so as to pr some dissimilarity (similarity) measure, such as the Euclidean distance for example. In fact, although adopting distinct mathe formulations designed to favor different aspects of the data, most multidimensional projection methods strive to preserve dissi measures that encapsulate geometric properties such as distances or the proximity relation between data objects. However, ge relations are not the only interesting property to be preserved in a projection. For instance, the analysis of particular structure as clusters and outliers could be more reliably performed if the mapping process gives some guarantee as to topological inv such as connected components and loops. This paper introduces *TopoMap*, a novel projection technique which provides topo guarantees during the mapping process. In particular, the proposed method performs the mapping from a high-dimensional spa visual space, while preserving the 0-dimensional persistence diagram of the Rips filtration of the high-dimensional data, ensuri the filtrations generate the same connected components when applied to the original as well as projected data. The presente studies show that the topological guarantee provided by TopoMap not only brings confidence to the visual analytic process but a be used to assist in the assessment of other projection methods.

Index Terms—Topological data analysis, computational topology, high-dimensional data, projection.

1 Introduction

Multidimensional Scaling (MDS) accounts for the problem of embedding data in a Cartesian space while preserving intrinsic properties of the data. A particularly important task in the context of MDS is dimensionality reduction, which aims to map data from a d-dimensional to a k-dimensional Cartesian space where $k << d$. In the context of visualization, where the embedding space is 2D or 3D, MI called multidimensional projection (MDP).

Over the last decades, a multitude of MDP methods ha oped to map high-dimensional data to a visual space wh geometric properties such as the Euclidean distance bet jects. A main issue shared by all those methods is that th of geometric properties can only be guaranteed under v conditions. Thus, errors and distortions are highly likely i mapping, introducing uncertainties to analytical procedu from projection layouts [68]. For instance, structures ob point cloud resulting from a projection such as neighborh might not be the ones existing in the original data, th leading inexperienced practitioners to wrong conclusion

Although a number of alternatives have been proposed analysis of projection layouts more reliable [54], few a developing MDP methods with theoretical guarantees as

- *H. Doraiswamy and C. Silva are with New York University; J. Tierny is with CNRS and Sorbonne Université; P. J. S. Silva is with University of Campinas; and L. G. Nonato is with University of Sao Paulo, Sao Carlos.*
- *E-mail: {harishd,csilva}@nyu.edu, julien.tierny@sorbonne-universite.fr, pjssilva@ime.unicamp.br, gnonato@icmc.usp.br*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

