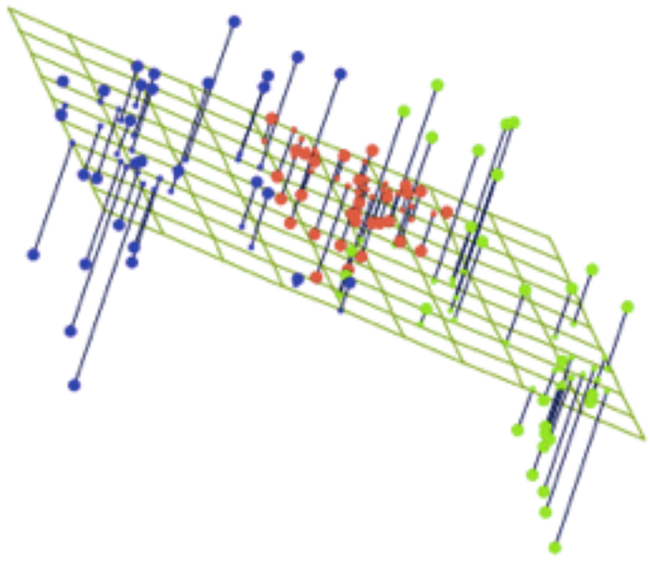
CS109 Data Science
Classification & PCA

Hanspeter Pfister, Joe Blitzstein, and Verena Kaynig



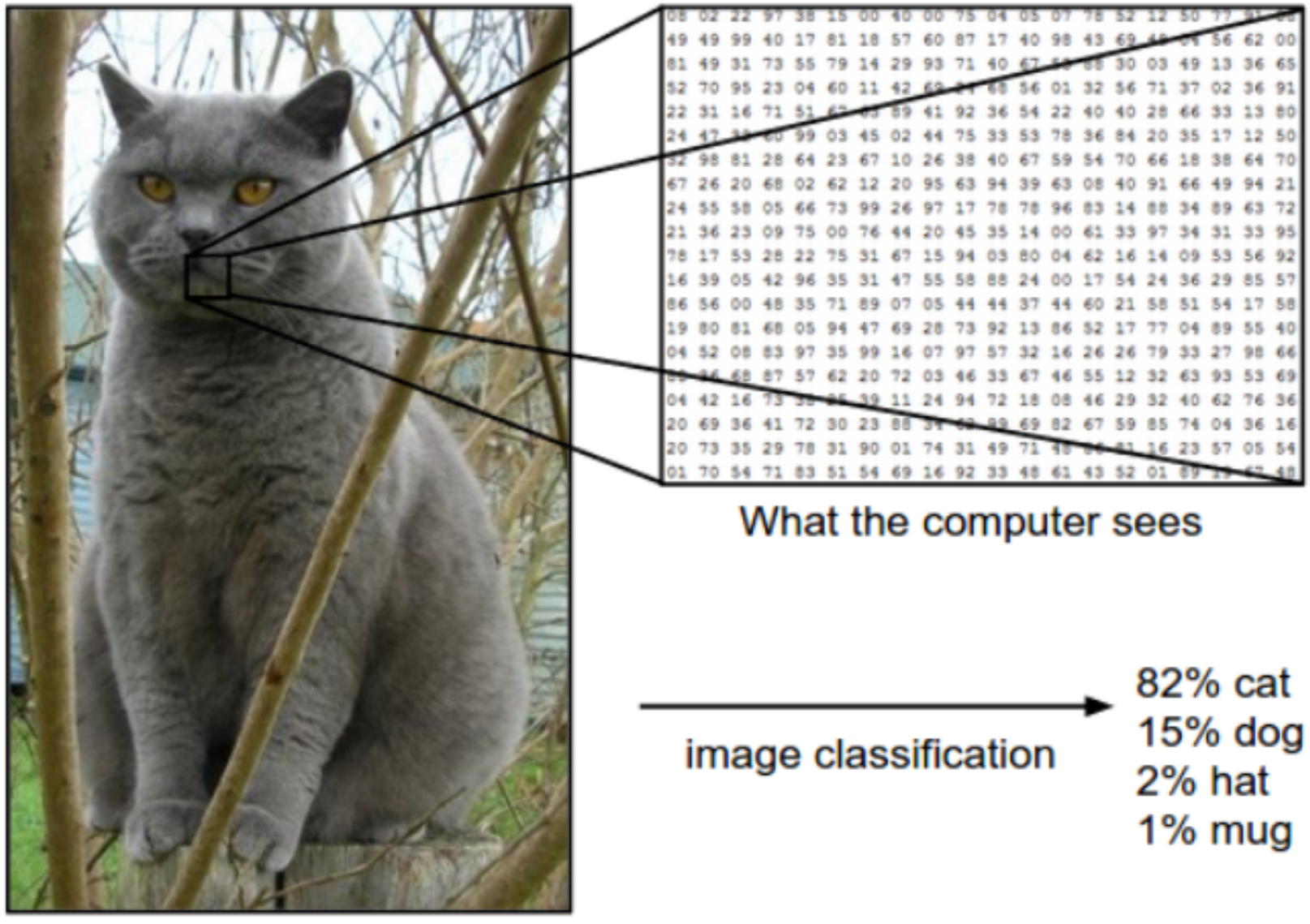
This Week

- HW2 due on Thursday
- HW1 grades will be out Sunday night
- HW1 solution is available next Tuesday

Classification

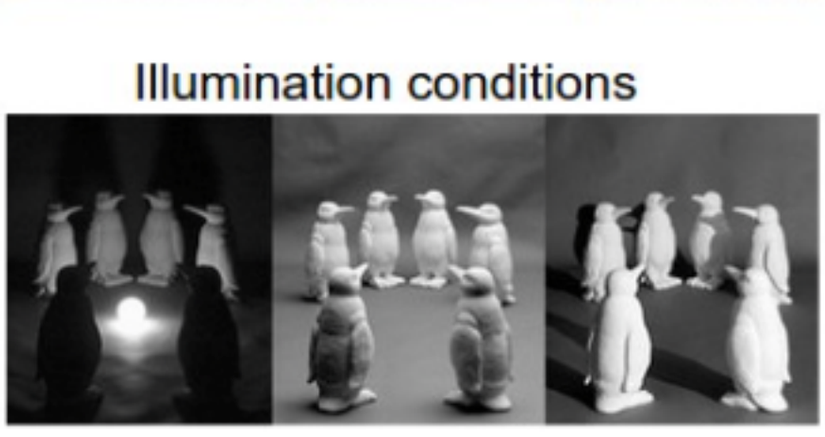
↳ Take data + assign labels to it.

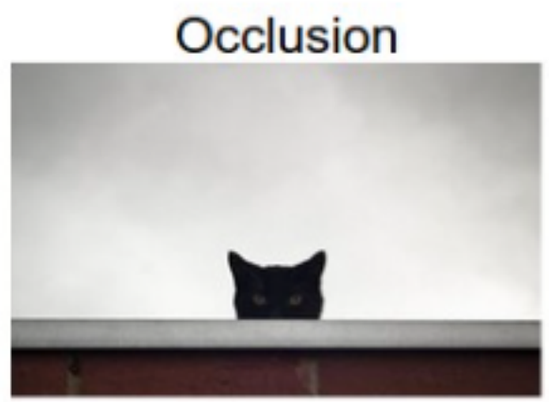
Classification



Alex Krizhevsky, Stanford

Problems





Alex Krizhevsky, Stanford

Training Data

Do Need a lot of datasets.

Alex Krizhevsky, Stanford

Machine Learning

- Input: A **training set** of N data points, each labeled with one of K different classes.

Ex cat, dog, person.

- Learning: Use the training set to learn what every one of the classes looks like.

- Evaluation: Predict labels for a **test set** of data and compare the true labels (ground truth) to the ones predicted by the classifier.

Machine Learning

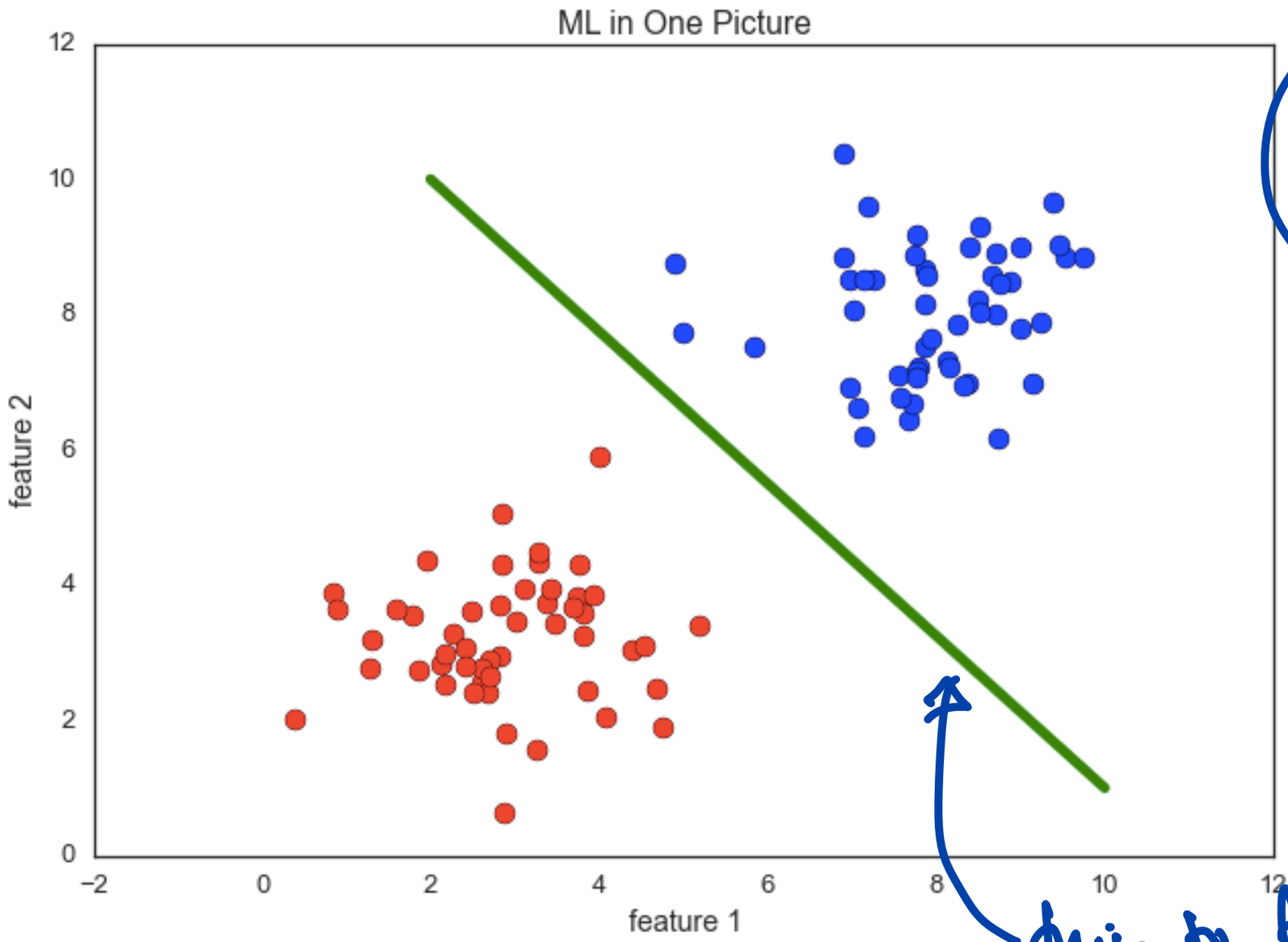
- Make predictions for new data points
  - data has labels
  - supervised learning: kNN, SVM, Decision Trees, Random Forests, Bagging, Boosting, etc.
- Find patterns in the data
  - data has no labels
  - unsupervised learning: PCA, MDS, Clustering

Supervised

not supervised.

Might be too big.

Classification



scikit-learn too.

x: data points

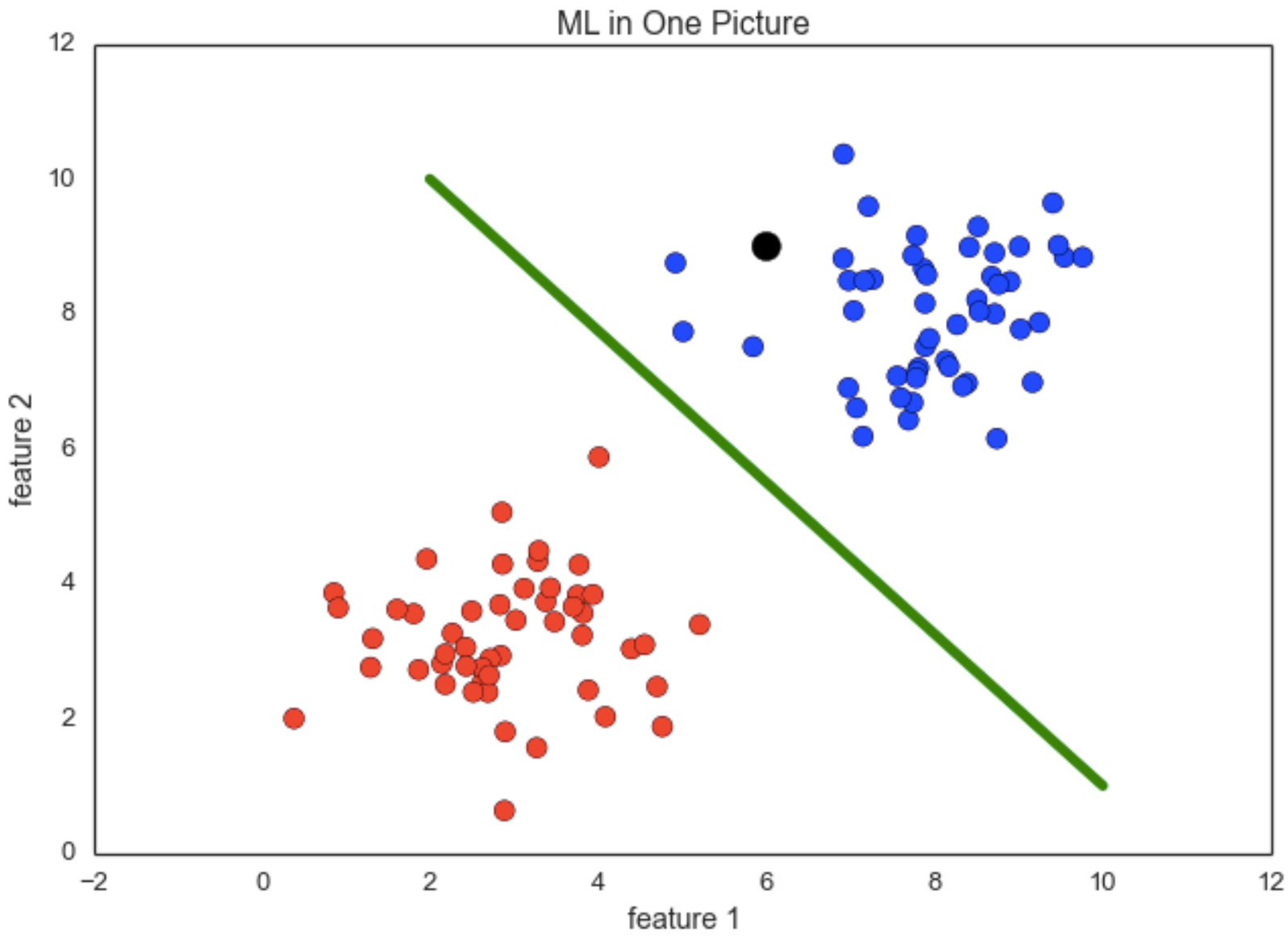
y: labels

features

decision boundary

trying to find this.

Classification



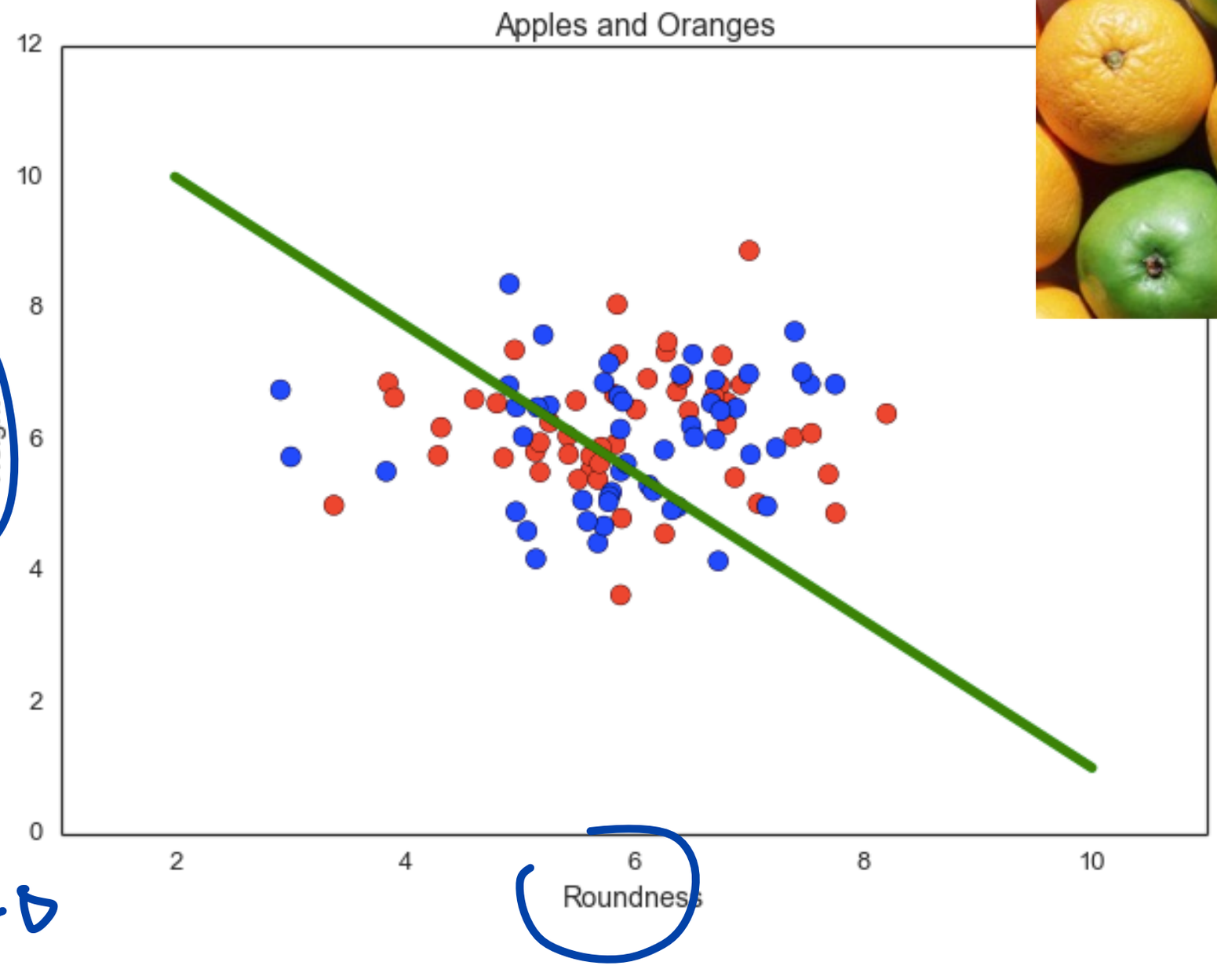
x: data points

y: labels

features

decision boundary

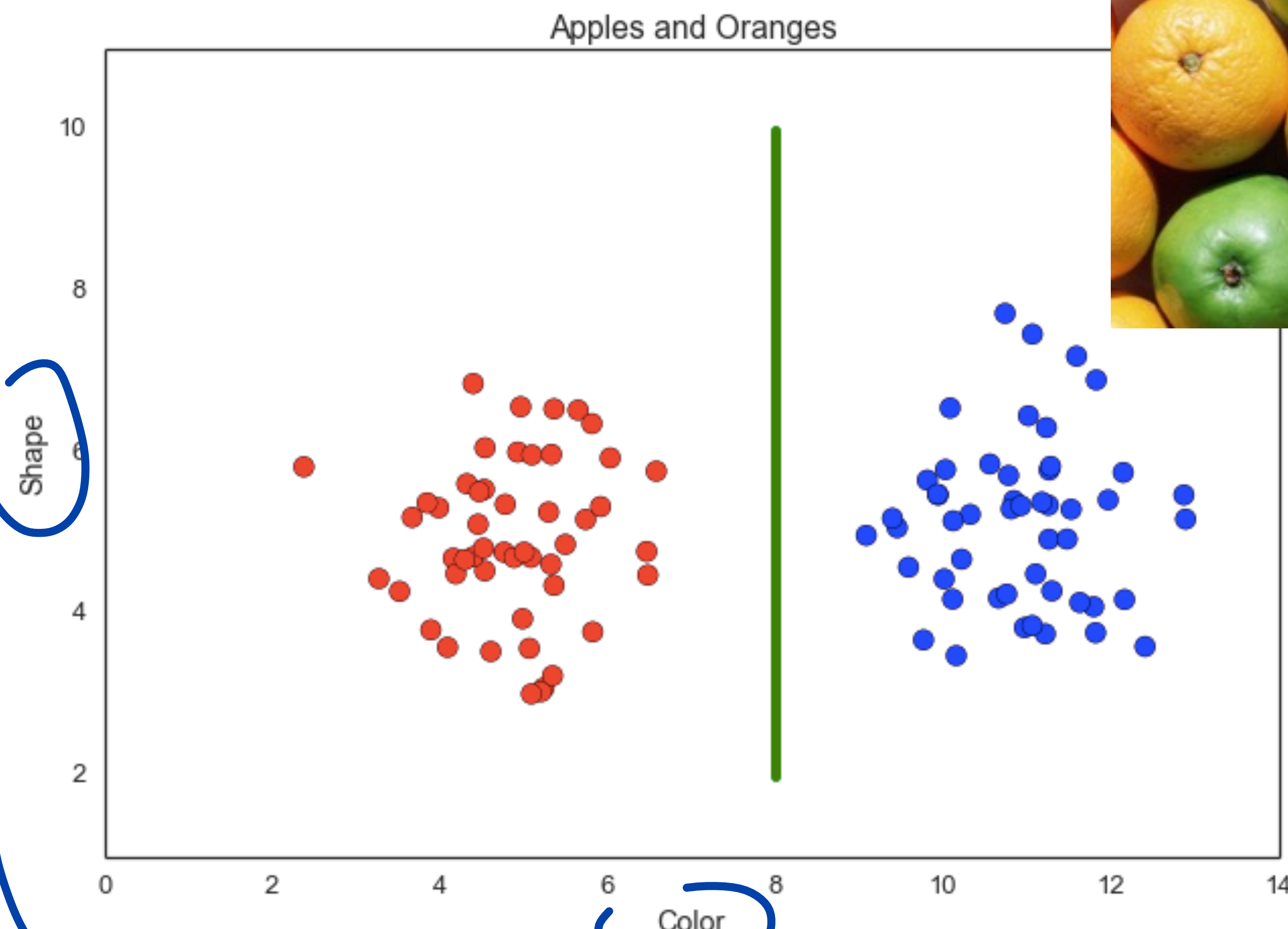
Feature Selection



Not Good. Similar

Art, not science.
Cross-validation.