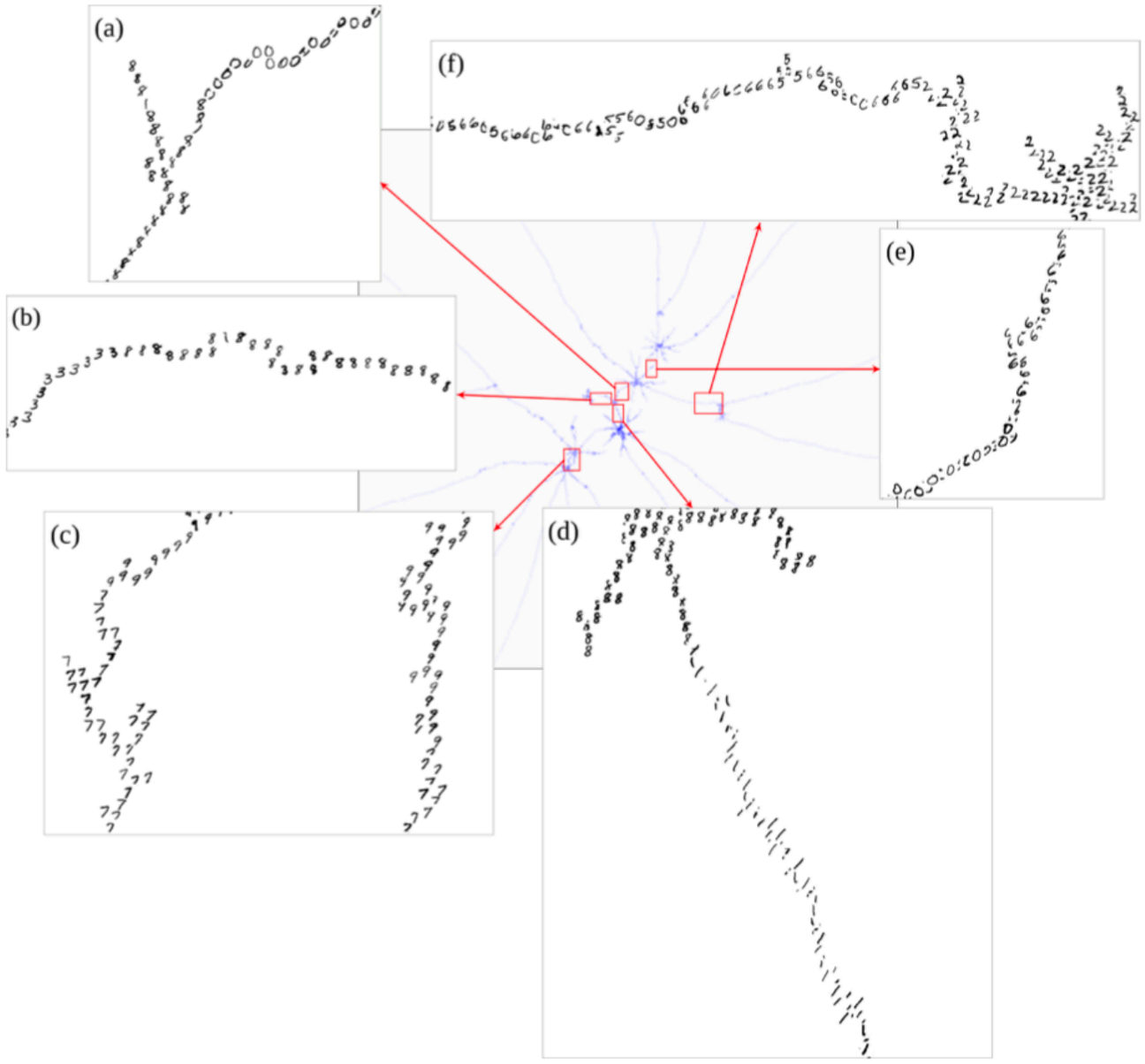


Fig. 7. Mnist data projected using TopoMap (using cosine distance). Transitions between the different starred ensembles clusters: (a) 0 and 8. (b) 3 and 8. (c) 7 and 9. (d) 1 and 8. (e) 0 and 6. (f) class 2 while being a cluster, is far from 0 and is connected to it via outliers.

Please cite the journal version instead of this preprint:
Spathis, D., Passalis, N., & Tefas, A. (2018). Interactive dimensionality reduction using similarity projections. *Knowledge-Based Systems.*

Interactive dimensionality reduction using similarity projections

Dimitris Spathis *[a,b], Nikolaos Passalis[b], Anastasios Tefas[b]

[a] *Department of Computer Science and Technology, University of Cambridge, UK*
[b] *Department of Informatics, Aristotle University of Thessaloniki, Greece*

Abstract

Recent advances in machine learning allow us to analyze and describe the content of high-dimensional data like text, audio, images or other signals. In order to visualize that data in 2D or 3D, usually Dimensionality Reduction (DR) techniques are employed. Most of these techniques, e.g., PCA or t-SNE, produce static projections without taking into account corrections from humans or other data exploration scenarios. In this work, we propose the *interactive Similarity Projection (iSP)*, a novel interactive DR framework based on similarity embeddings, where we form a differentiable objective based on the user interactions and perform learning using gradient descent, with an end-to-end trainable architecture. Two interaction scenarios are evaluated. First, a common methodology in multidimensional projection is to project a subset of data, arrange them in classes or clusters, and project the rest unseen dataset based on that manipulation, in a kind of semi-supervised interpolation. We report results that outperform competitive baselines in a wide range of metrics and datasets. Second, we explore the

*Corresponding author. Work done while at the Aristotle University of Thessaloniki.

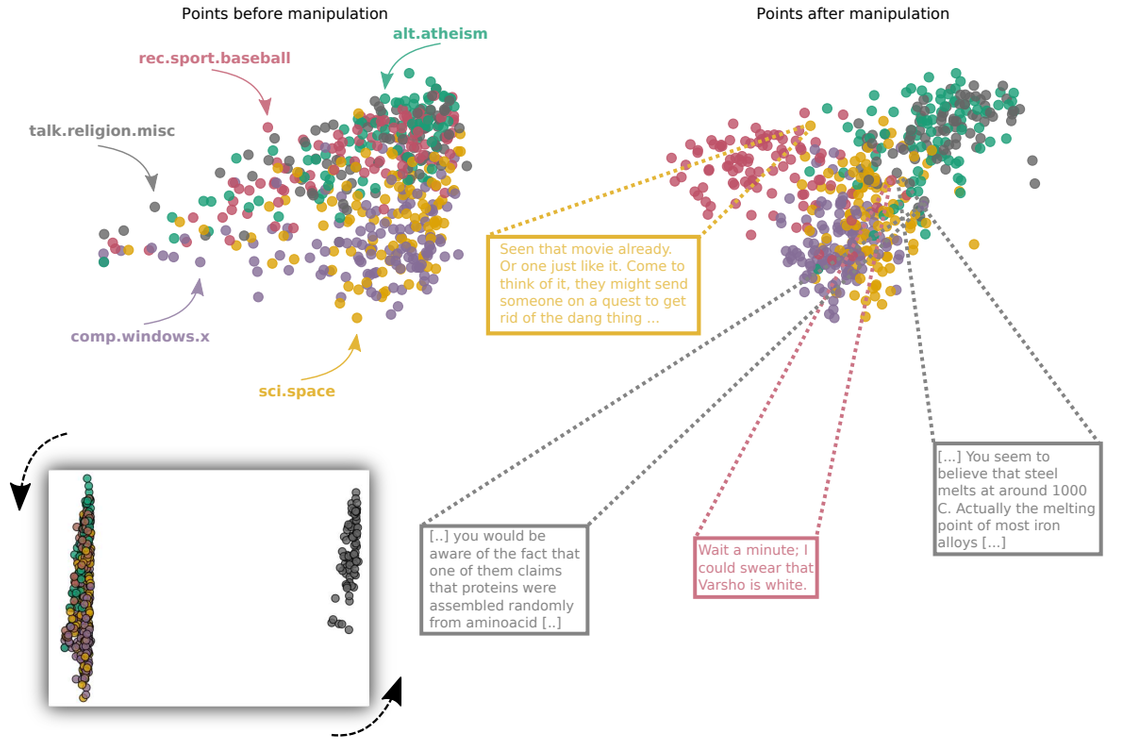
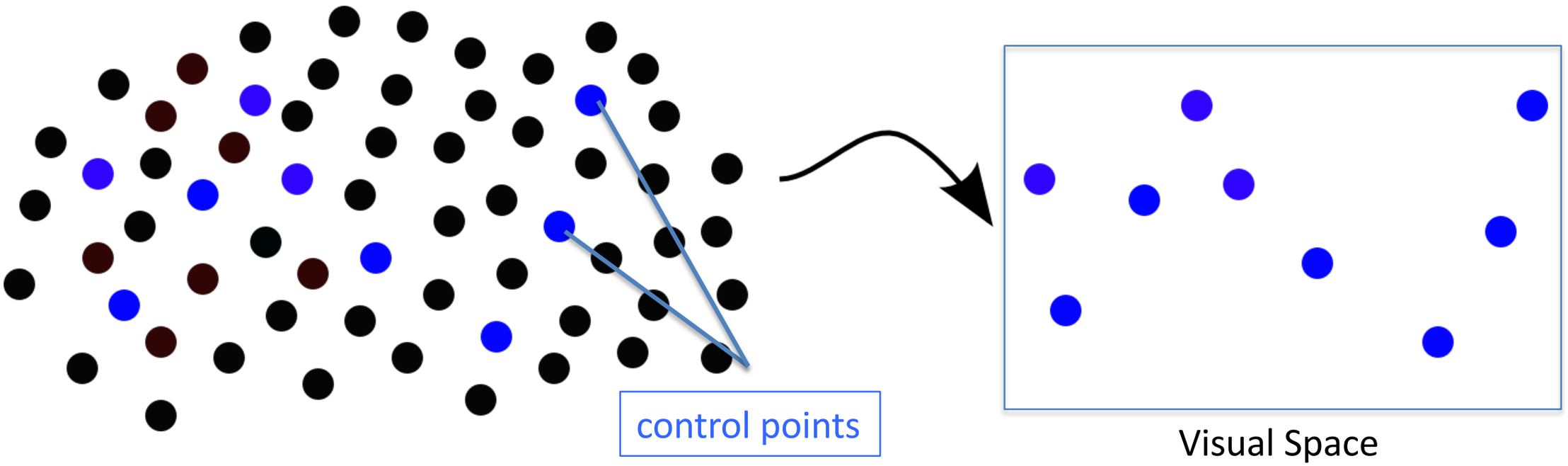