Feature Selection

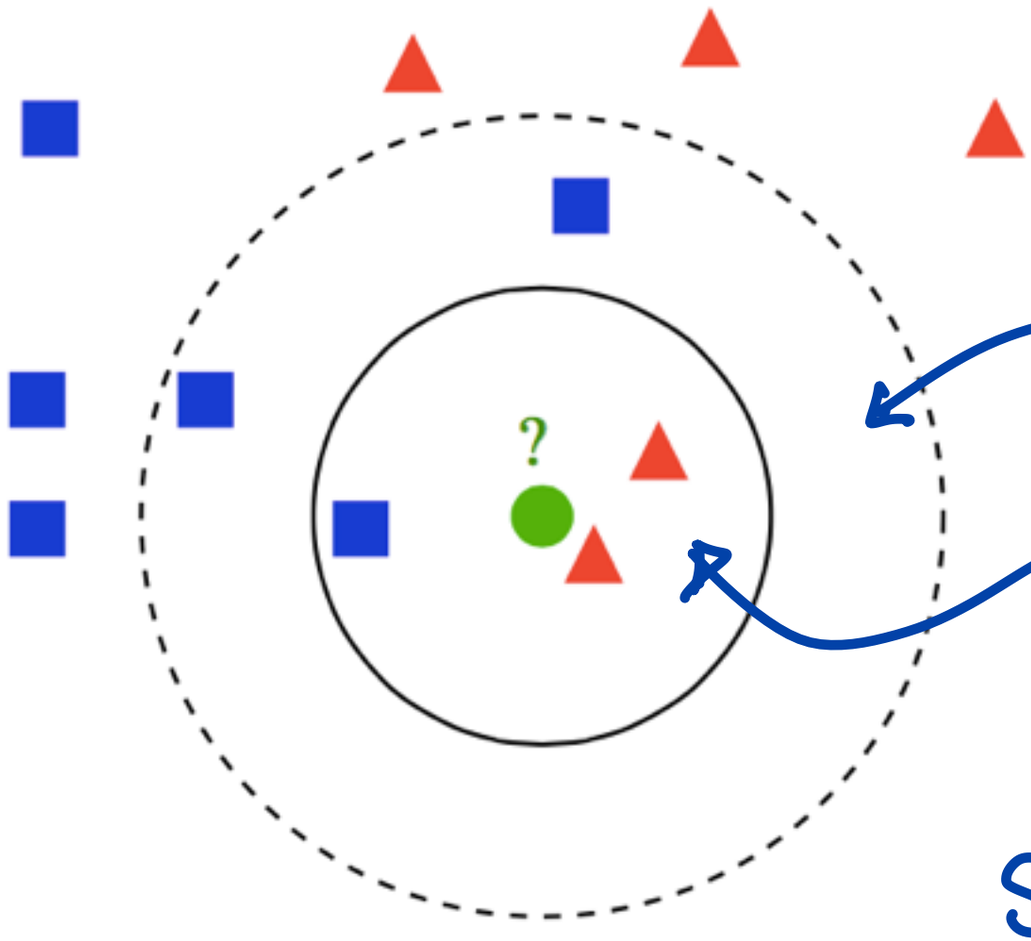


Much better

Nearest Neighbor Classifier

Nearest Neighbor

Predict class of new data point by majority vote of K nearest neighbors



always odd to have a winner.

dotted will be blue square.

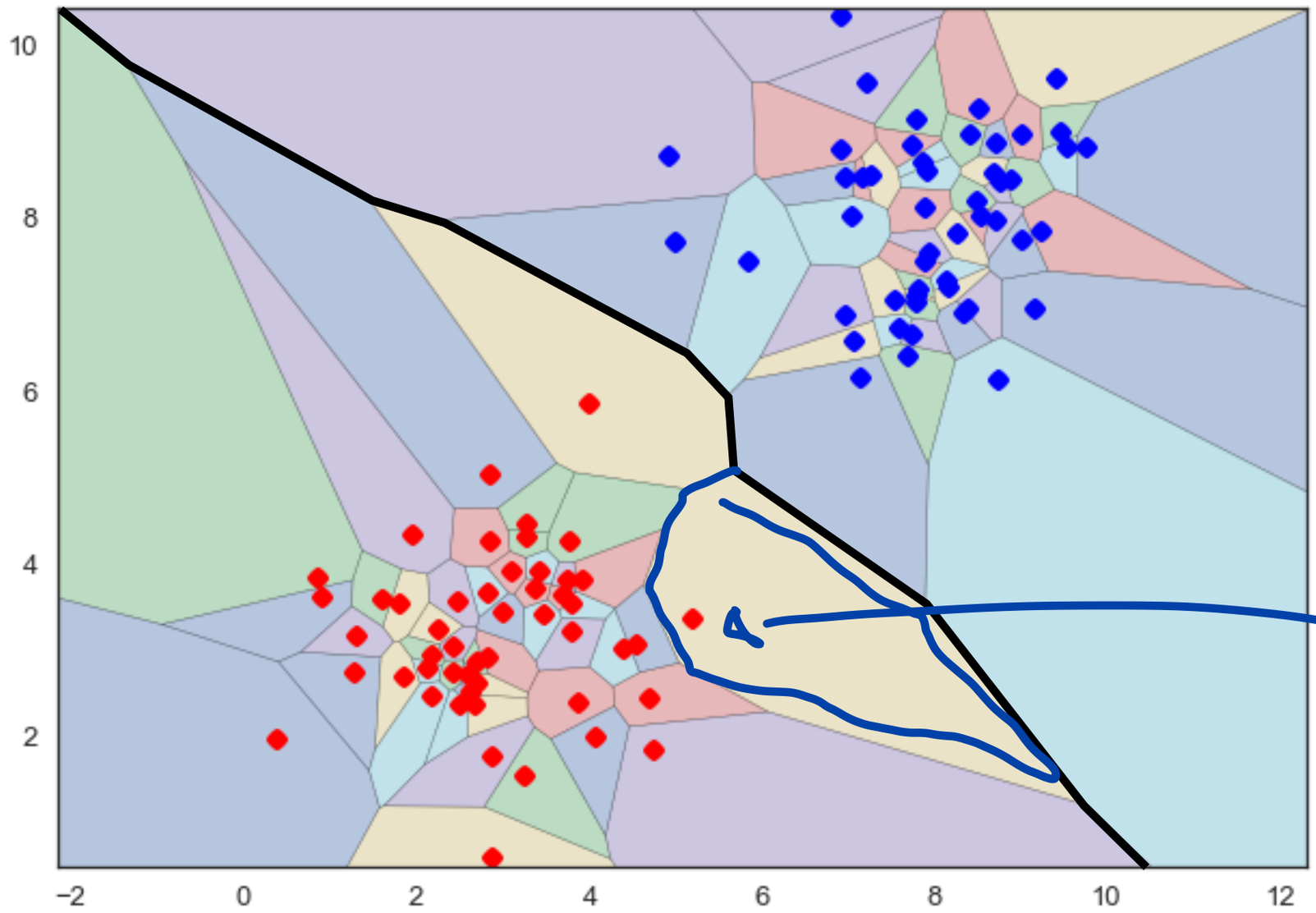
Will be a red triagle.

Majority voting.

Size matters.

1-Nearest Neighbor

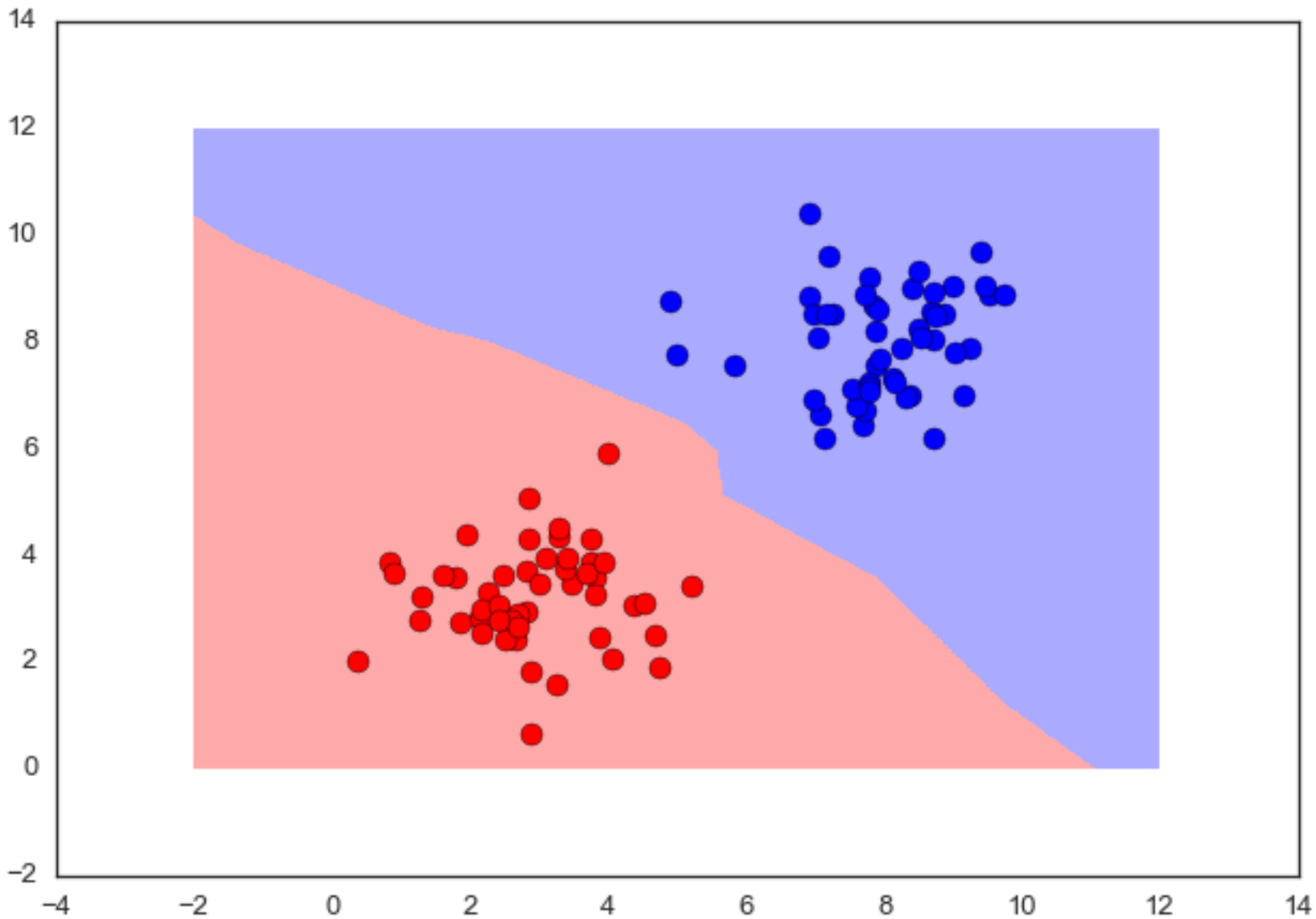
One.



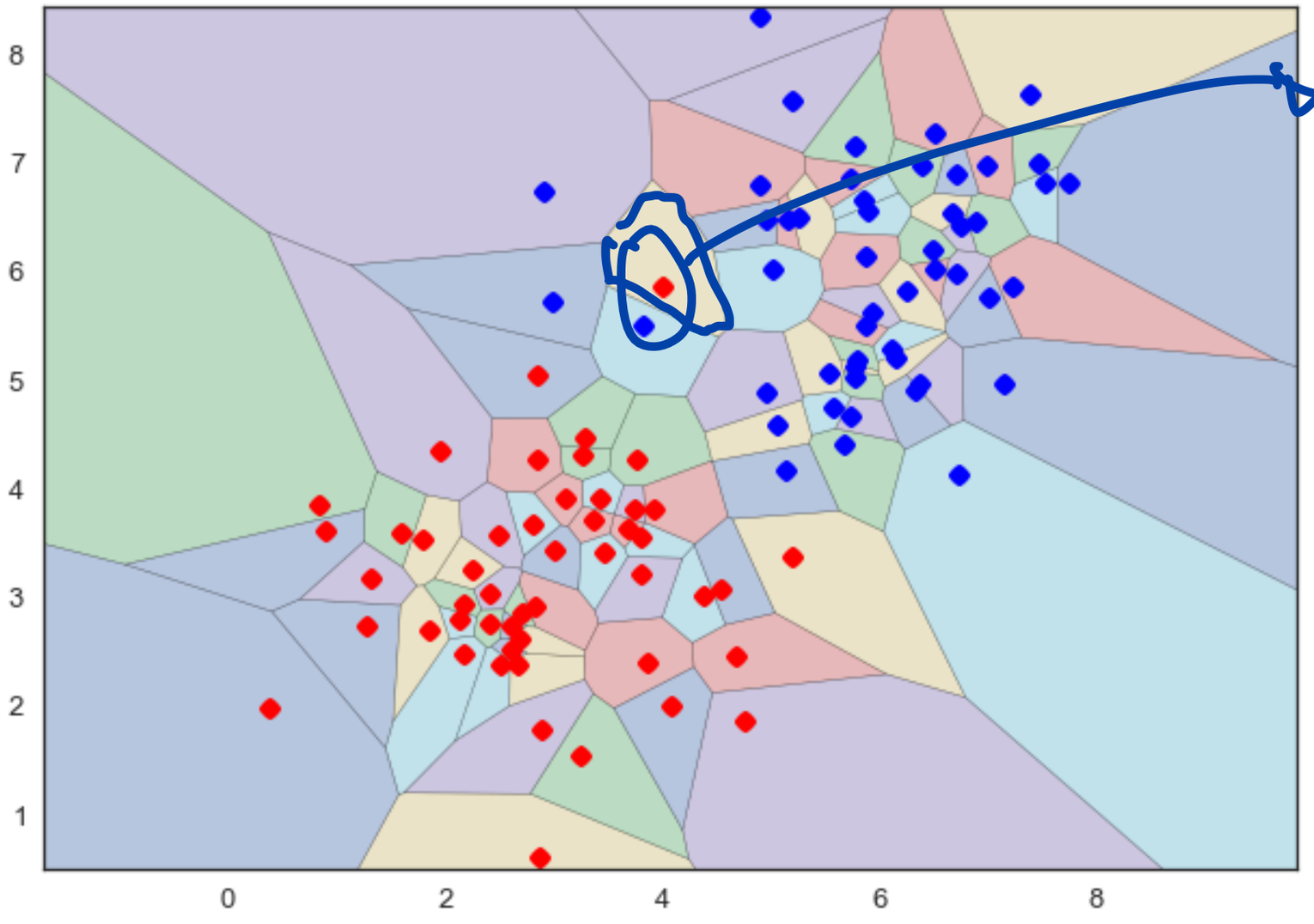
* Pick size based on performance of classifier.

region where this is closest point.

1-Nearest Neighbor



1-Nearest Neighbor

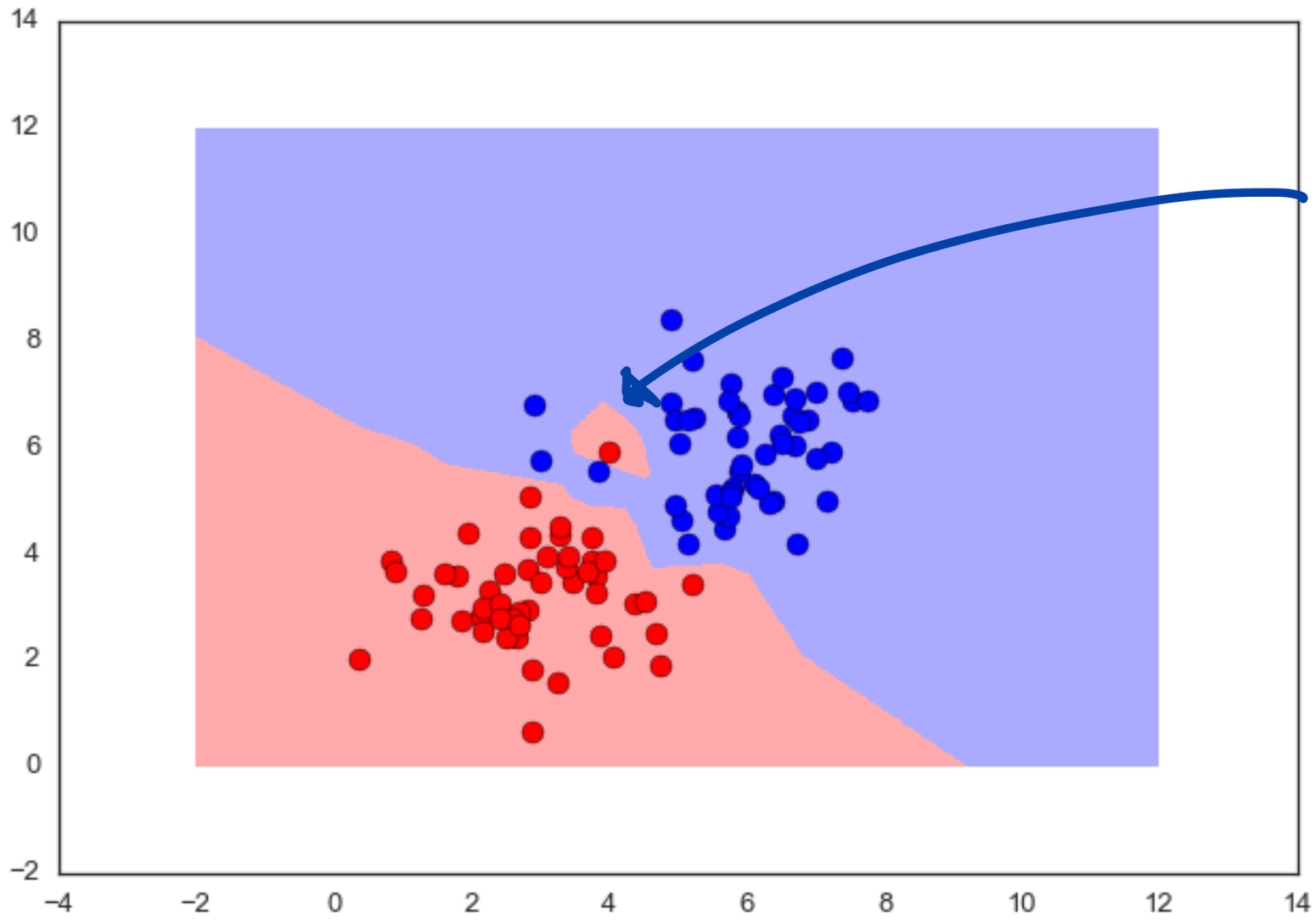


outlier.

ML algorithm doesn't know

Anything within is red.

1-Nearest Neighbor



1-NN Properties

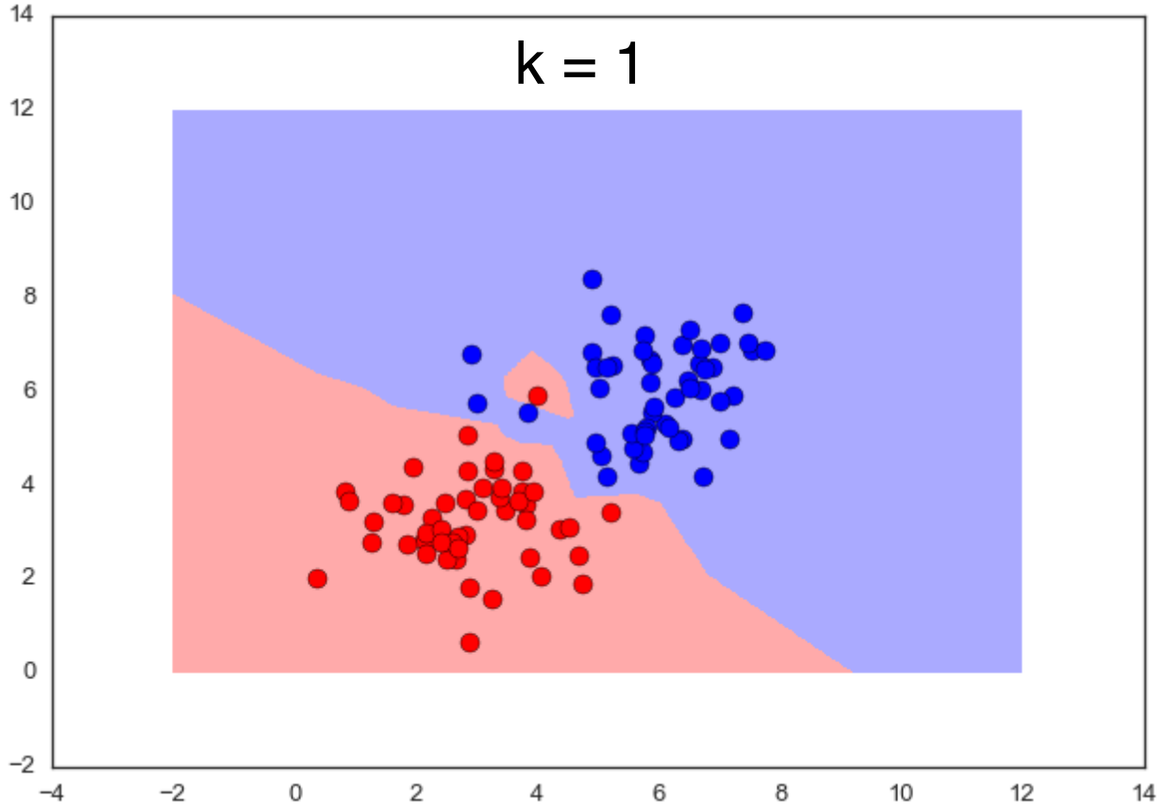
One

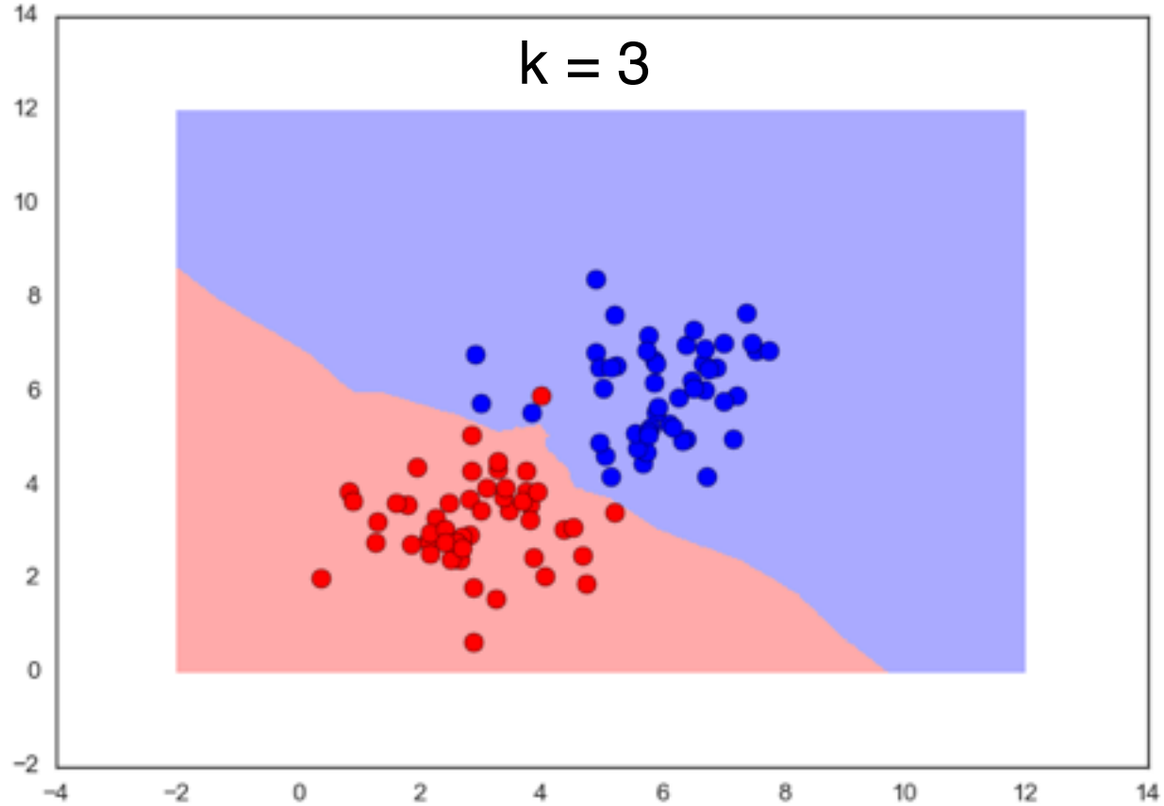
- High time in testing.
- Ppl care about testing time.