Figure 8: Points of Newsgroup before and after manipulation along with some highlighted documents. Kernel version.

Name	Source	Complexity	Local/Global	Steerable
Kruskal	Kruskal (1964)	$O(n^2)$	Global	
Classical MDS	Torgeson (1965)	$O(n^3)$	Global	
Smacof	de Leeuw (1977)	$O(cn^2)$	Global	
FastMap	Faloutsos and Lin (1995)	$O(n)$	Global	
Chalmers	Chalmers (1996)	$O(cn)$	Global	
Isomap	Tenenbaum et al. (2000)	$O(n^3)$	Global	
LLE	Roweis and Saul (2000)	$O(n^2)$	Local / Global	
L-MDS	de Silva and Tenenbaum (2003)	$O(k^3 + kn)$	Local / Global	
t-SNE	Maaten and Hinton (2008)	$O(n^2)$	Global	
LSP	Paulovich et al. (2008)	$O(k^2 + n^2)$	Global	✓
Glimmer	S. Ingram et al. (2009)	$O(cnlog(m))$	Local / Global	○
PLMP	Paulovich et al. (2010)	$O(k^2 + n)$	Global	✓
PLP	Paulovich et al. (2011)	$O(k^2 + n^2)$	Local	✓
LAMP	Joia et al. (2011)	$O(nk^2)$	Local	✓

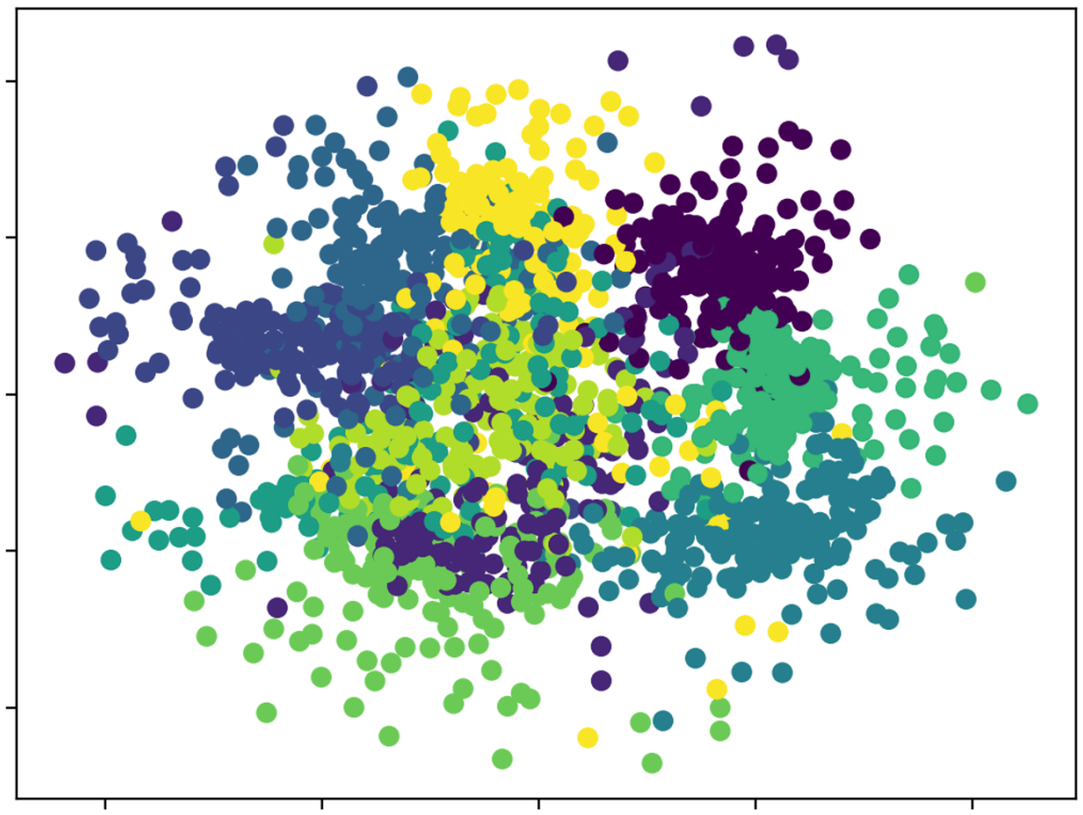
n - number of instances; k - number of pivot/control points;
m - data dimension; c - number of iteractions