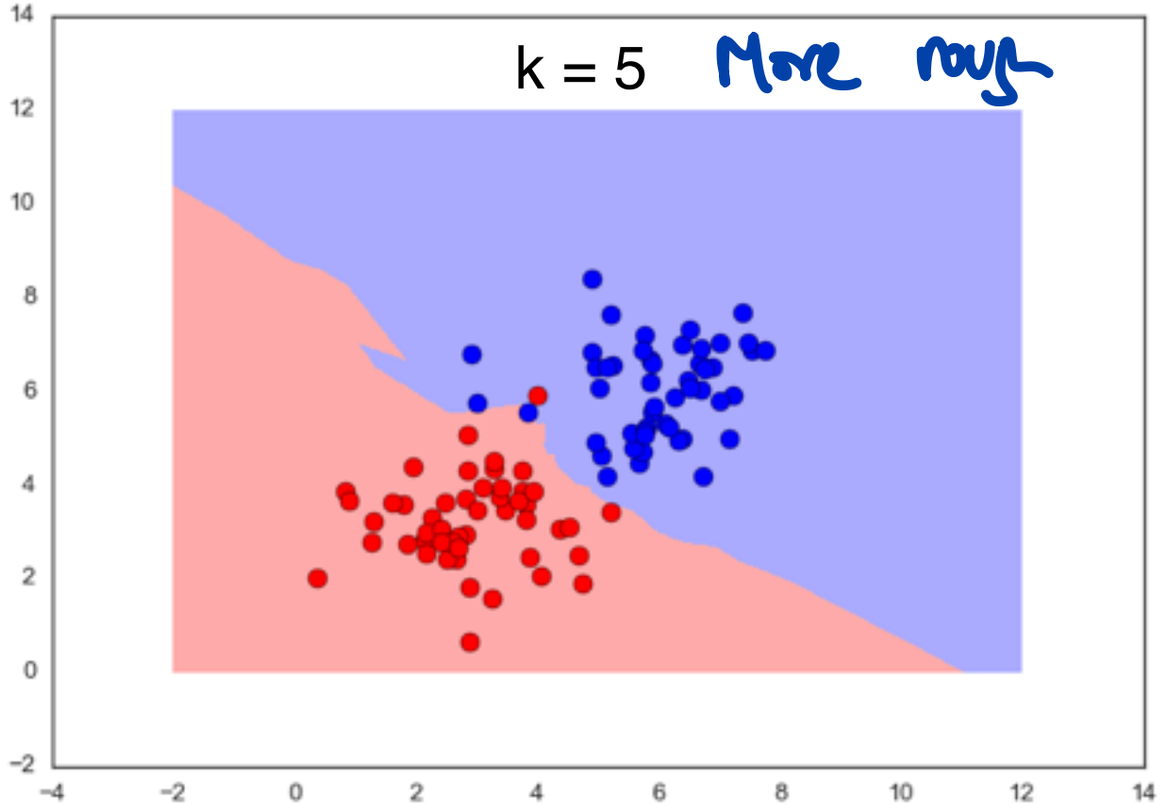
- Simple and quite good for low dimensional data
- "Rough" decision boundary, may have "islands"
- Training complexity for N data points?
  - Order 1. Basically not work
- Test complexity for M data points?
  - Have to go through all
  - O(MxN).
- Error on training set?
  - Zero.
- Variance? Bias?
  - How much does the algorithm change with different training sets?
    - High.
  - High/low.
    - given new data?
      - Low. Performance good.

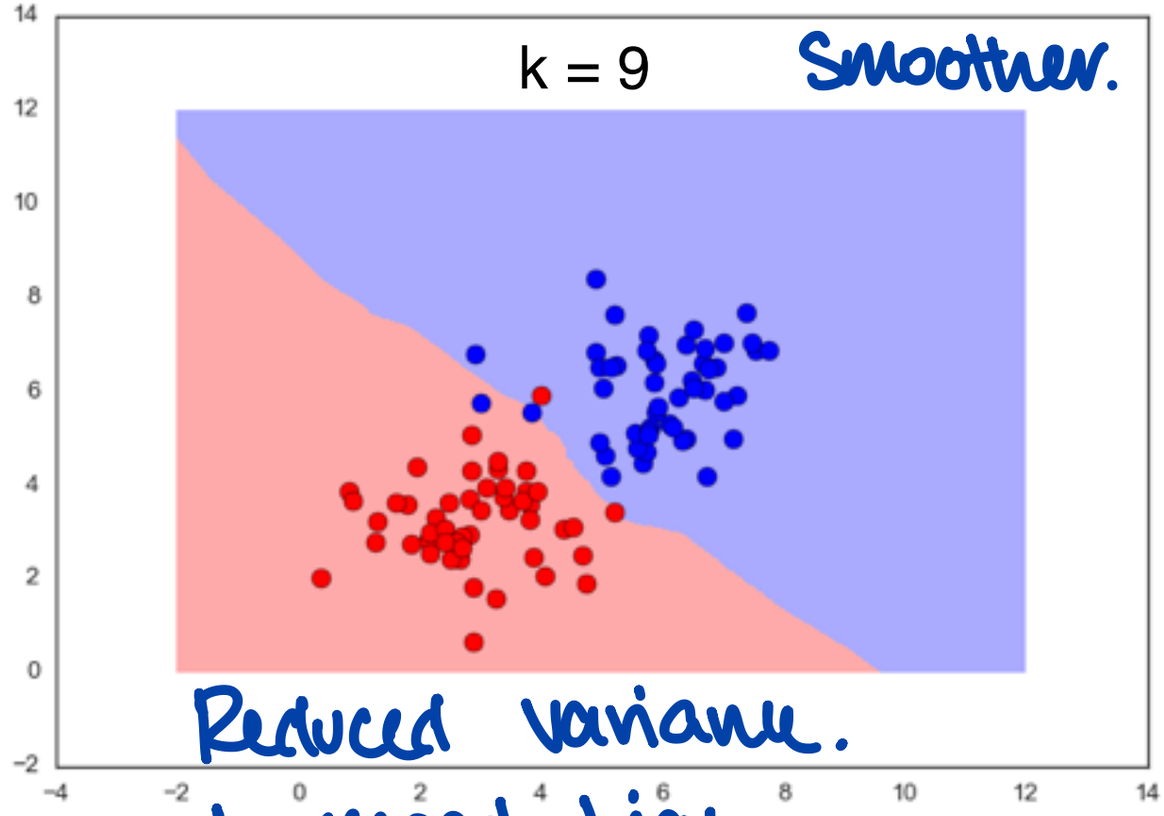
Quite good for how simple it is.

- Have to run through all n.
- All the work during test; none w/ training









Reduced variance.
Increased bias.

k-NN Properties

- Gets rid of "islands"
- If k is too large, the boundary may become too smooth
- Lower variance, but increased bias
- How do we choose the ideal k?

→ hyper parameter.

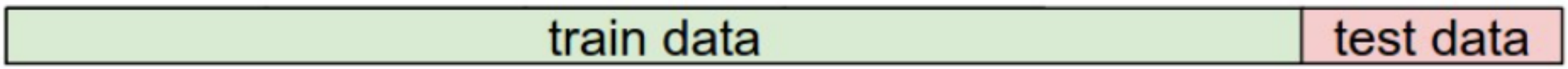
↳ Cross-validation

↳ split into training/test

↳ Decide k; test size; distance function; probabilistic?

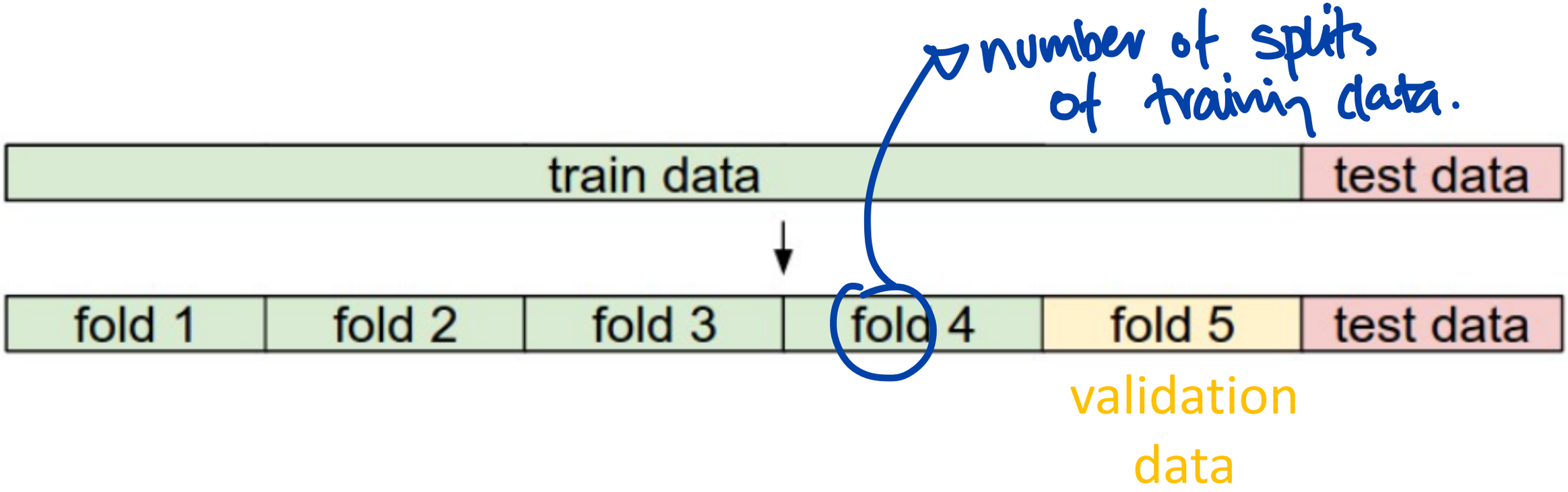
Validation

- Train on training data, test on test data
- Pick the k with the lowest test error



- Training data: train classifier
- Test data: measure performance

Cross-Validation

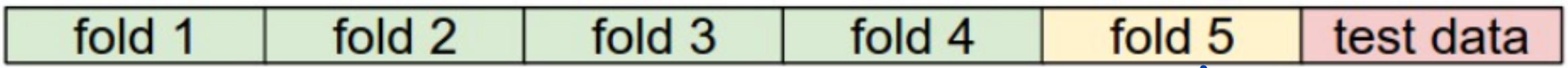


- Training data: train classifier
- Validation data: estimate (hyper) parameters (k)
- Test data: measure performance

↳ Get optimal parameter before going to test data.

Choosing # of folds is not set. 5-fold + 10-fold are typical. Depends on # size of data.

Cross-Validation



Maybe even 3-fold.

1. Iterate over choice of validation fold ← #s.
2. For all parameter values:
   a. Train with training data (folds 1-4).
   b. Validate with validation data
3. Average the parameters with best performance on validation data } repeat w/ each fold as validation then average.

The test data is NOT used to determine the parameters!

Cross-Validation

- Take best parameters 
- Train on training data and validation data together 
- Test performance on test data 

Evaluate on the test set only **a single time**, at the very end!

Not generalizable if not done.

No guarantee on future data sets.

CIFAR-10 Data Set

airplane

automobile

bird

cat

deer

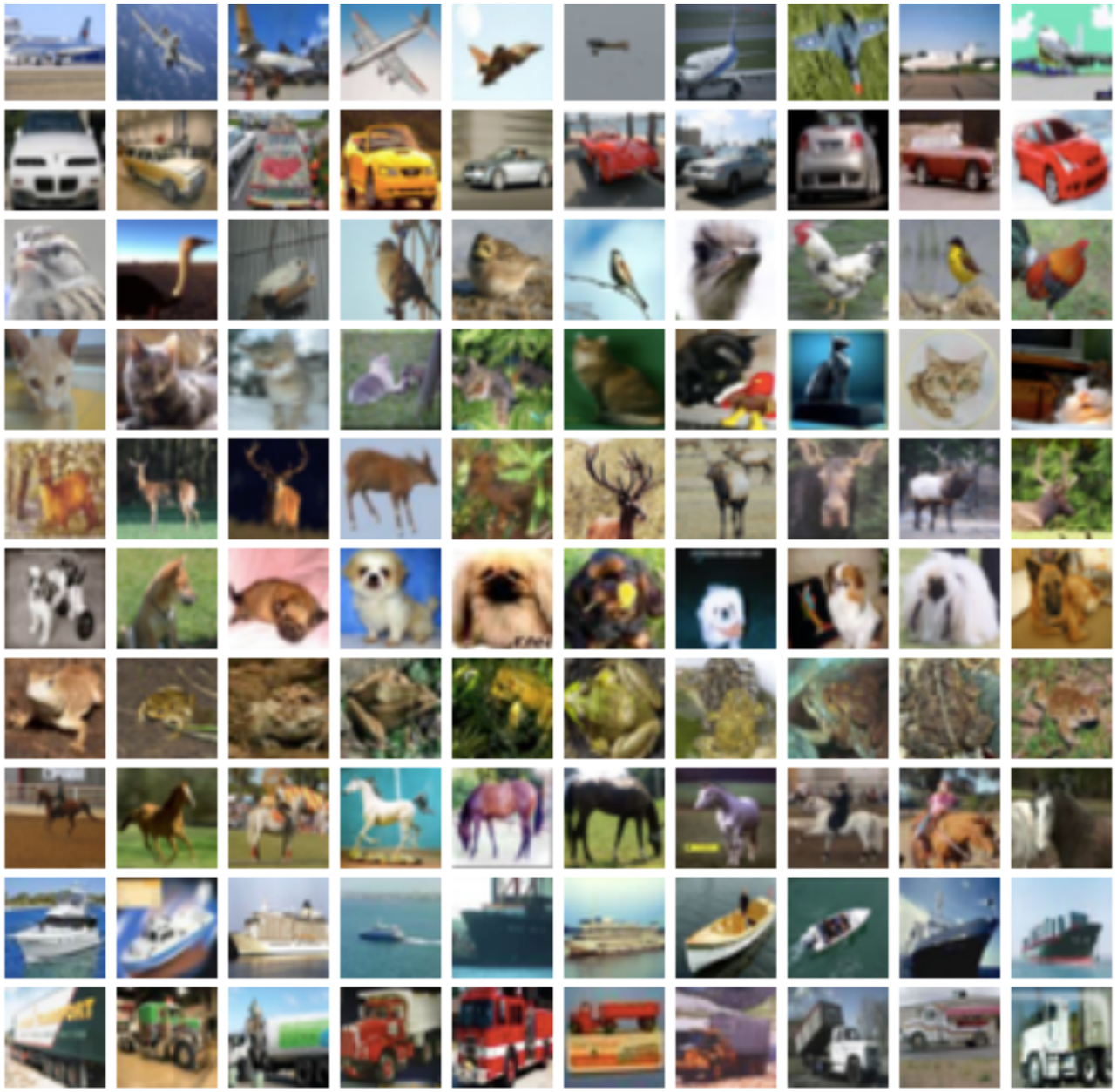
dog

frog

horse

ship

truck



60,000 images
32x32 pixels

10 classes

Training set:
50,000 images

Test set:
10,000 images

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton

High-Dimensional Data

Each image is a vector in 32 x 32 x 3 = 3,072 dimensional space

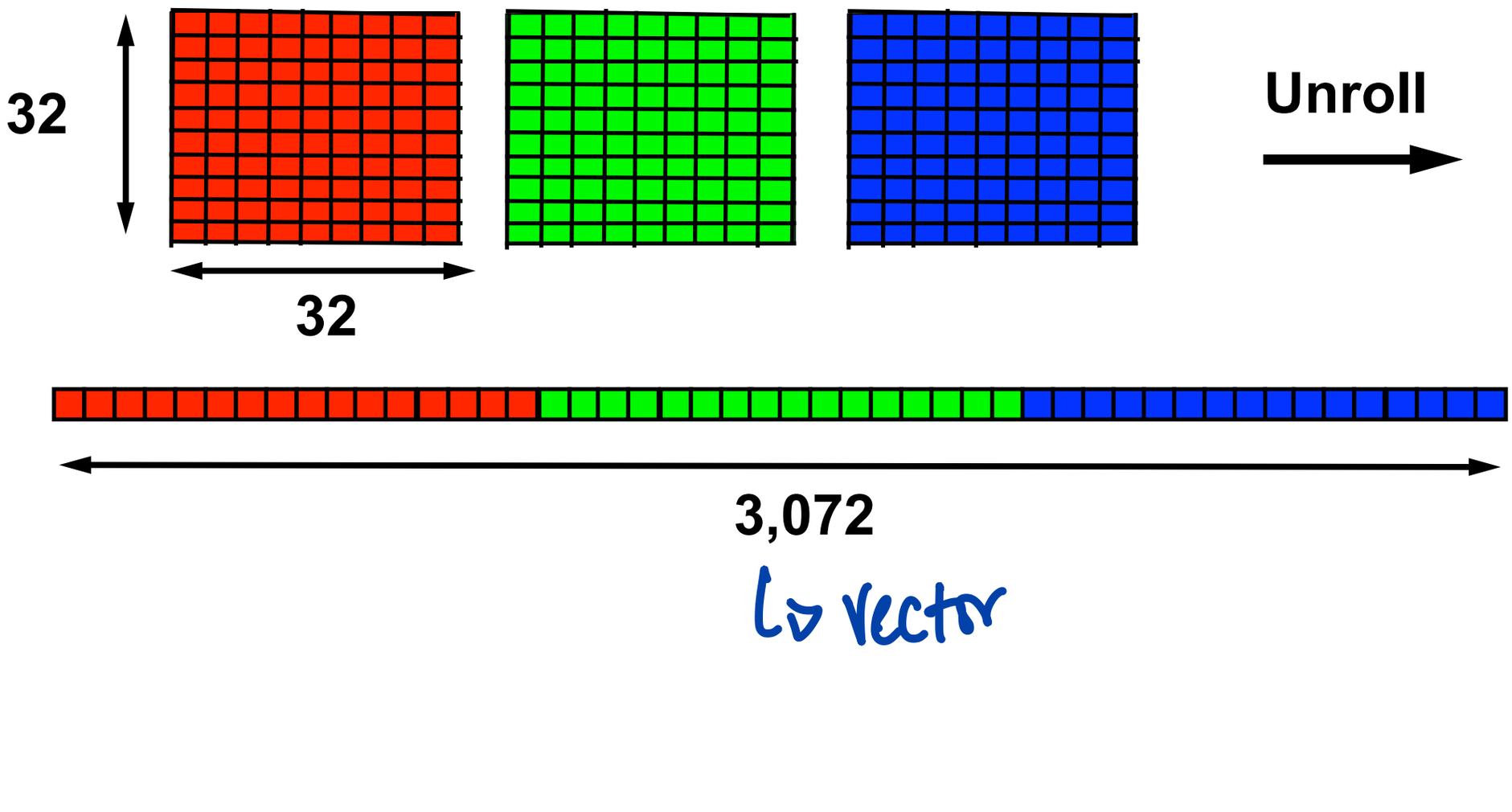


Image Comparison

test image			
56	32	10	18
90	23	128	133
24	26	178	200
2	0	255	220

-

training image			
10	20	24	17
8	10	89	100
12	16	178	170
4	32	233	112

=

pixel-wise absolute value differences			
46	12	14	1
82	13	39	33
12	10	0	30
2	32	22	108

→ 456

distance "L1" ~ Manhattan

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

p: pixel
I_1: image 1

Distance Metrics

L1 (Manhattan) distance: $$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

L2 (Euclidean) distance: $$d_2(I_1, I_2) = \sqrt{\sum_p \left(I_1^p - I_2^p\right)^2}$$

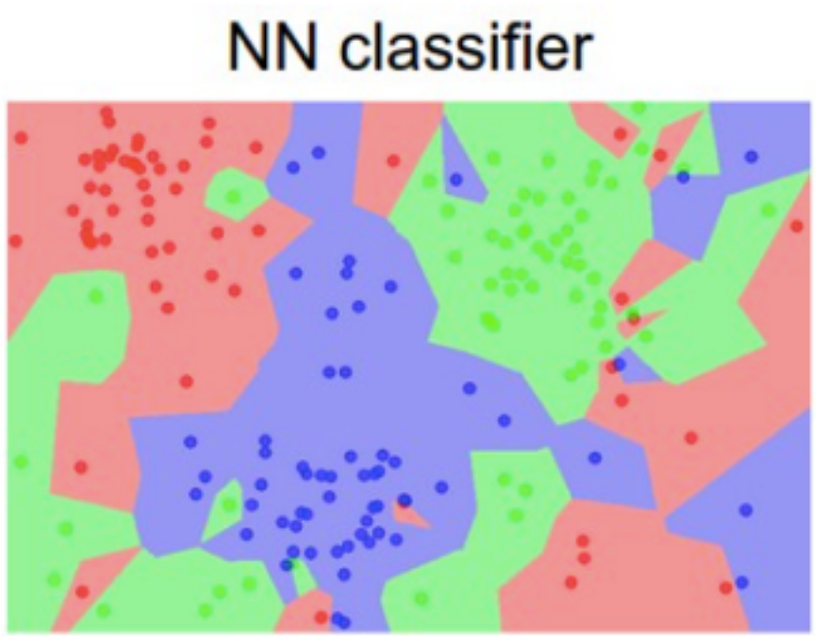
Most common.

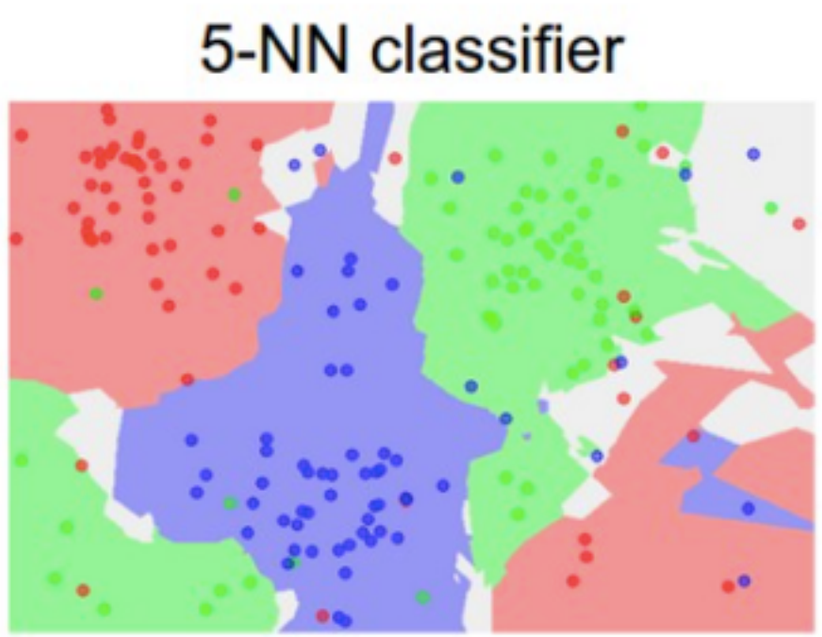
Shortest path between 2 points.

More general Lp norms:

$$||x||_p = \left(|x_1|^p + \cdots + |x_n|^p\right)^{\frac{1}{p}} \qquad p \geq 1, x \in \mathbb{R}^n$$

k-NN Classifiers

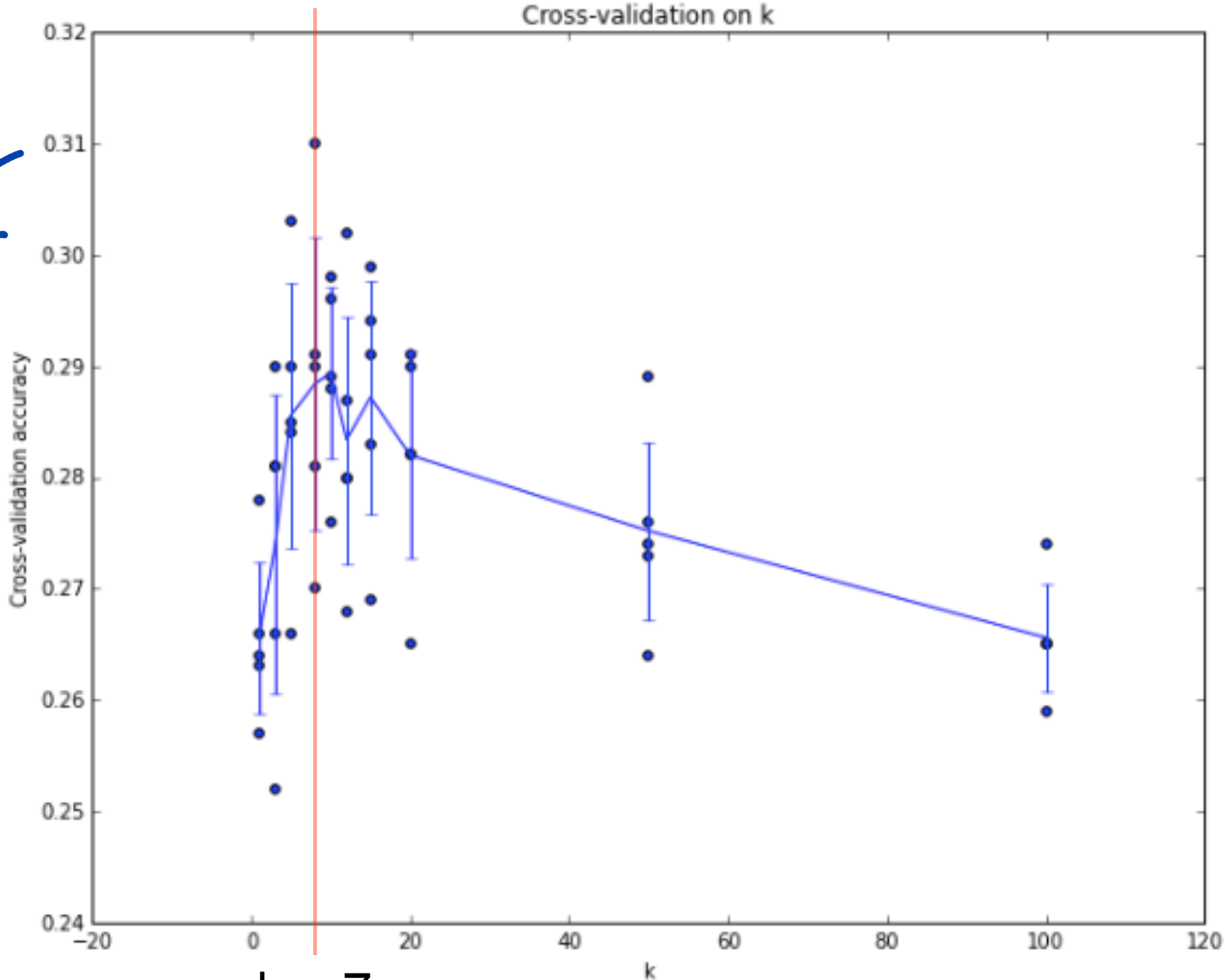




https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

5-fold Cross Validation

Correct / # in Set.



k = 7

Alex Krizhevsky, Stanford

10 Nearest Neighbors

Accuracy: L1 distance: 38% / L2 distance 35%

Looks mostly C th background b/c most pixels are in th background

↓

need a different feature, not pixels

Alex Krizhevsky, Stanford

Pixel-based L2 Distances

These images have the same L2 pixel distance

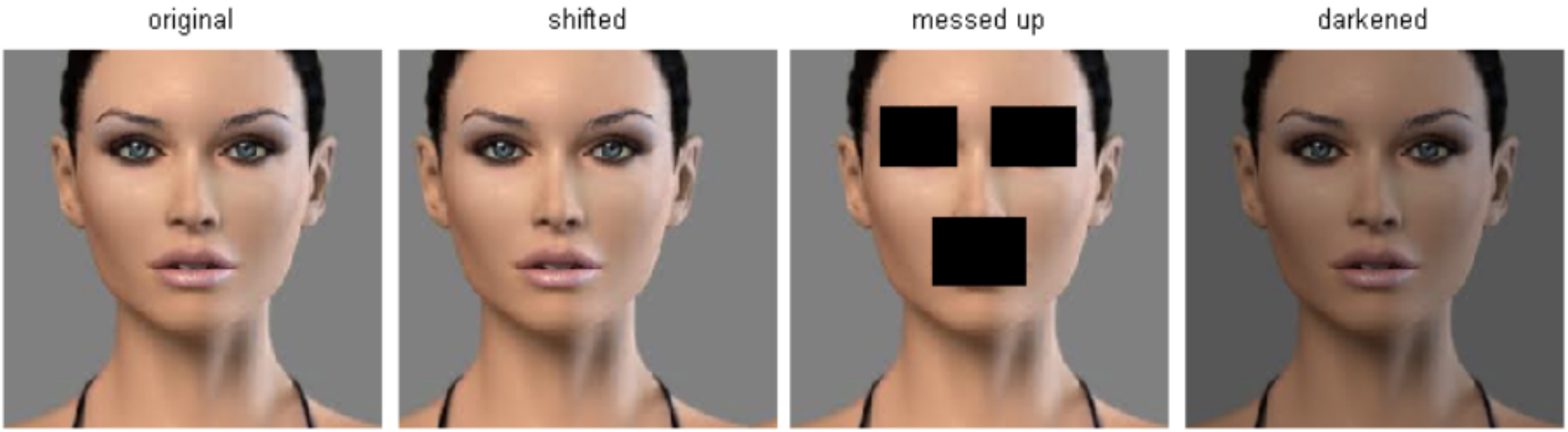
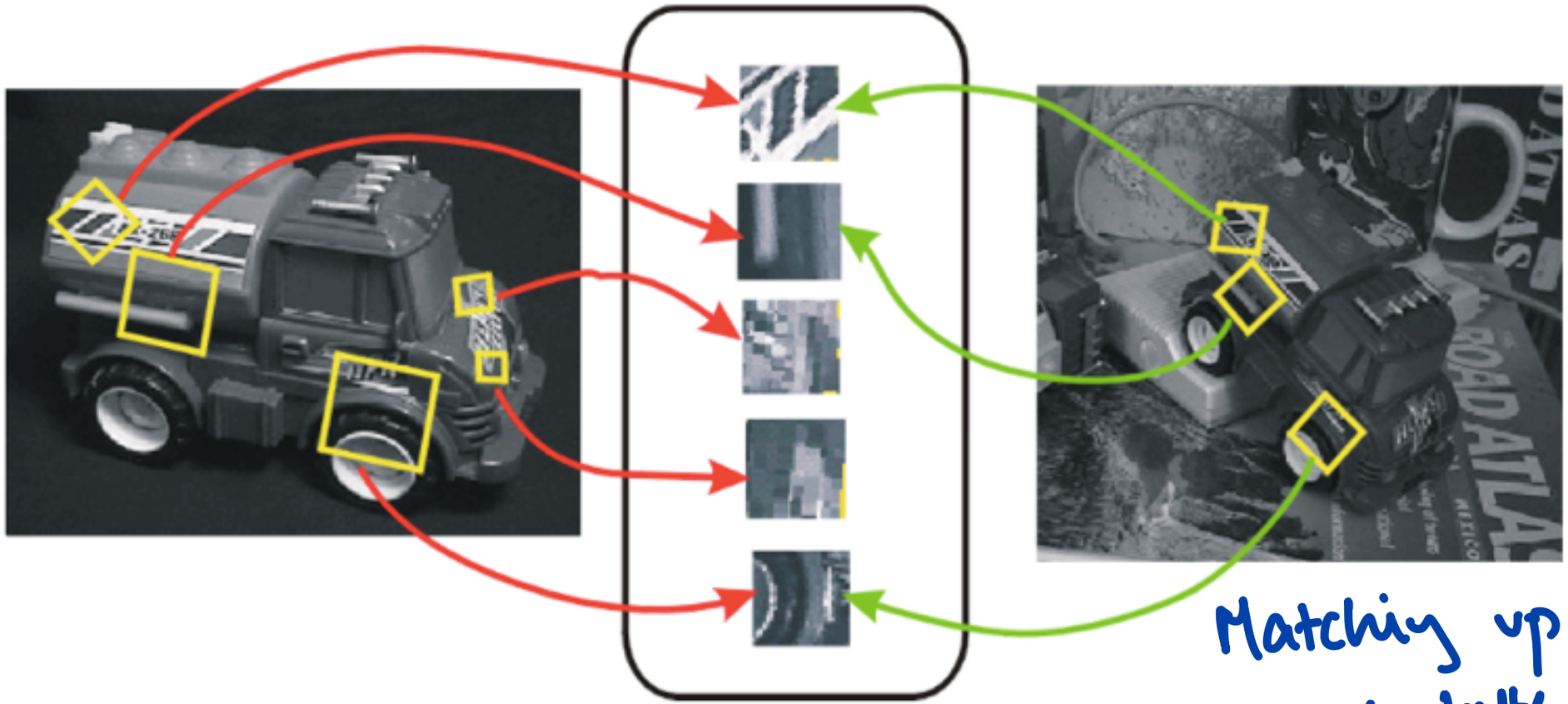
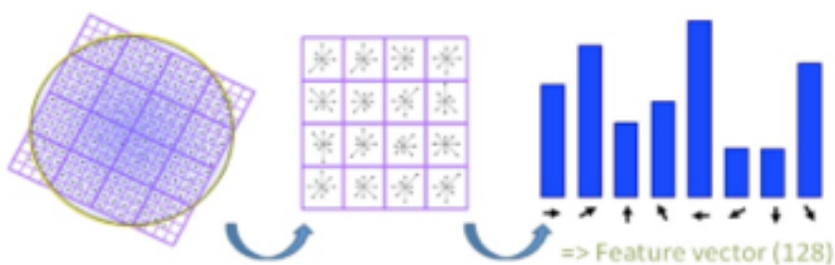


Image Features



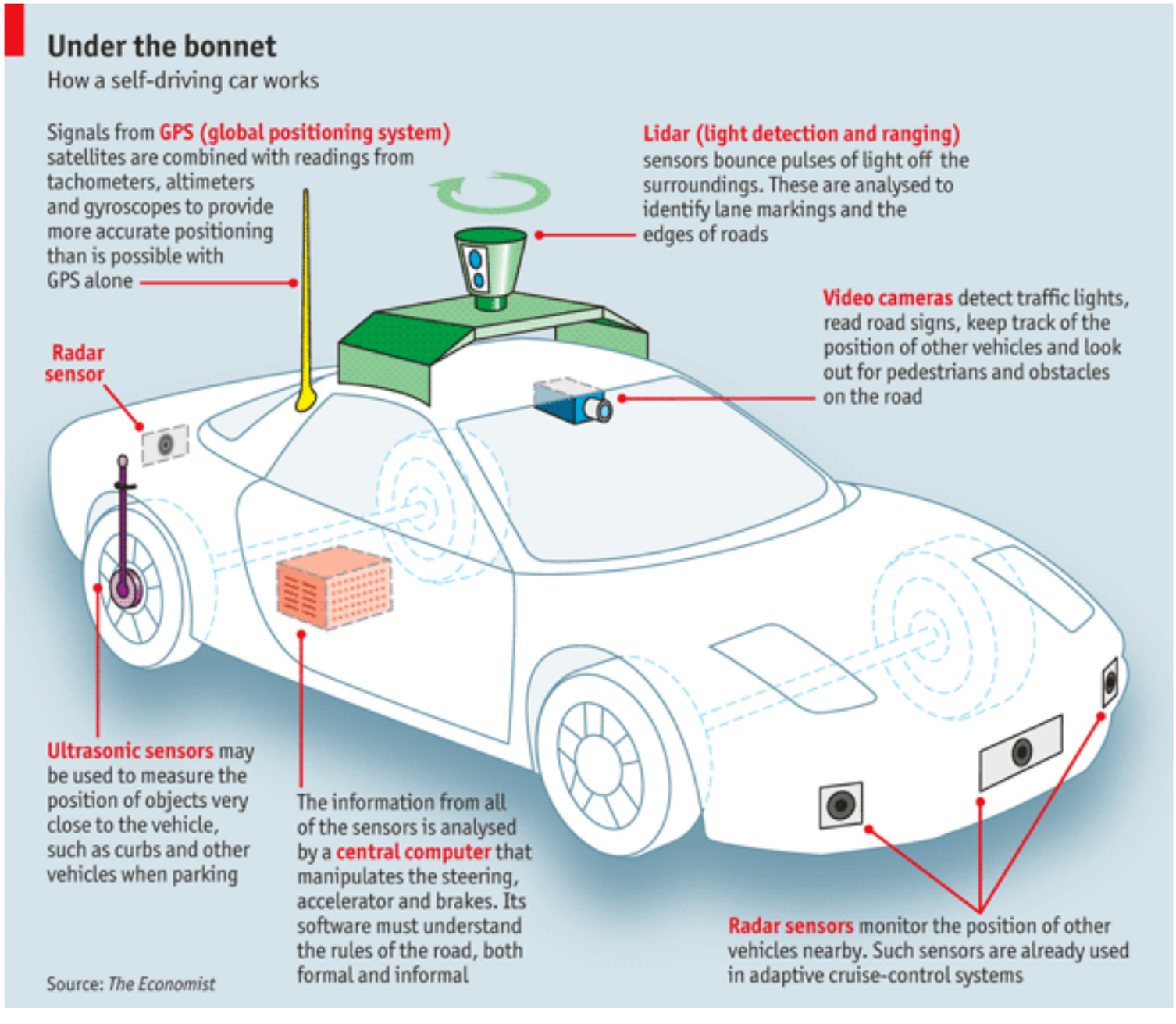


Matching up features much better than pixel-by-pixel feature

"SIFT" & Object Recognition, David Lowe, 1999

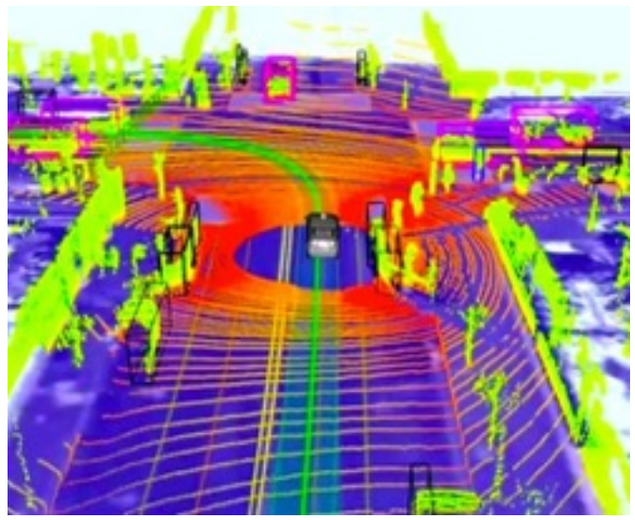
Self-Driving Cars

Diff sensor give diff feature



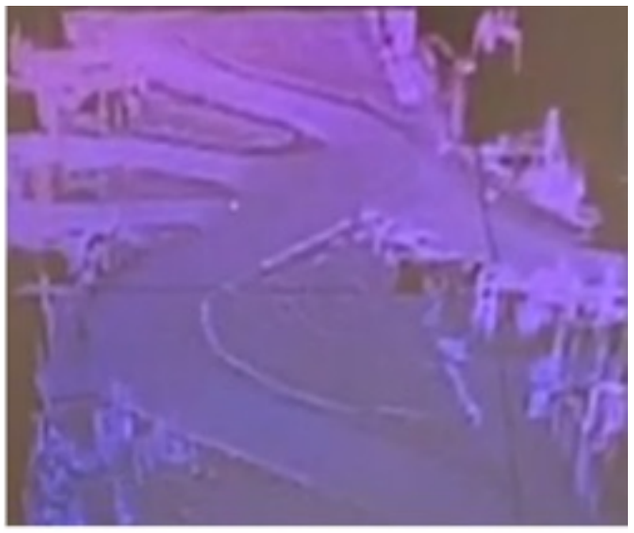
Car Features

Laser scan



Intensity model

Elevation model



Camera vision

2D stationary map

Lane model

Why don't we just use more and more features?

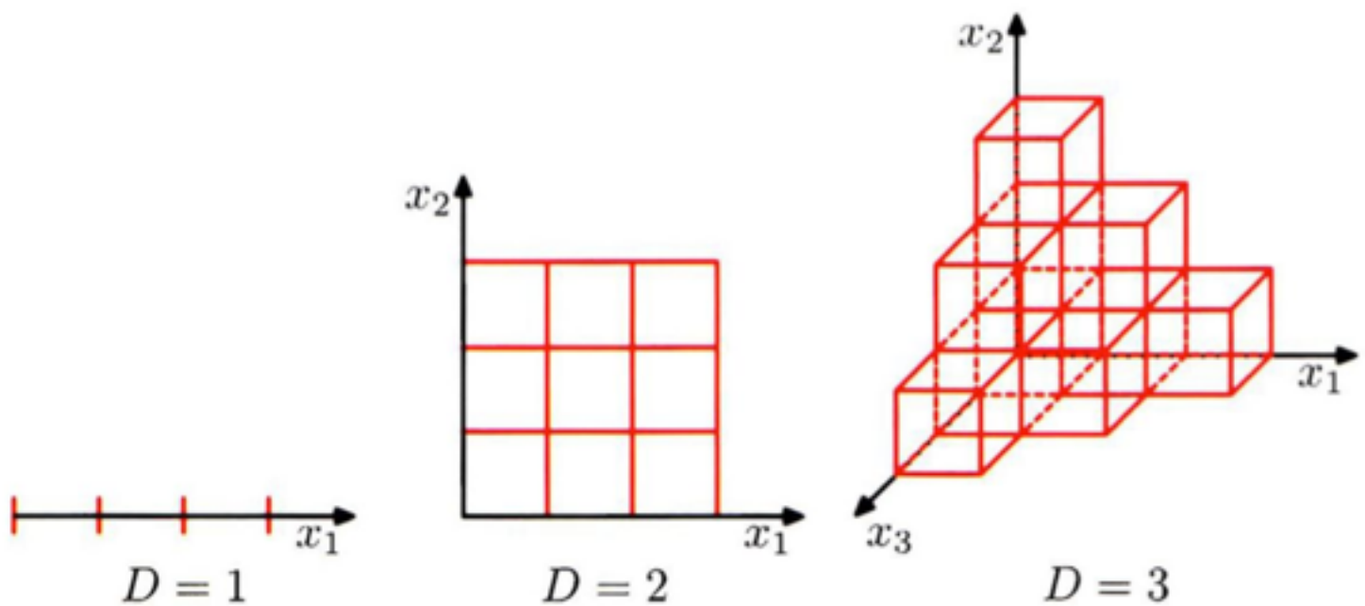
↳ More time. (1, not deep learn)

↳ Data becom sparse

Curse of Dimensionality

more features

- When dimensionality increases, the volume of the space increases so fast that the available data becomes sparse → no nearest neighbors!
- Statistically sound result requires the sample size N to grow exponentially with d



Dimensionality Reduction

- break down into lower dimensional problems.
- want to preserve ↑ distance.

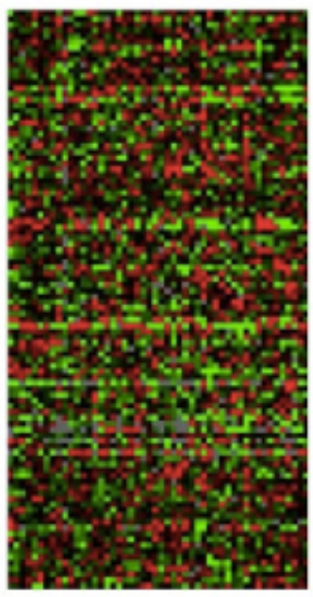
High-dimensional Data

face images

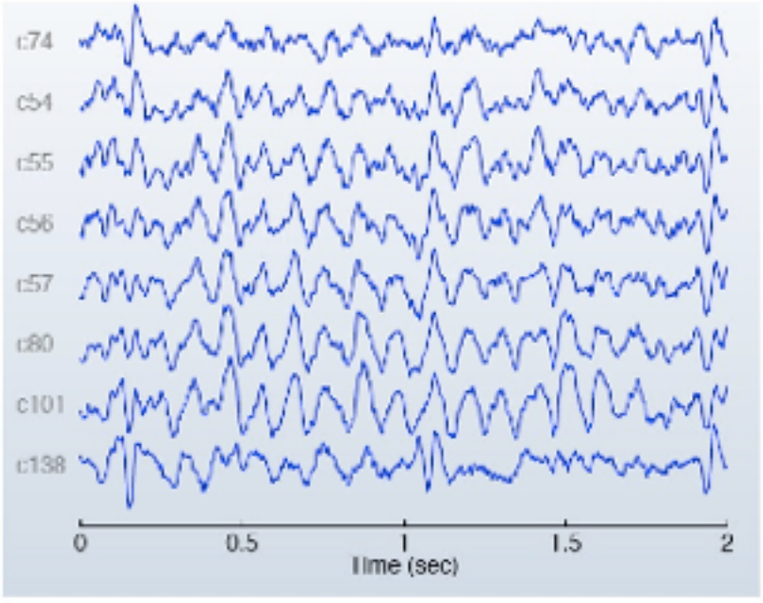
Zambian President Levy Mwanawasa has won a second term in office in an election his challenger Michael Sata accused him of rigging, official results showed on Monday.

According to media reports, a pair of hackers said on Saturday that the Firefox Web browser, commonly perceived as the safer and more customizable alternative to market leader Internet Explorer, is critically flawed. A presentation on the flaw was shown during the ToorCon hacker conference in San Diego.

documents



gene expression data



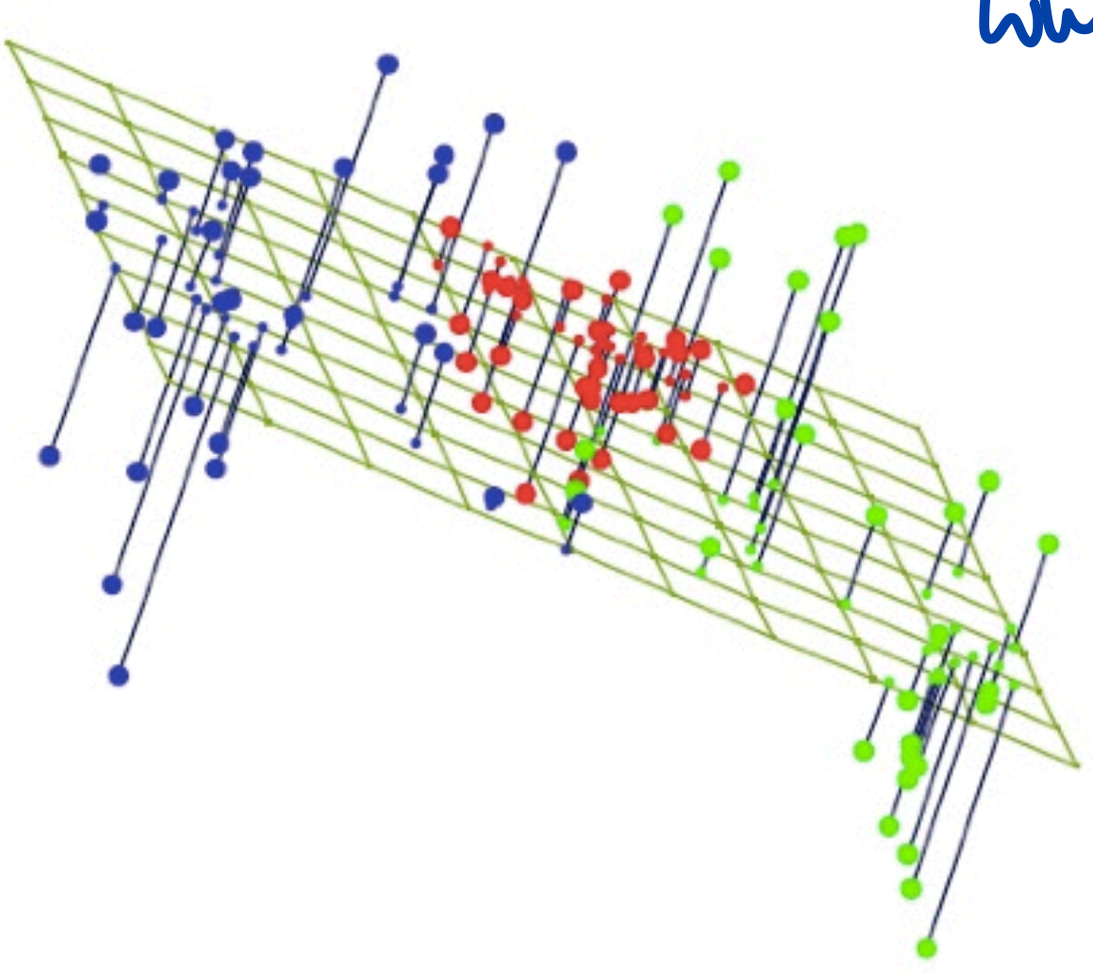
MEG readings

Based on slide from P. Liang

Basic Idea

Project the high-dimensional data onto a lower-dimensional subspace that best "fits" the data

while preserving distance.