Constraints for the homogeneous systems are given by control points in the visual, which can be interactively displaced to steer the projection process.

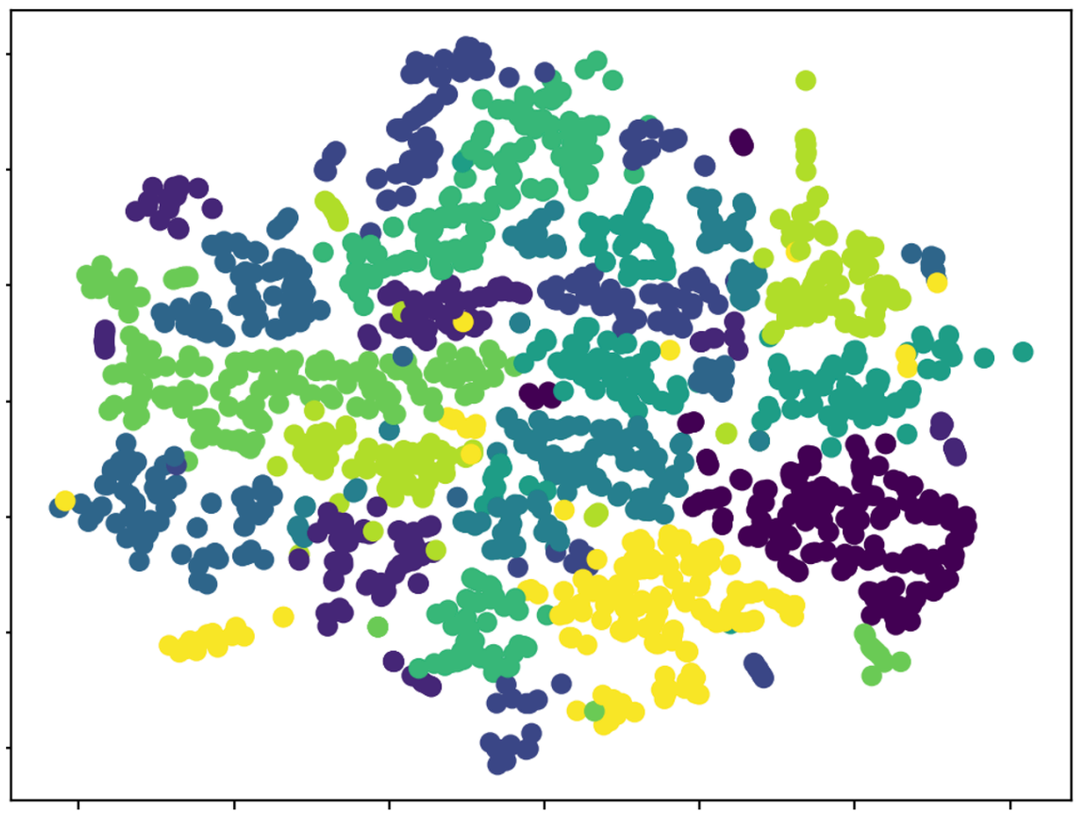


Project Idea:
Evaluating Projection Techniques

How should an MPD layout be interpreted?



Lamp



t-SNE