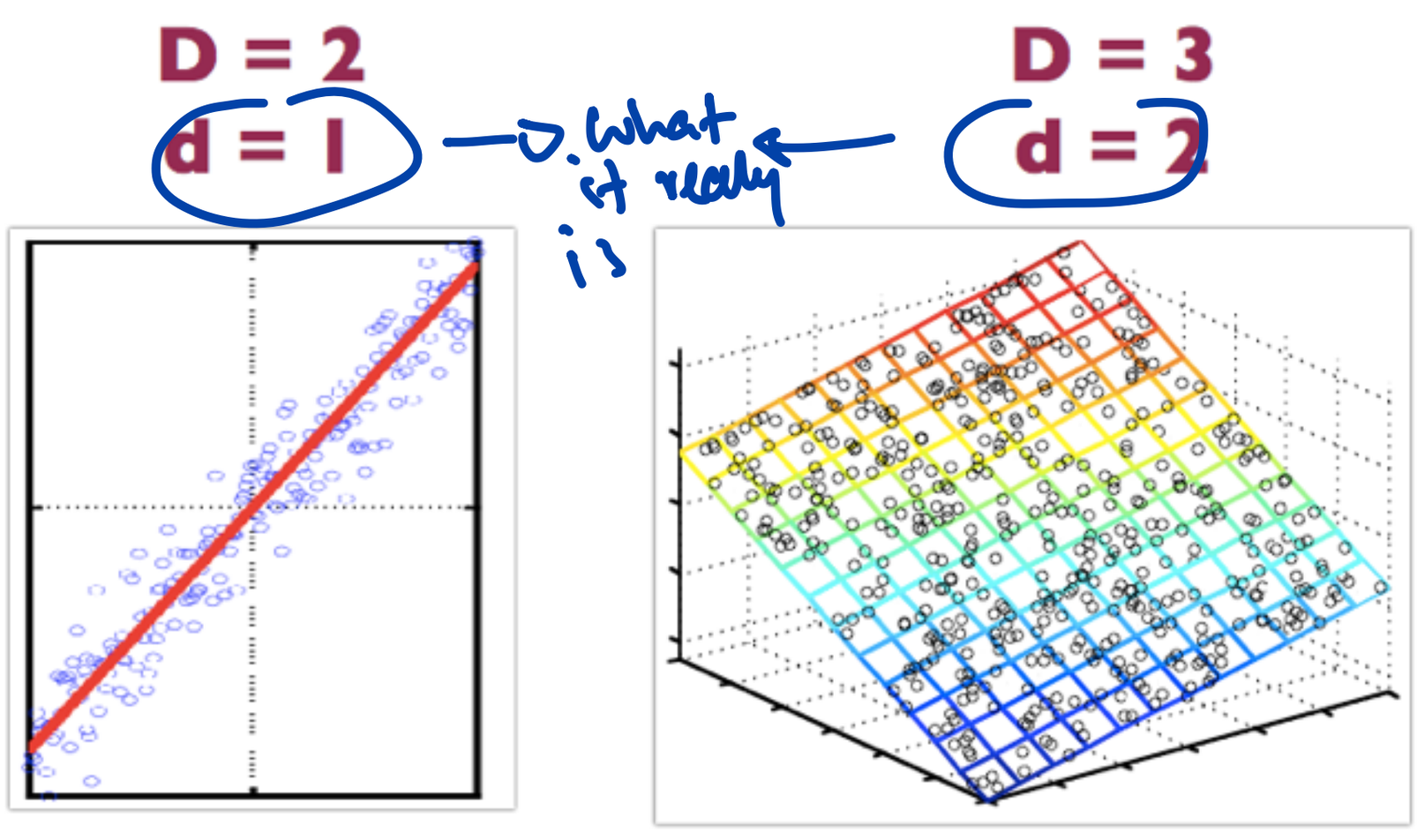


Technique are linear.

Based on slide from P. Liang

Linear Methods

- Does the data lie mostly in a hyperplane?
- If so, what is its *intrinsic* dimensionality d?

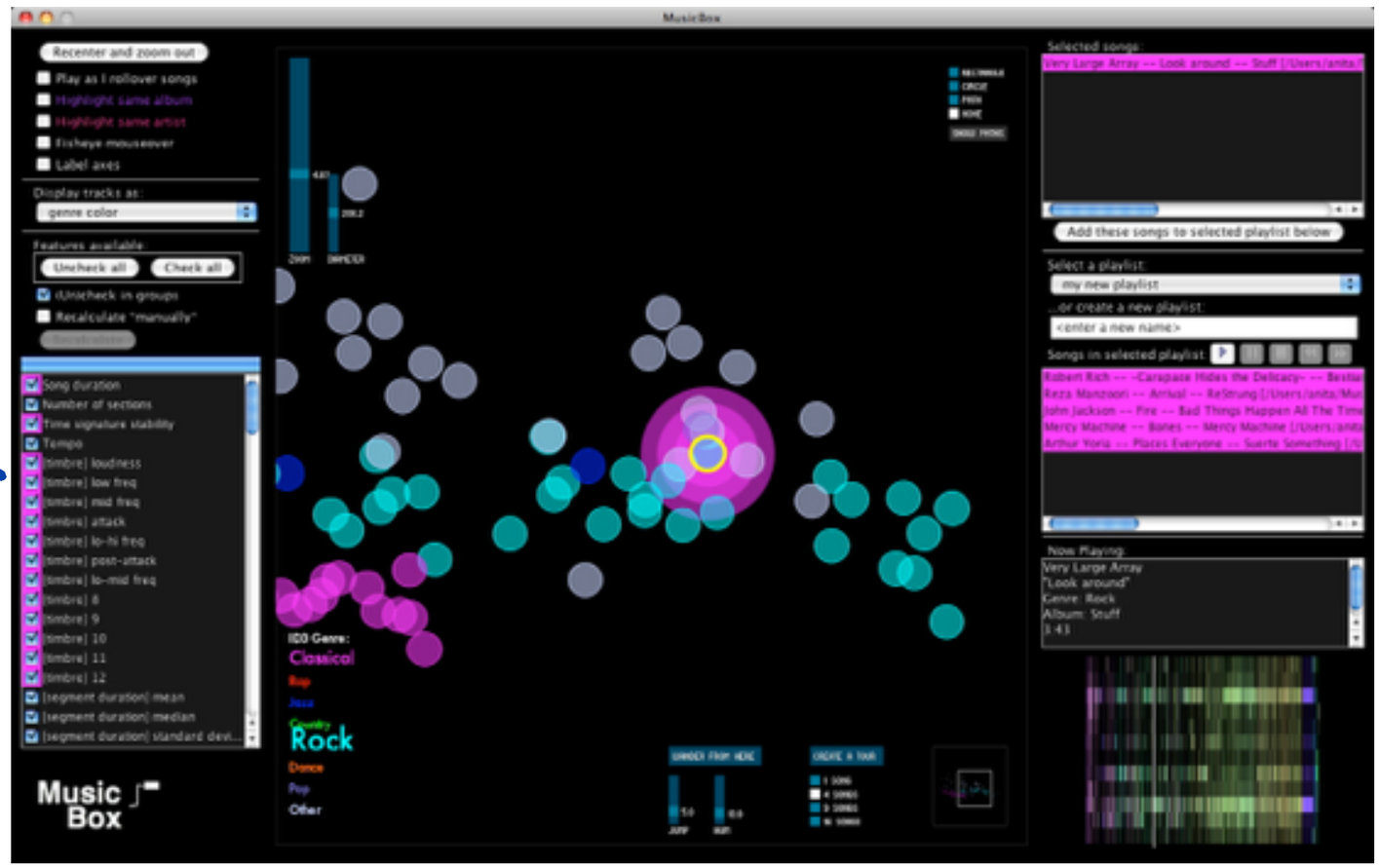


Based on slide from F. Sha

PCA

Uses

- Dimensionality reduction for supervised learning
- Compression
- Visualization



3000 features

x, y are the top two features.

labels.

MusicBox, Anita Lillie, MIT

MusicBox

Recenter and zoom out

- [ ] Play as I rollover songs
- [ ] Highlight same album
- [ ] Highlight same artist
- [ ] Fisheye mouseover
- [ ] Label axes

Display tracks as:

genre color

Features available:

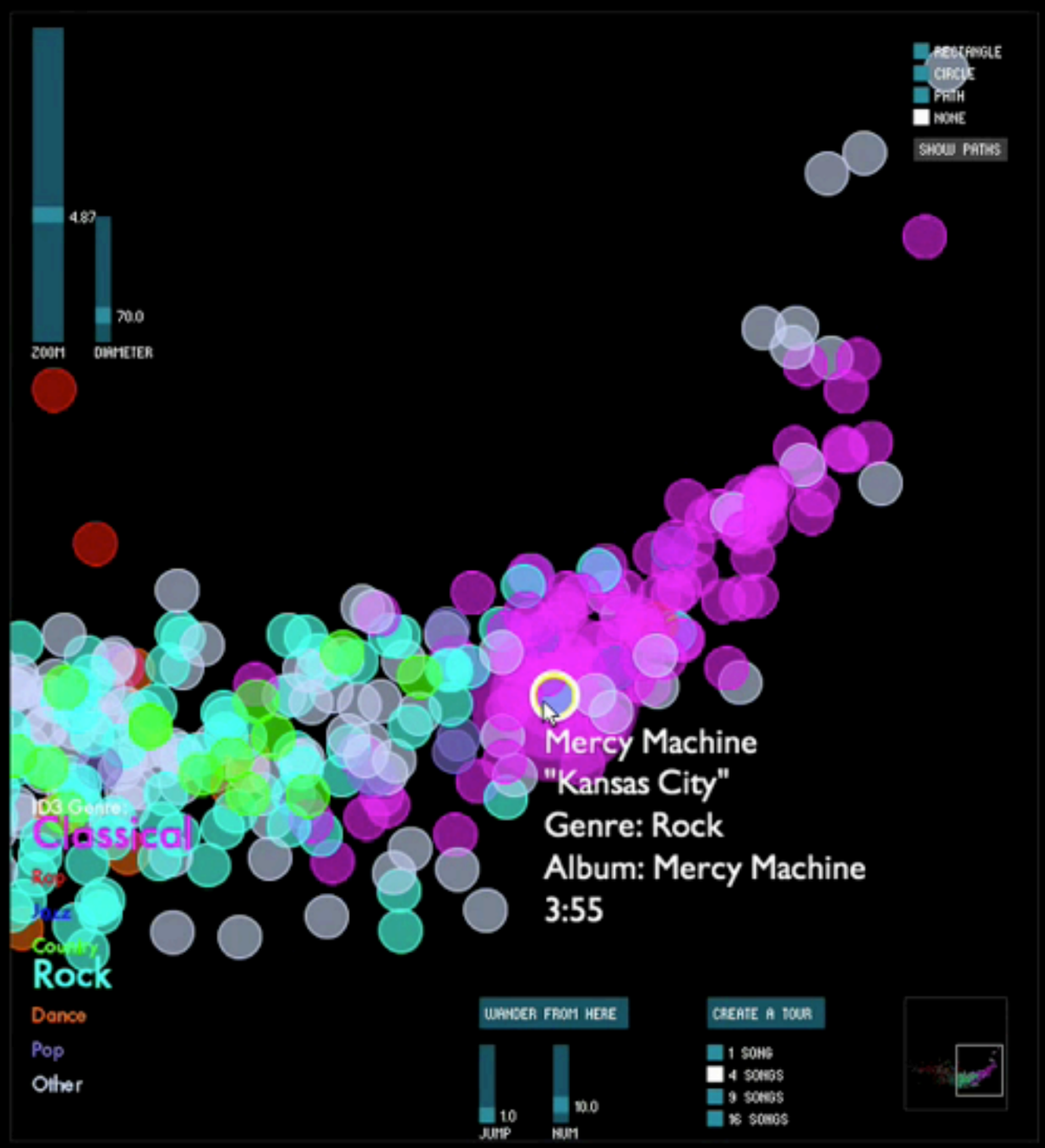
Uncheck all | Check all

- [x] (Un)check in groups
- [ ] Recalculate "manually"

Recalculate

- [x] Song duration
- [x] Number of sections
- [x] Time signature stability
- [x] Tempo
- [x] [timbre] loudness
- [x] [timbre] low freq
- [x] [timbre] mid freq
- [x] [timbre] attack
- [x] [timbre] lo-hi freq
- [x] [timbre] post-attack
- [x] [timbre] lo-mid freq
- [x] [timbre] 8
- [x] [timbre] 9
- [x] [timbre] 10

Music Box



Selected songs:

Mercy Machine -- Kansas City -- Mercy Machine [/Users

Add these songs to selected playlist below

Select a playlist:

empty

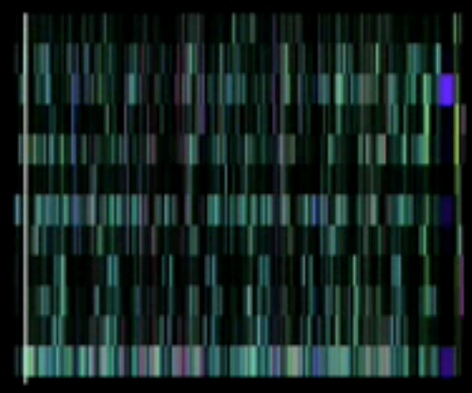
...or create a new playlist:

<enter a new name>

Songs in selected playlist:

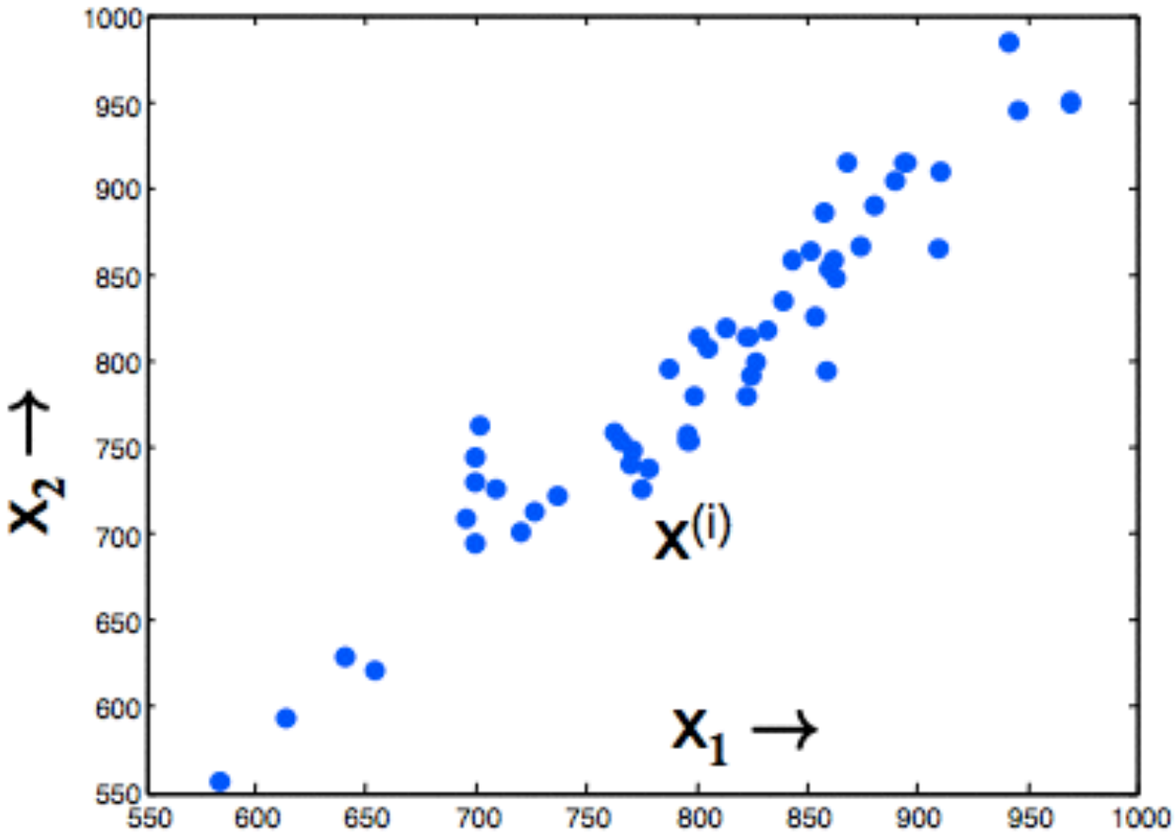
Now Playing:

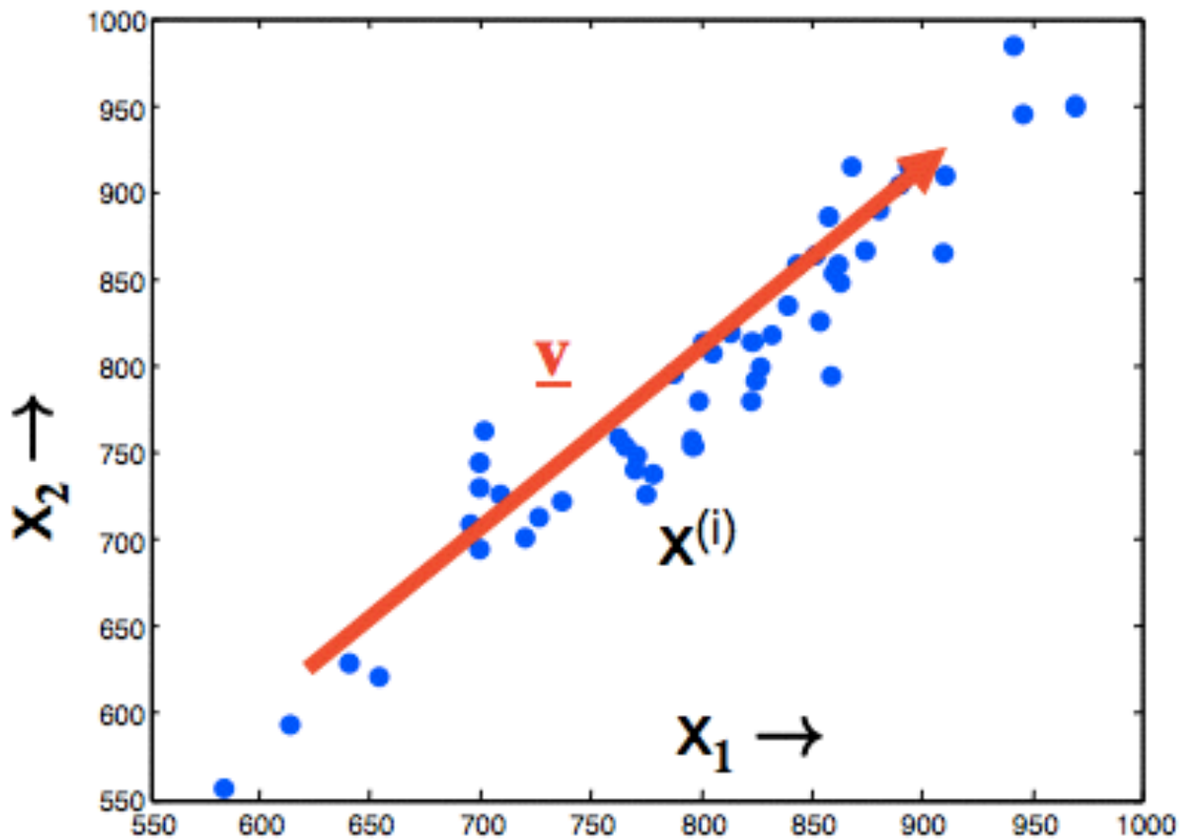
Mercy Machine
"Kansas City"
Genre: Rock
Album: Mercy Machine
3:55



Principal Components Analysis (PCA)

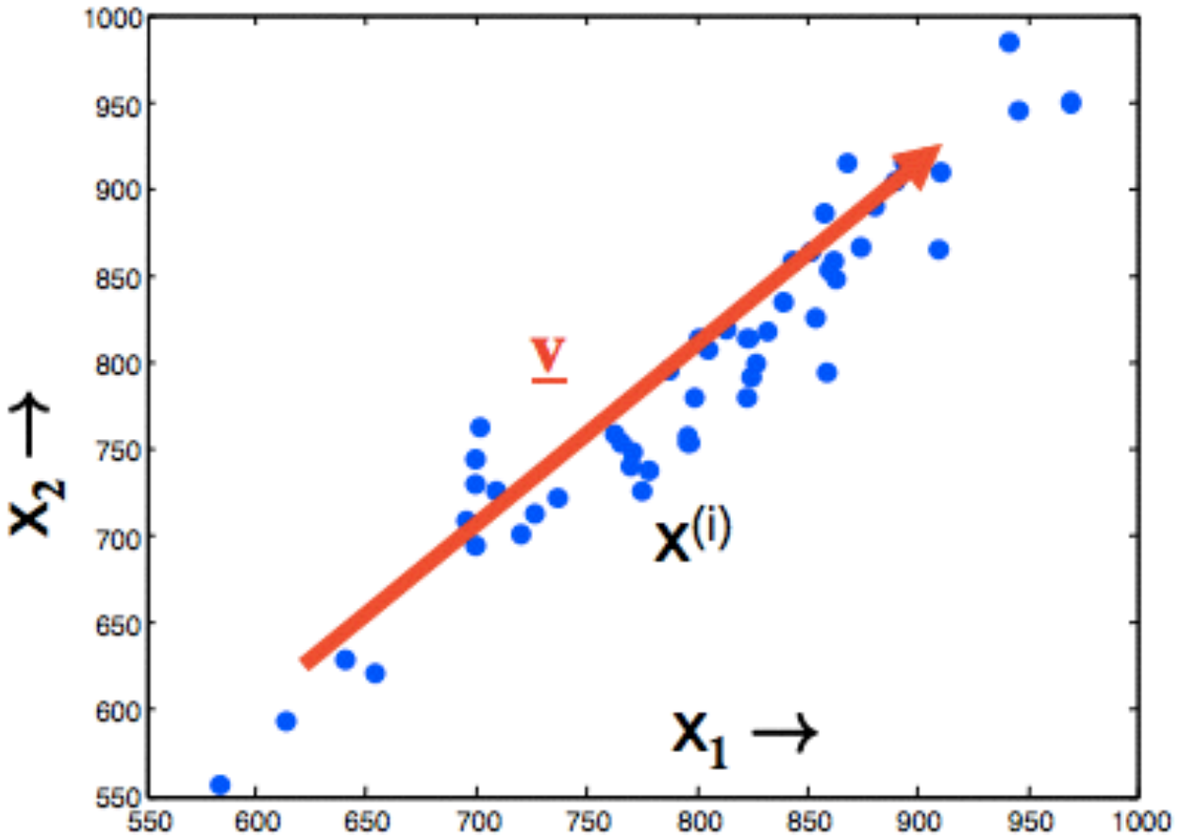
Example





Based on slide from A. Ihler

Example



v: chosen to minimize orthogonal distances

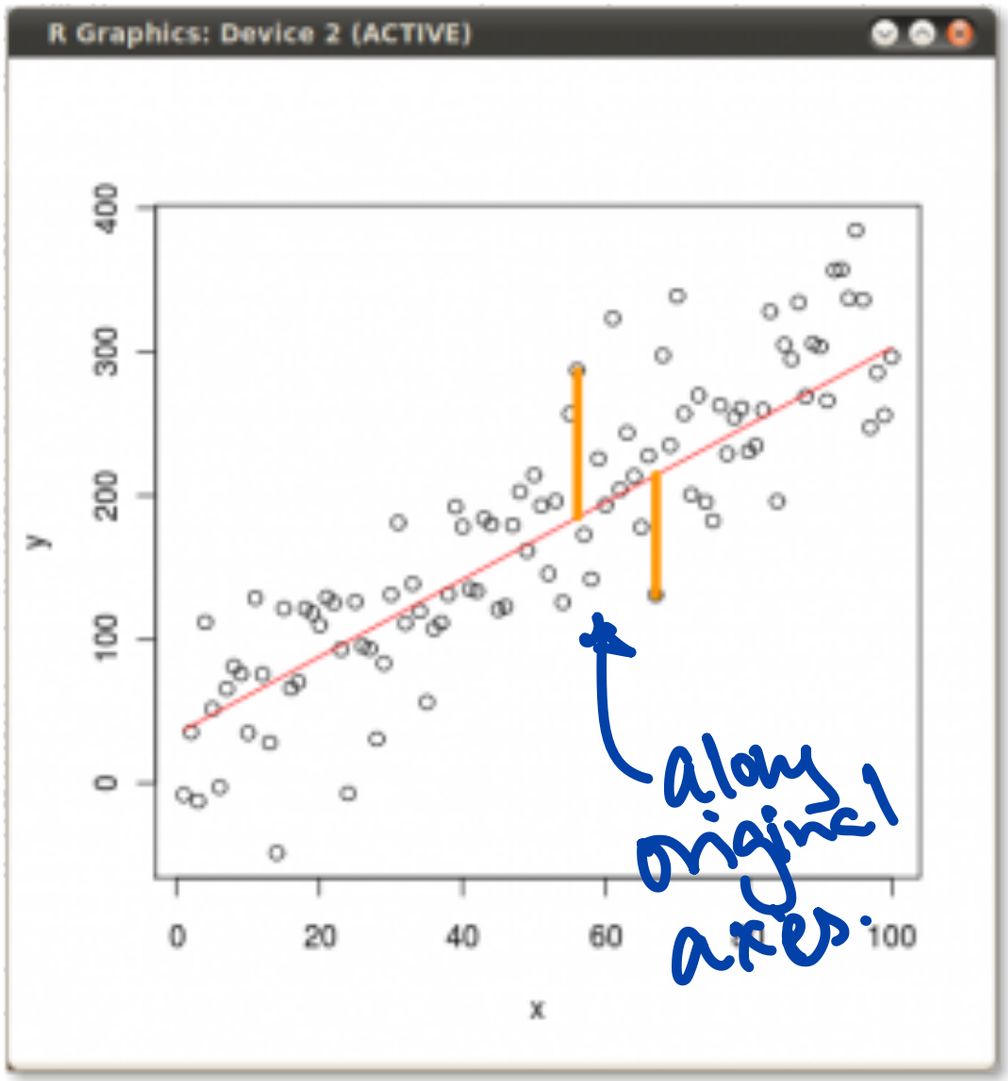
Equivalent: v is the direction of maximum variance (max. the spread along v)

minimize ⊥ distance

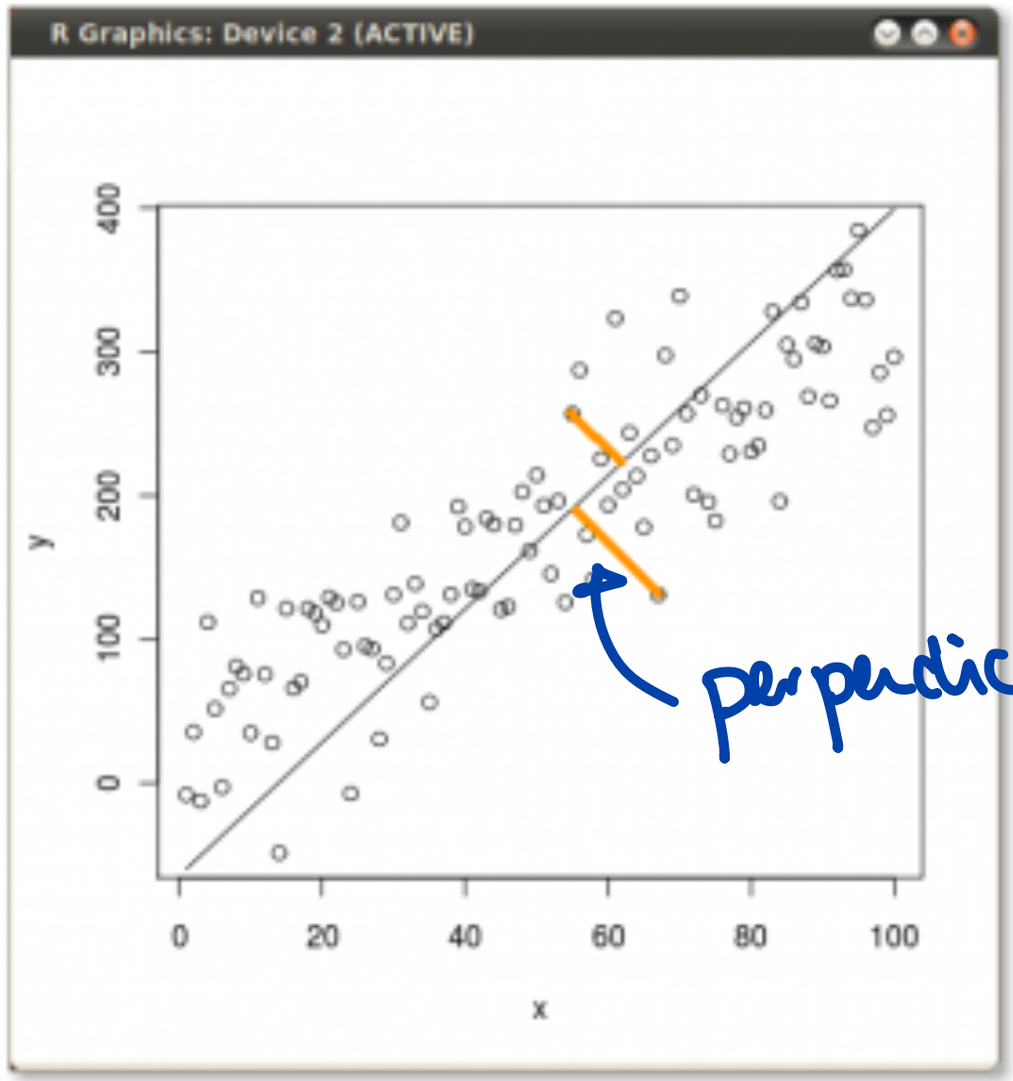
Based on slide from A. Ihler

Linear Regression vs. PCA

Linear Regression



PCA



http://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

PCA Algorithm

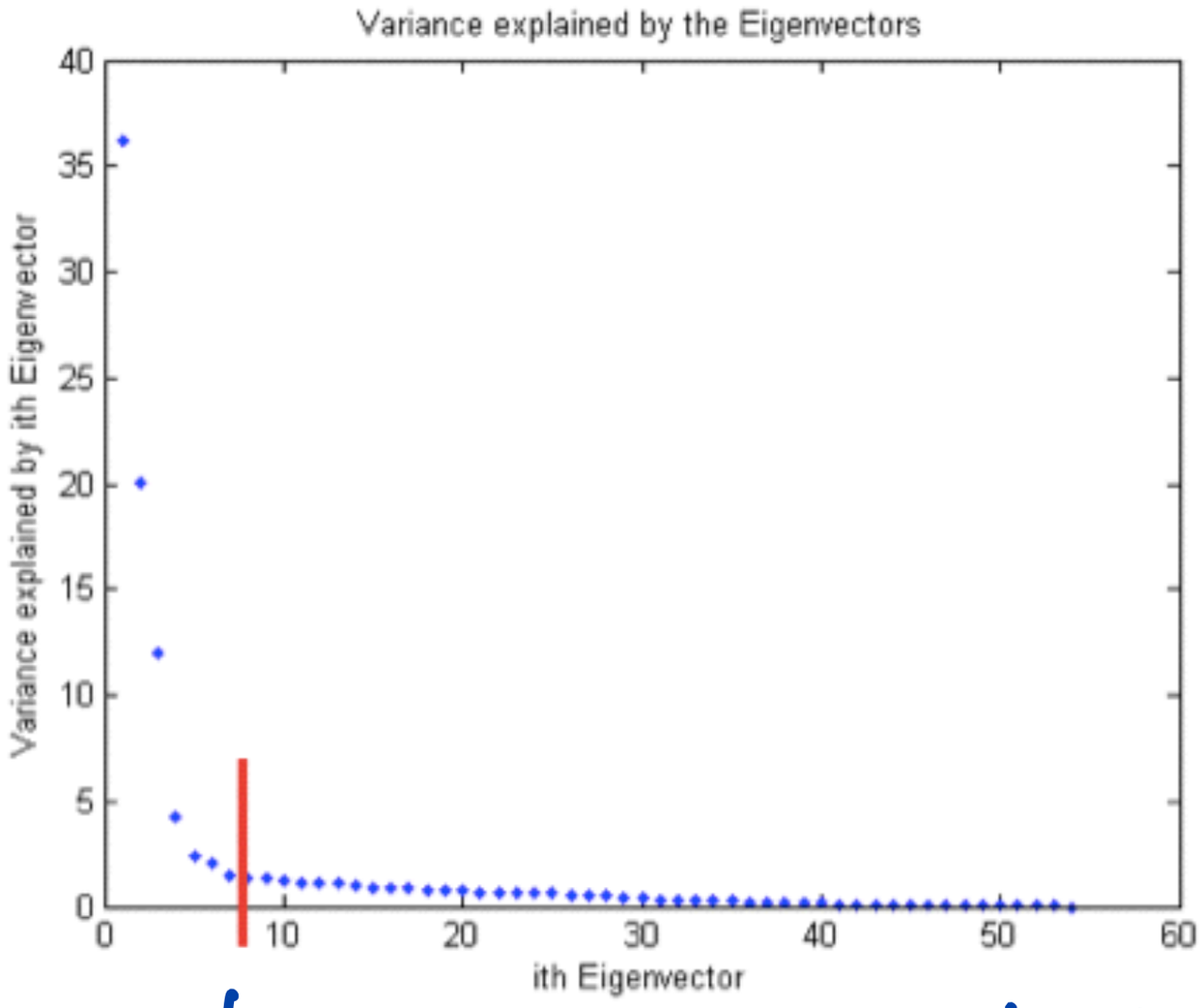
- Subtract mean from data (center X)
- (Typically) scale each dimension by its variance
  - Helps to pay less attention to magnitude of dimensions
- Compute covariance matrix S $\mathbf{S} = \frac{1}{N}\mathbf{X}^{\mathsf{T}}\mathbf{X}$
- Compute k largest eigenvectors of S
- These eigenvectors are the k principal components

Based on slide from A. Ihler

How do you pick the intrinsic vectors?

How many PC vectors?

Enough PC vectors to cover 80-90% of the variance

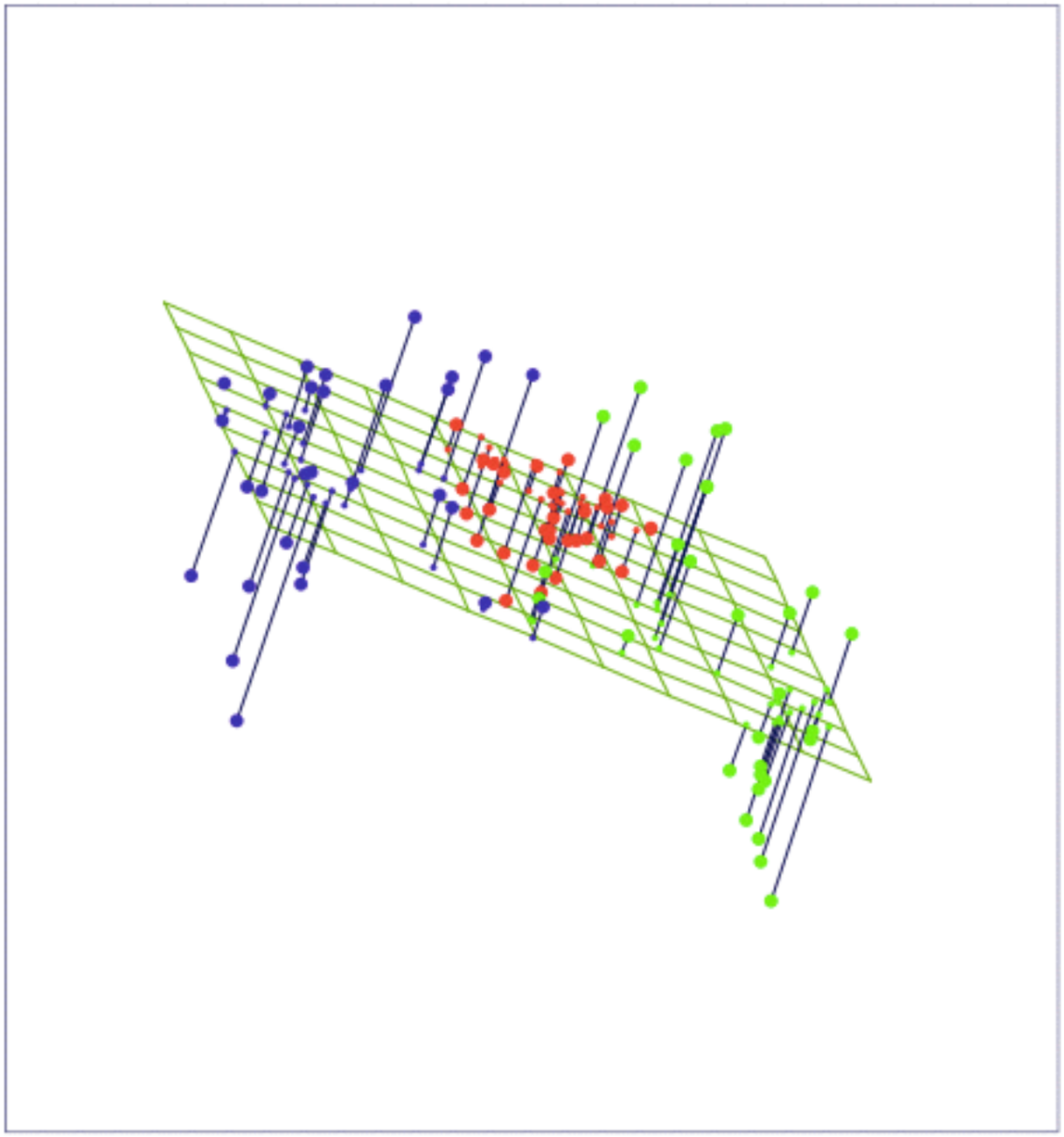


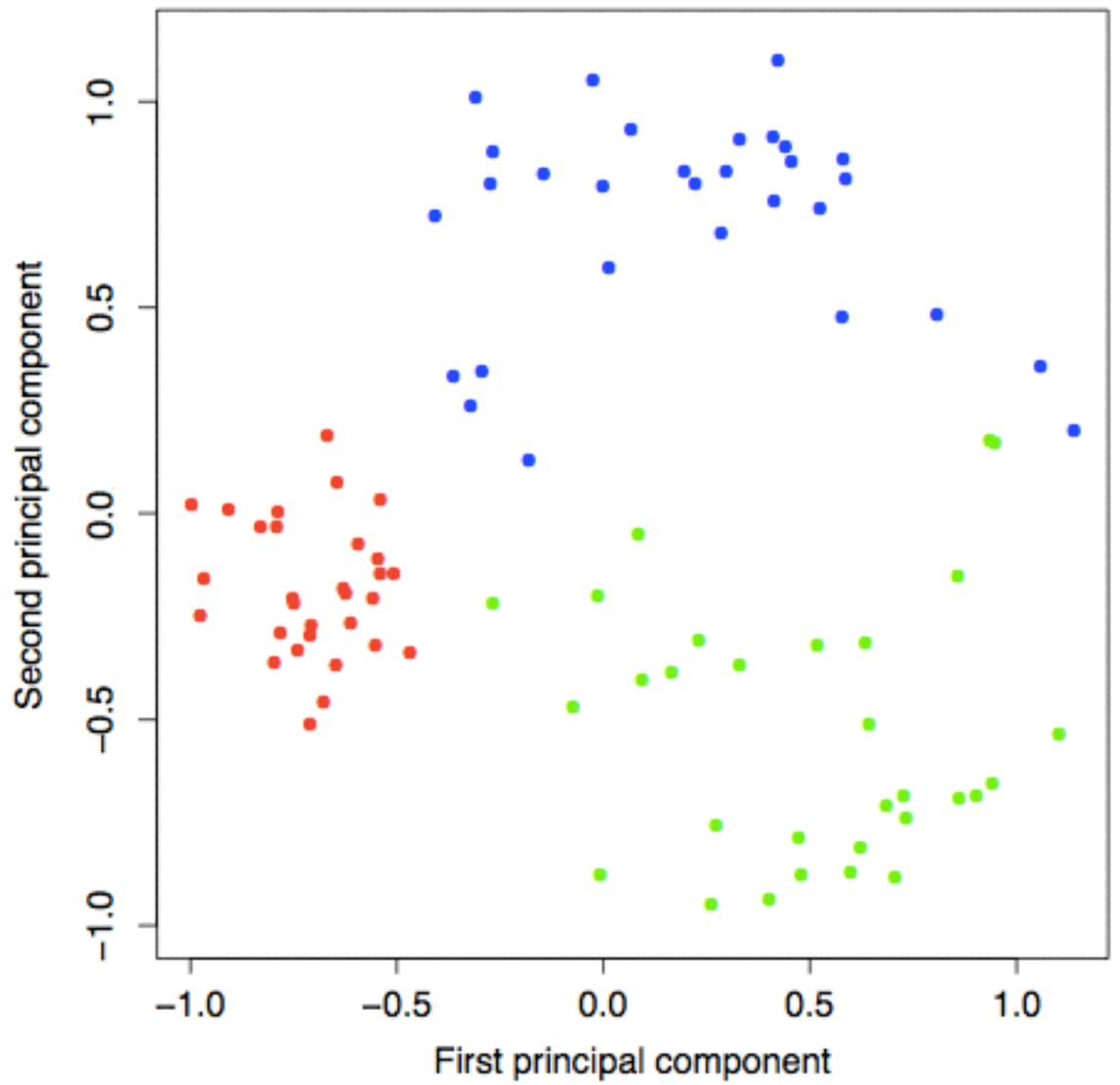
Screeplot

Look where curve starts to diminish.

Based on slide from J. Leskovec

Dimensionality Reduction





Hastie et al.,"The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer (2009)

PCA for Handwritten Digits

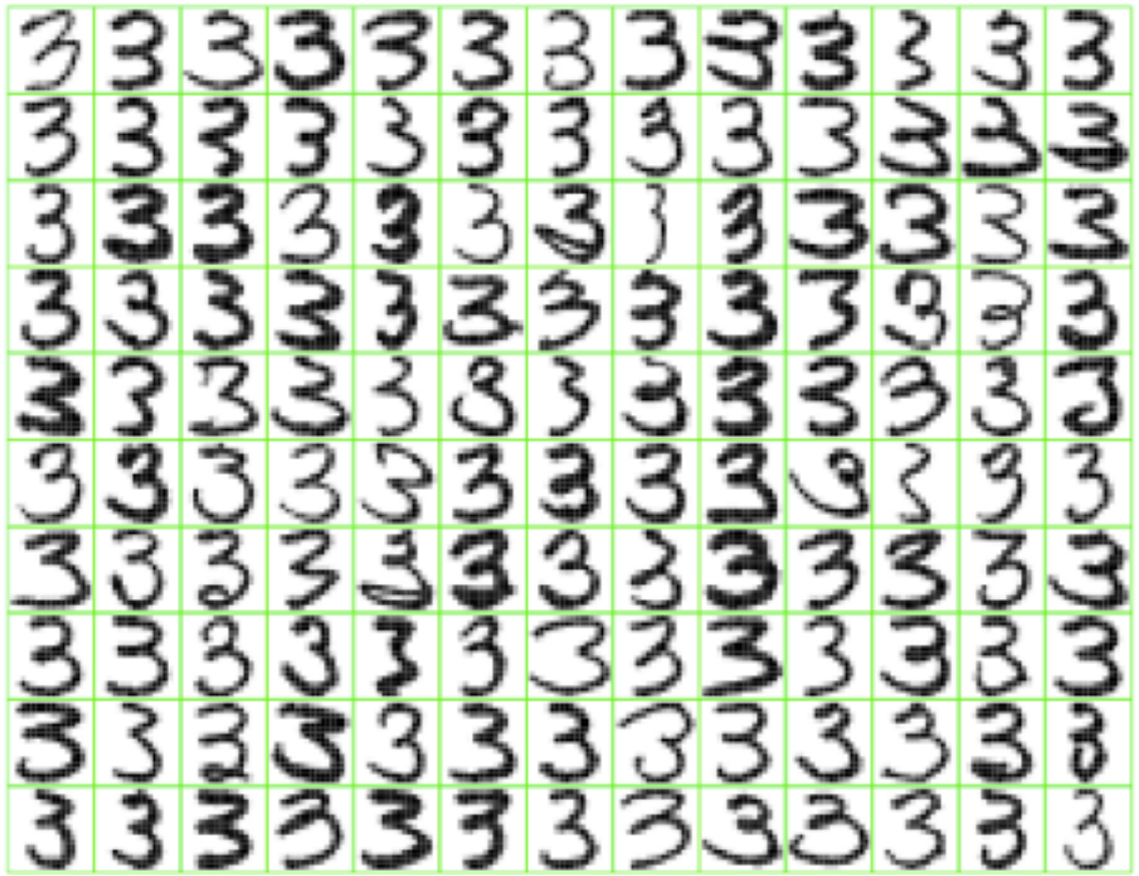


FIGURE 14.22. *A sample of* 130 *handwritten 3's shows a variety of writing styles.*

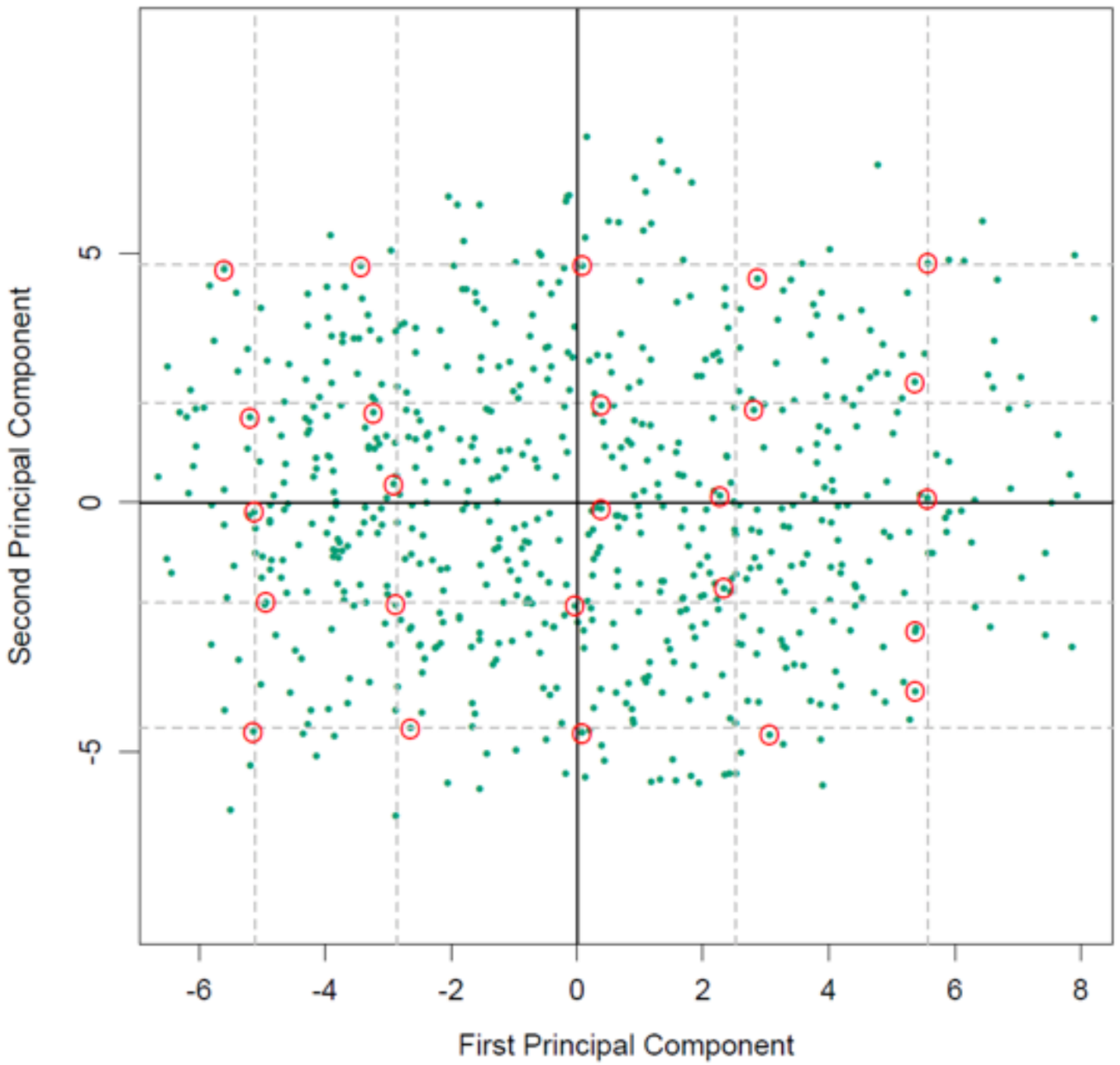
$$\hat{f}(\lambda) = \bar{x} + \lambda_1 v_1 + \lambda_2 v_2$$

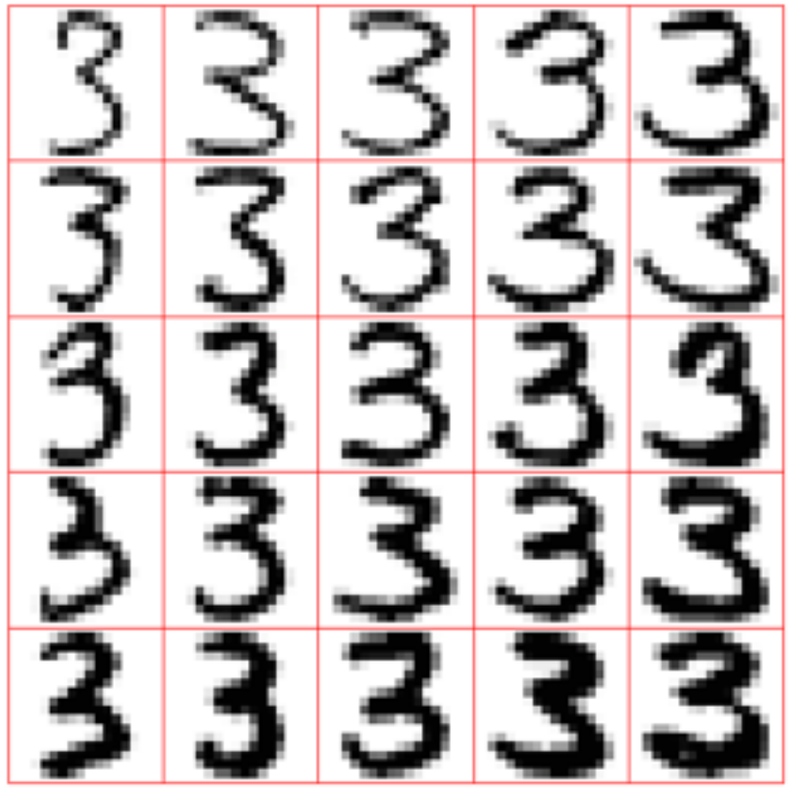
$$= \bar{x} + \lambda_1 \cdot v_1 + \lambda_2 \cdot v_2 .$$

dimensional reduction.

Hastie et al.,"The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer (2009)

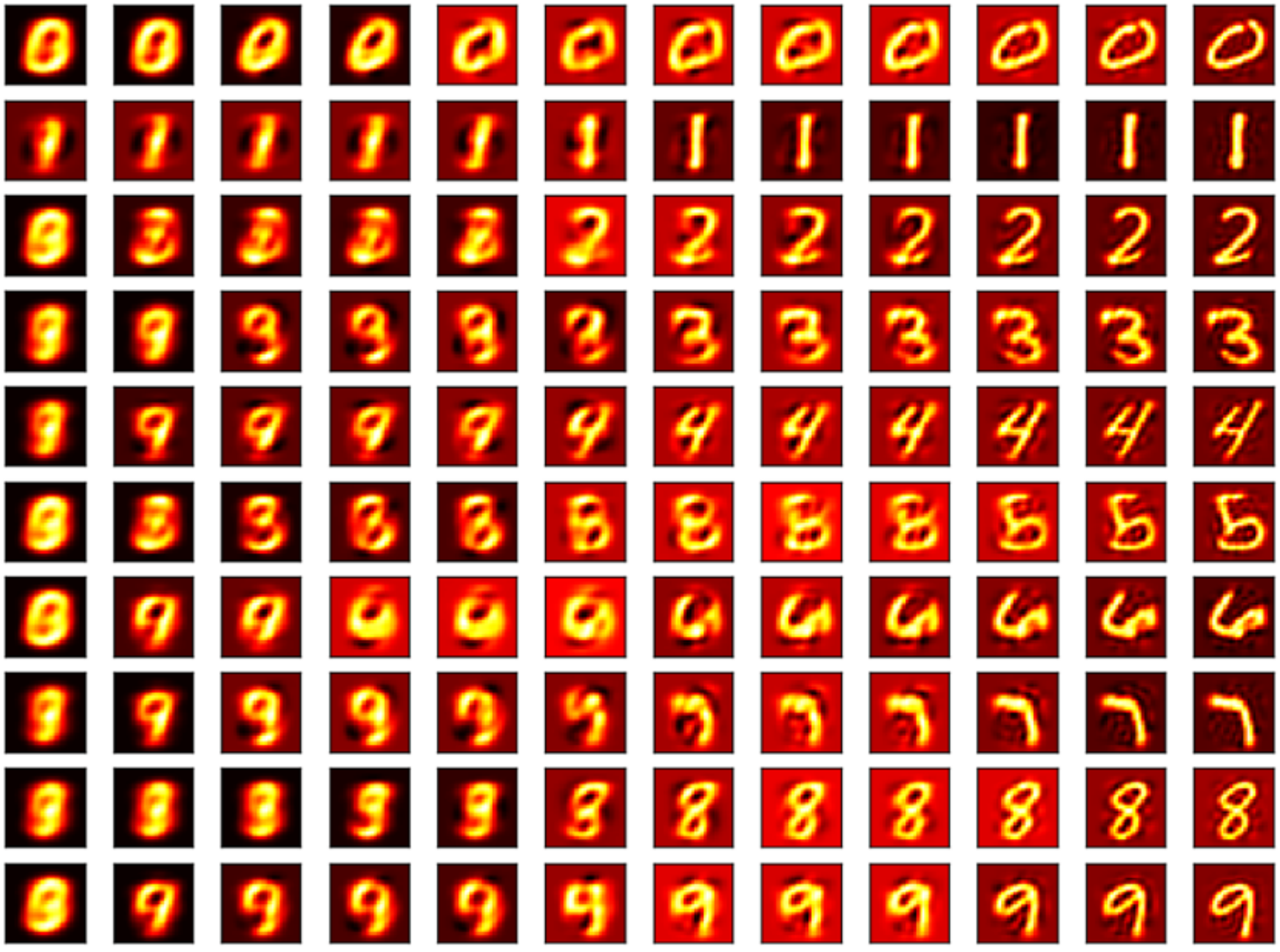
PCA for Handwritten Digits





Hastie et al.,"The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer (2009)

Image reconstruction after reduction with PCA



1 2 3 4 5 10 20 25 30 50 70 100

Number of components

PCA for Face Images

Based on slide from T. Yang

PCA for Face Images

- 64x64 images of faces = 4096 dimensional data

Based on slide from A. Ihler

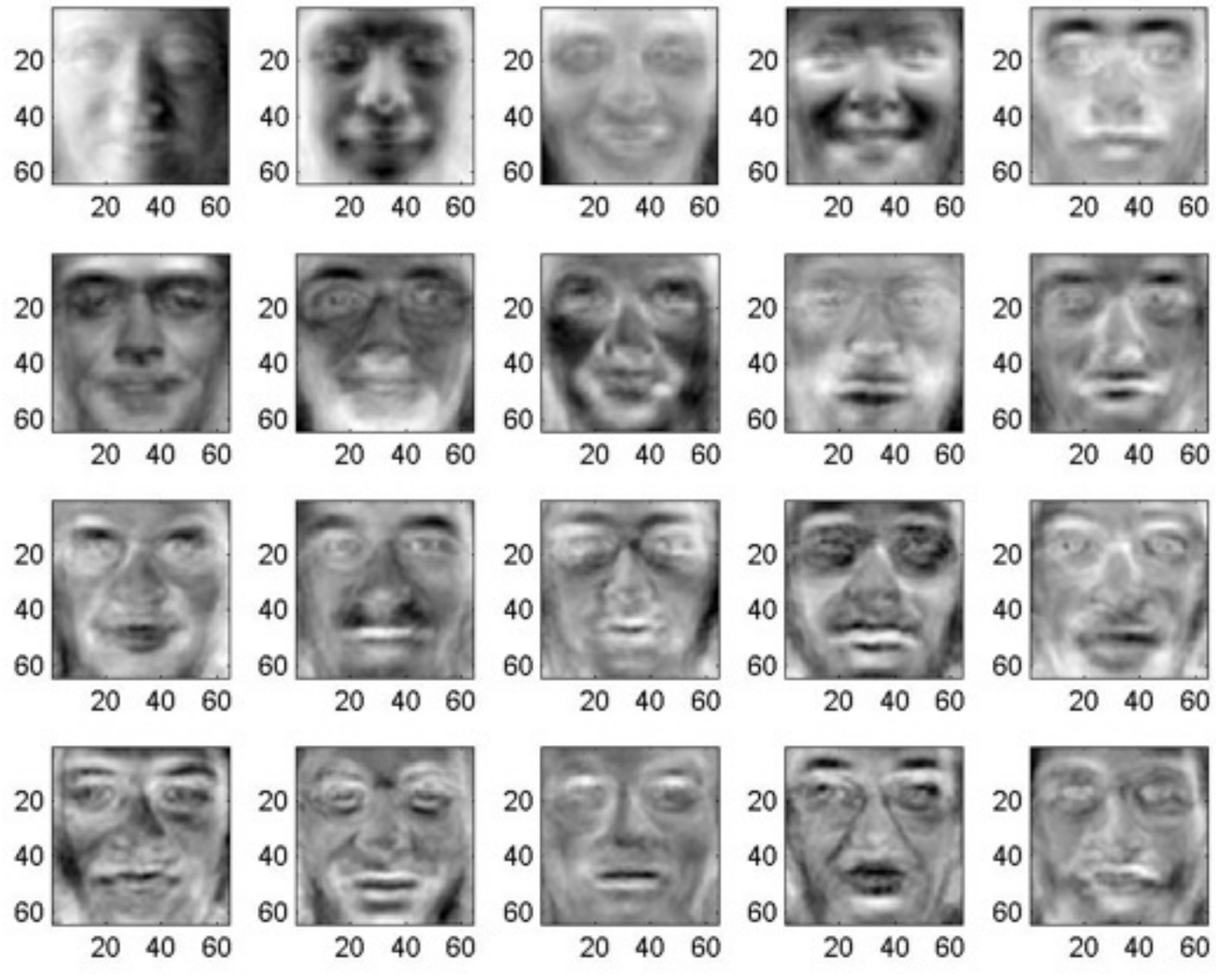
“Eigenfaces”

First 4-5 explain most of the variance.

We can reconstruct each face as a linear combination of “basis” PC vectors, or Eigenfaces [M. Turk and A. Pentland (1991)]

Average Face

+

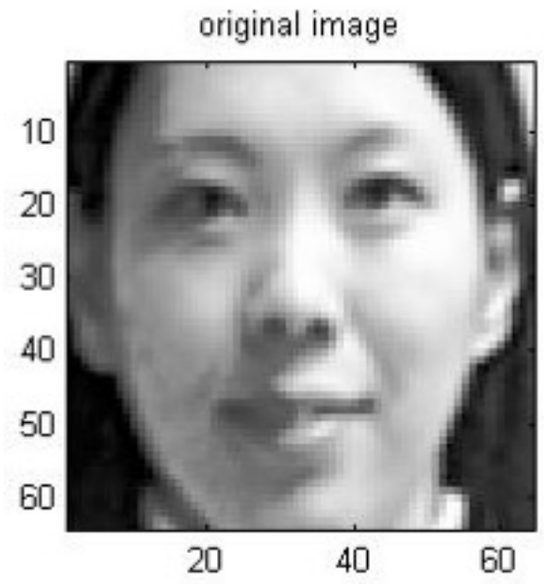


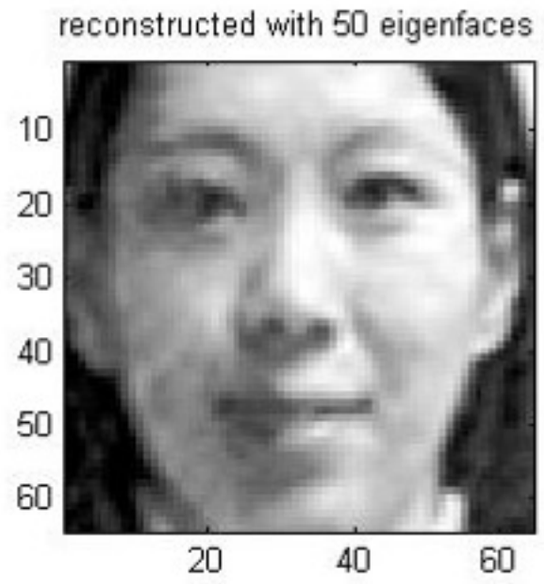
Eigenfaces

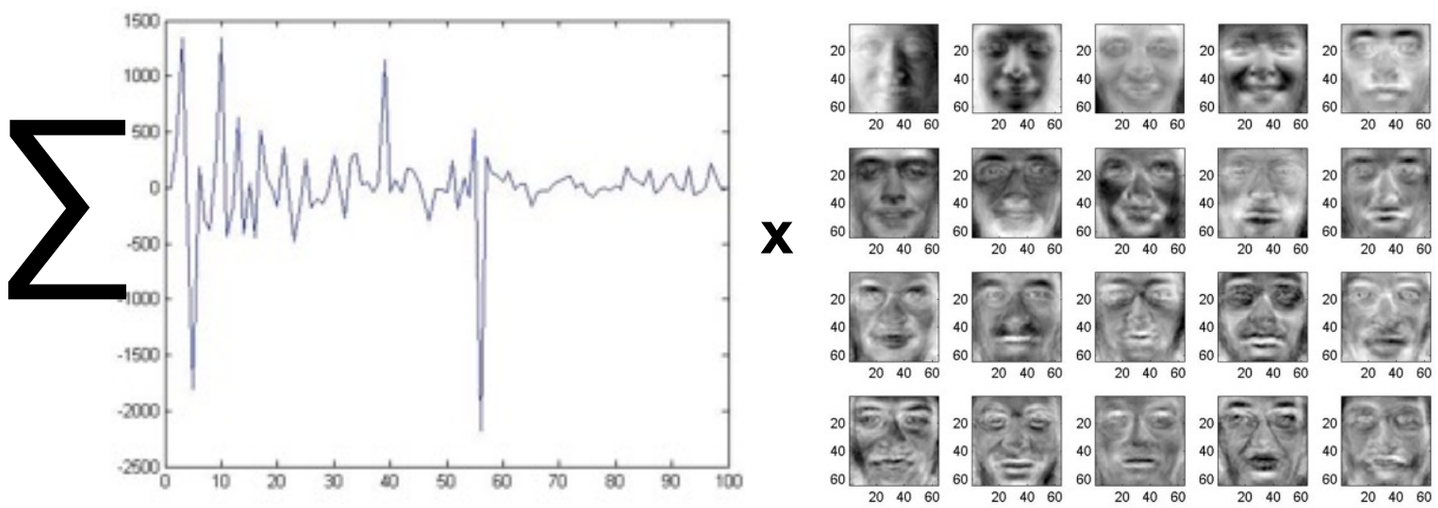
Based on slide from T. Yang

Reconstruction

90% variance is captured by the first 50 eigenvectors







Based on slide from T. Yang

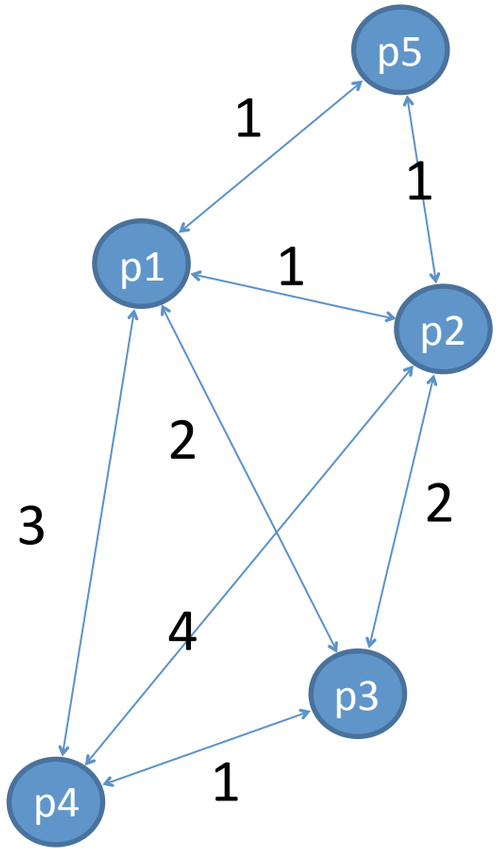
Multidimensional Scaling (MDS)

↳ Related to PCA.

Multi-Dimensional Scaling

- A different goal :
  - Find a set of points whose pairwise distances match a given distance matrix

	p1	**p2**	**p3**	**p4**	**p5**
p1	0	1	2	3	1
p2	1	0	2	4	1
p3	2	2	0	1	3
p4	3	4	1	0	1
p5	1	1	3	1	0



Classical MDS

- Given n x n matrix of pairwise distances between data points
- Compute n x k matrix X with coordinates of distances with some linear algebra magic
- Perform PCA on this matrix X

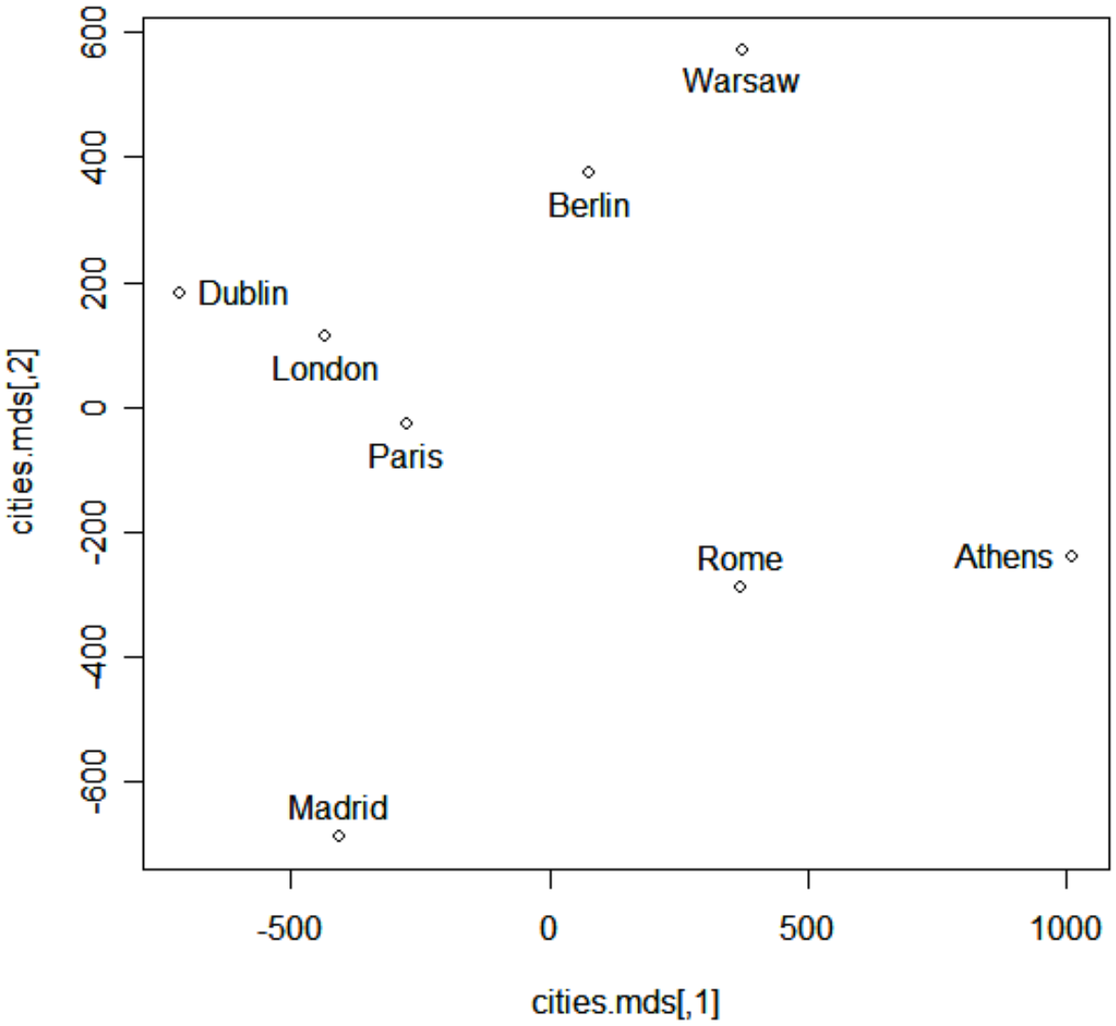
European Cities Data

- Distances between European cities:

	Athens	Berlin	Dublin	London	Madrid	Paris	Rome	Warsaw
Athens	0	1119	1777	1486	1475	1303	646	1013
Berlin	1119	0	817	577	1159	545	736	327
Dublin	1777	817	0	291	906	489	1182	1135
London	1486	577	291	0	783	213	897	904
Madrid	1475	1159	906	783	0	652	856	1483
Paris	1303	545	489	213	652	0	694	859
Rome	646	736	1182	897	856	694	0	839
Warsaw	1013	327	1135	904	1483	859	839	0

Result of MDS

Preserve original distance.

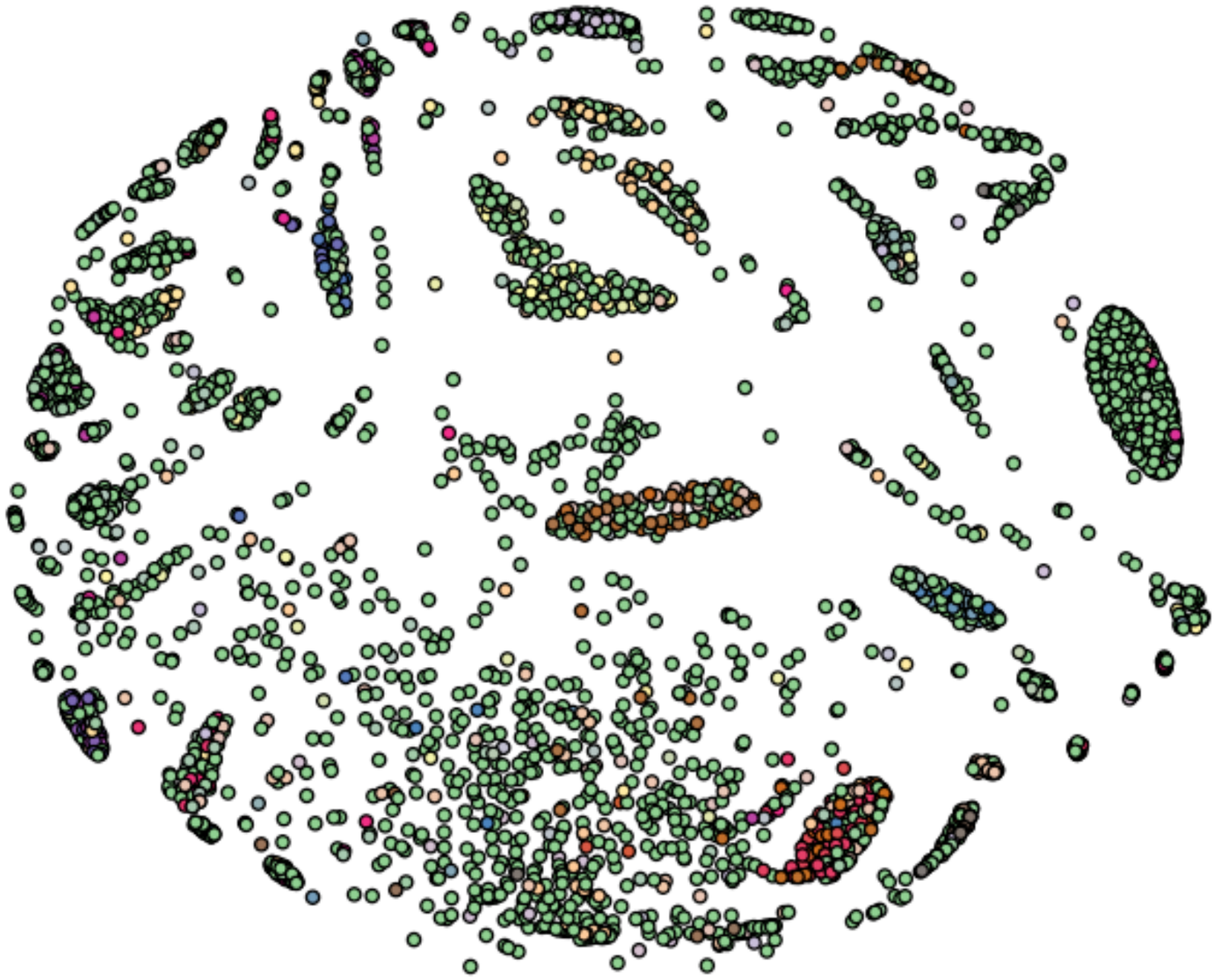


Based on slide from T. Yang

Color Images

N. Bonneel

Facebook Friends



– Distance = 1 for friends

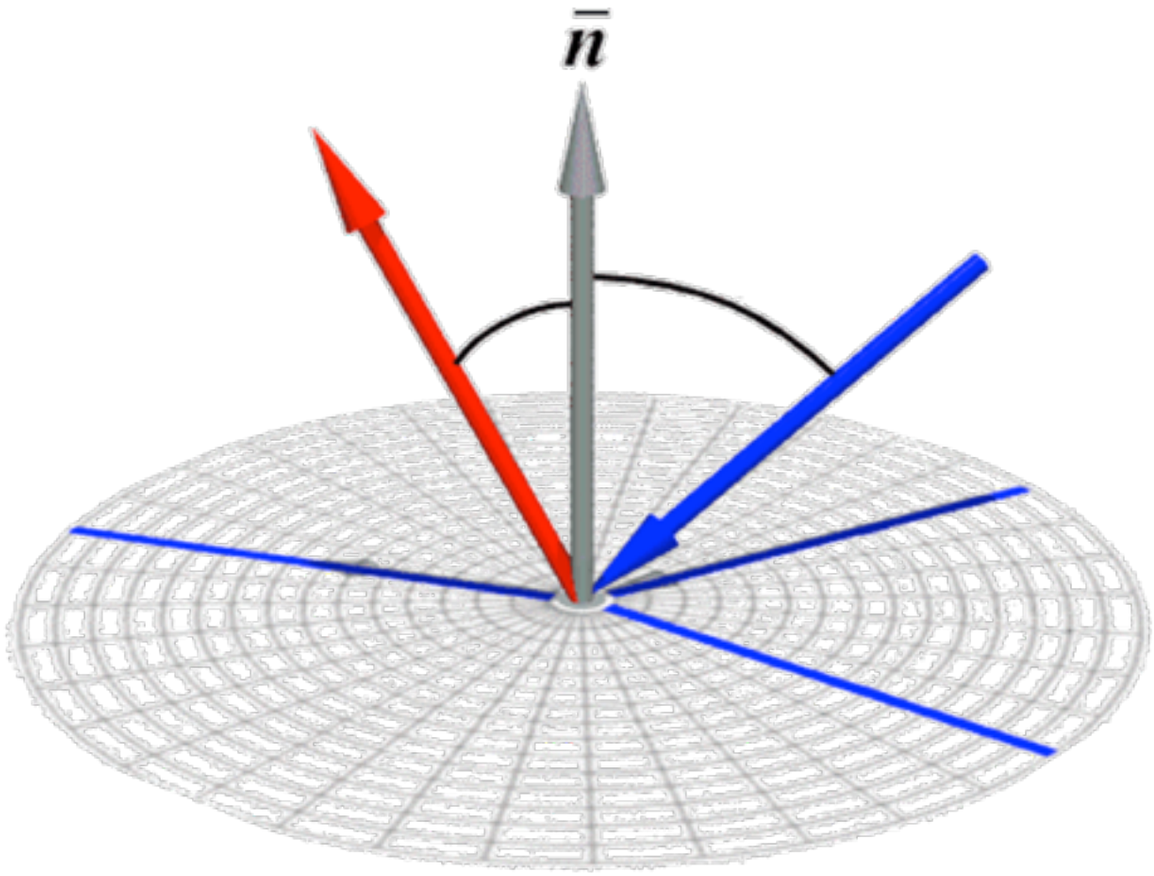
– Distance = 2 for friends of friends ; etc.

N. Bonneel

Nonlinear Methods

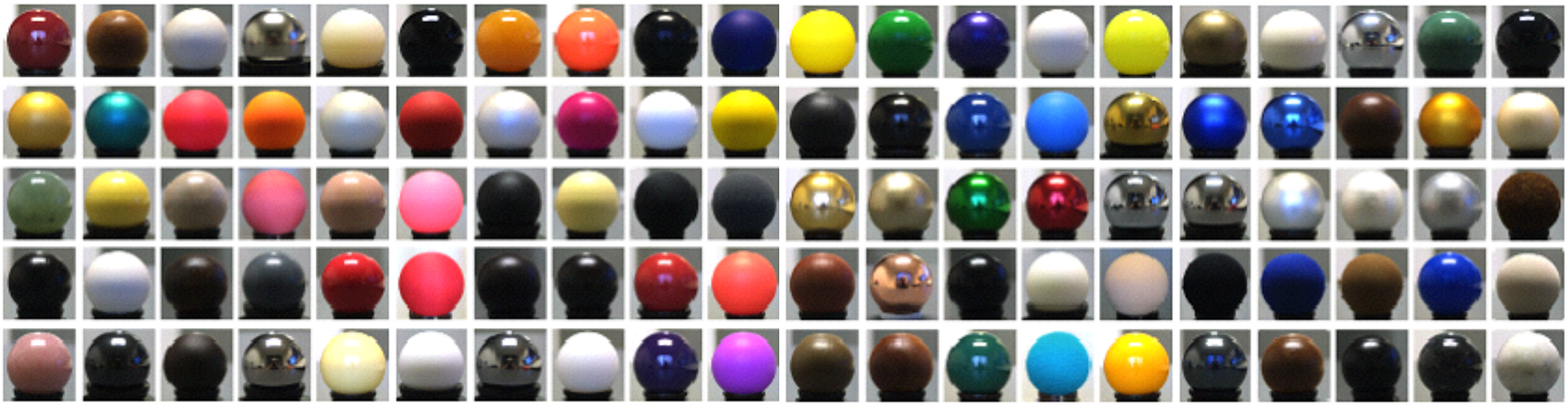
Data-Driven BRDFs

- Bi-Directional Reflectance Distribution Functions



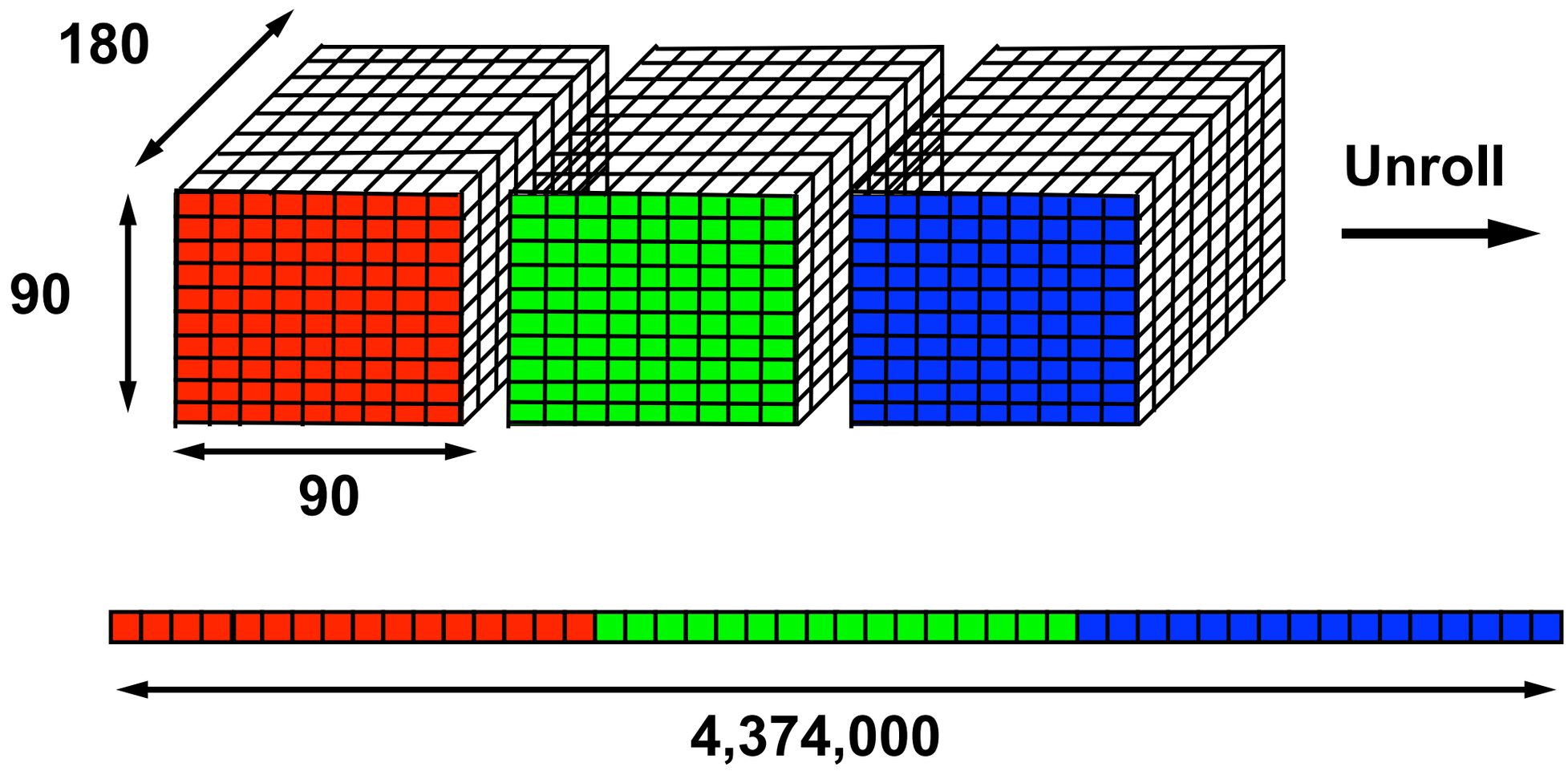
Data-Driven BRDFs

- Measure light reflected off a sphere
- 20-80 million measurements (6000 images) per material

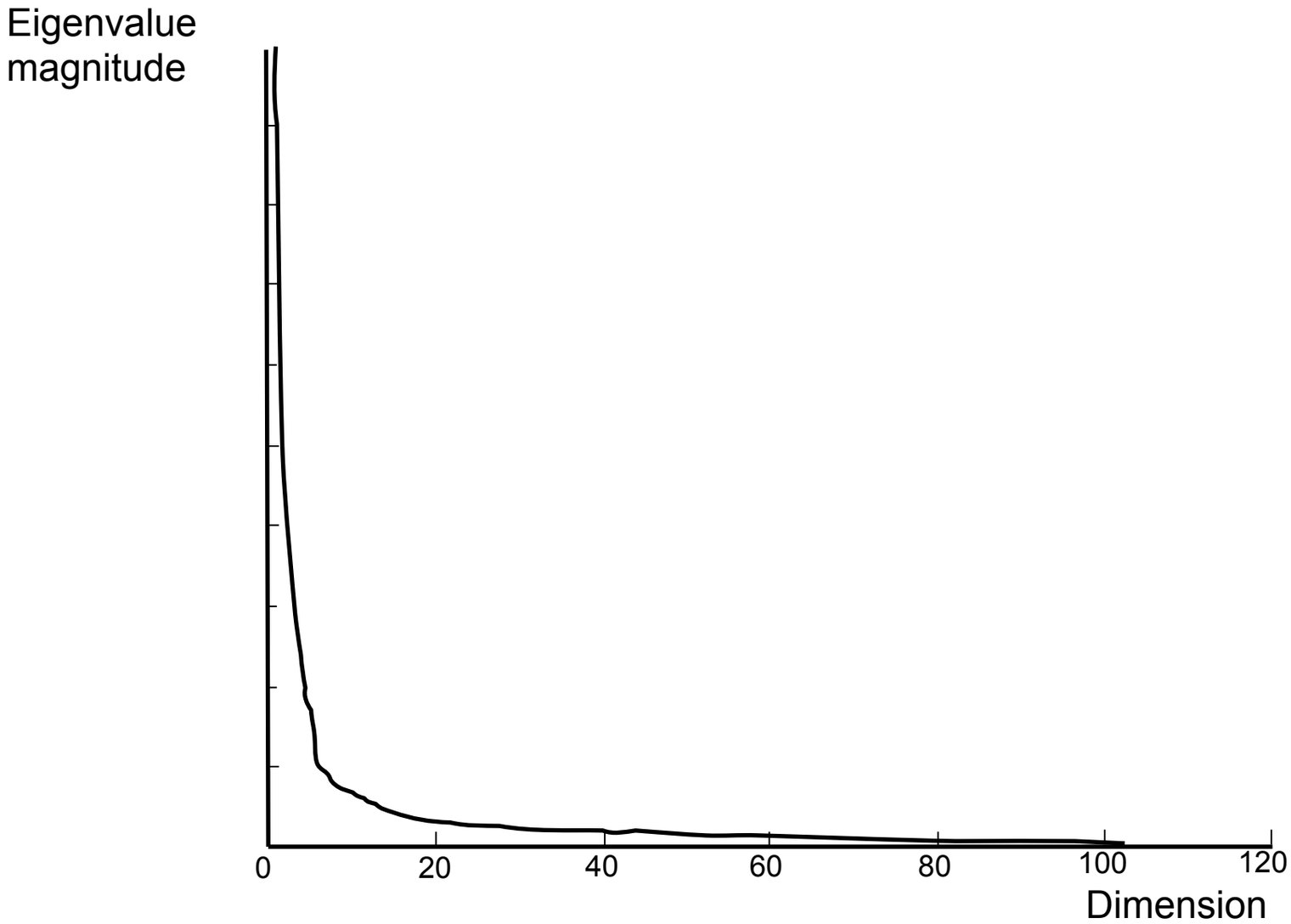


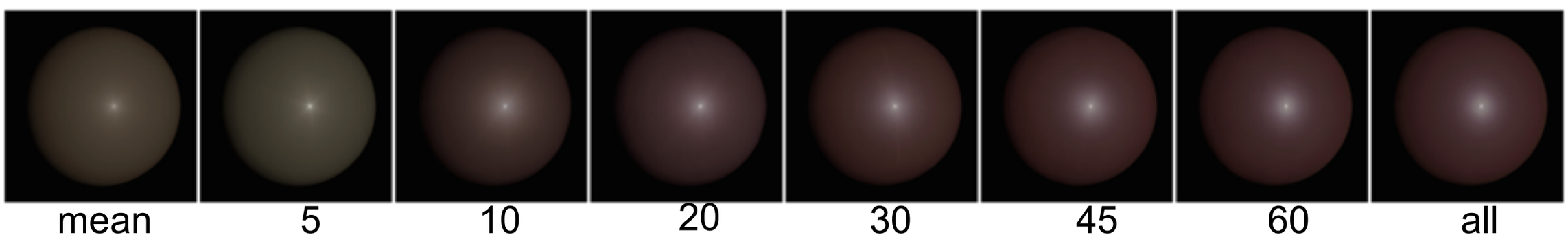
Data-Driven BRDFs

- Each tabulated BRDF is a vector in 90x90x180x3 =4,374,000 dimensional space



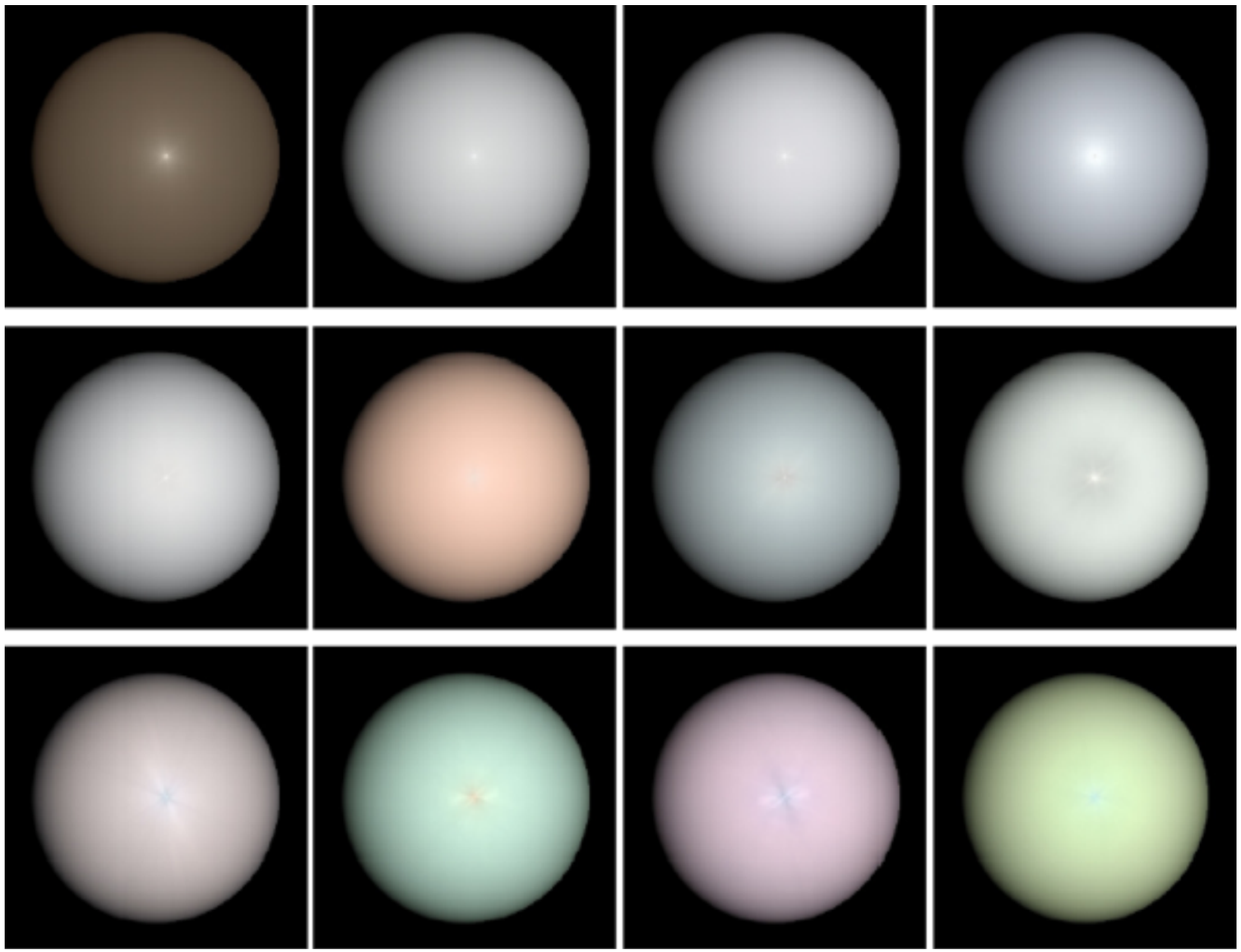
PCA


Eigenvalue
magnitude
0
20
40
60
80
100
120
Dimension


mean
5
10
20
30
45
60
all

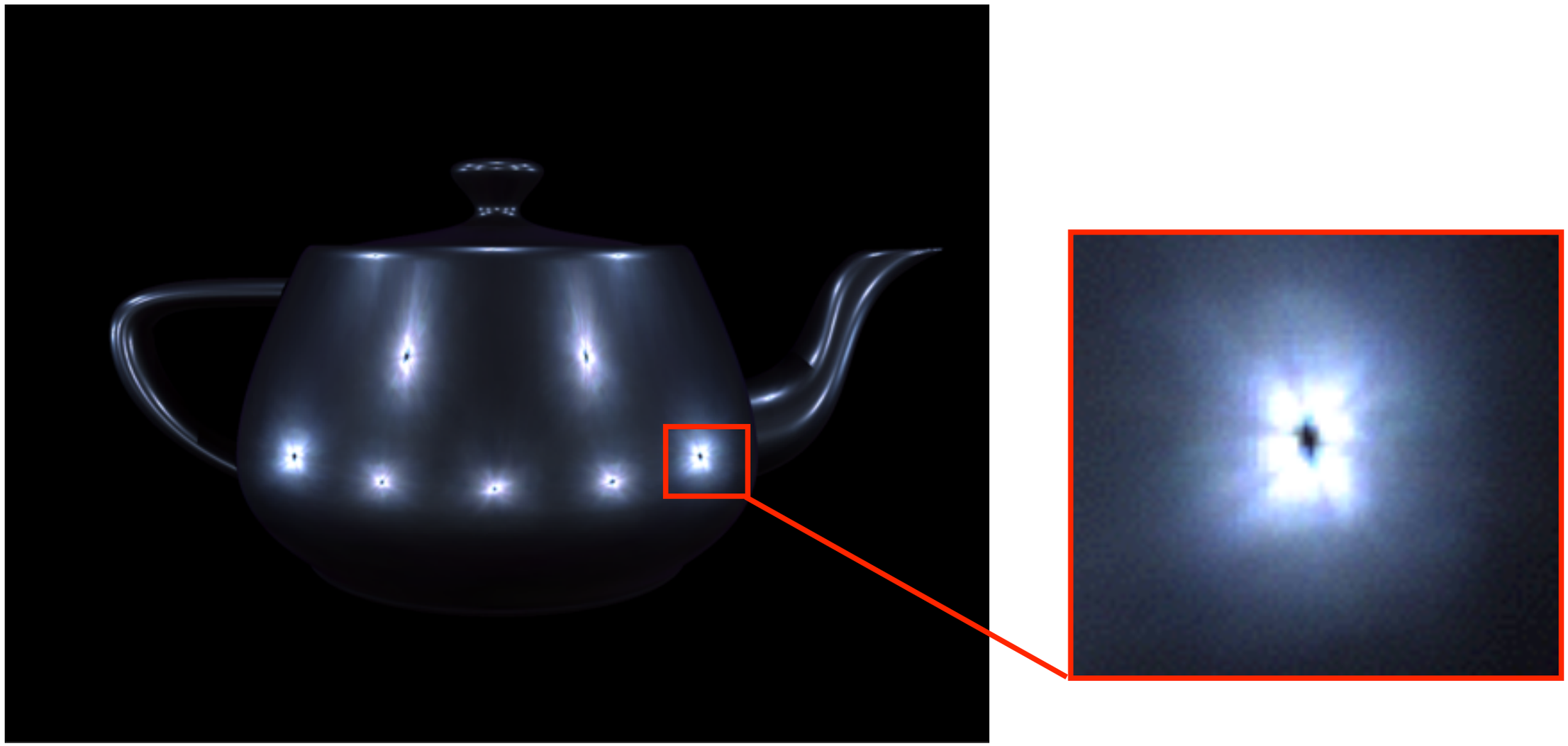
PCA

- First 11 PCA components

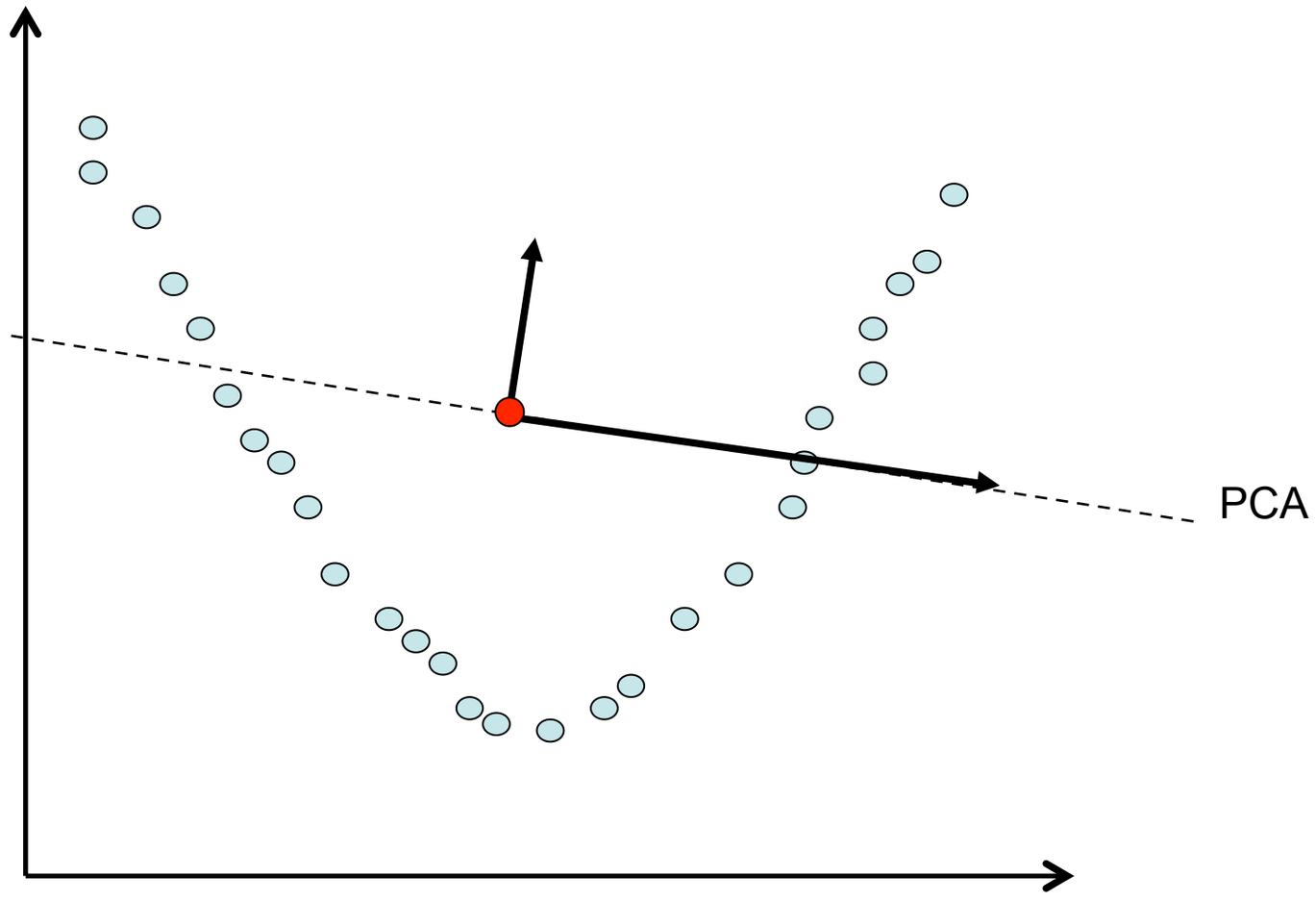


PCA Interpolation

Then, one day...

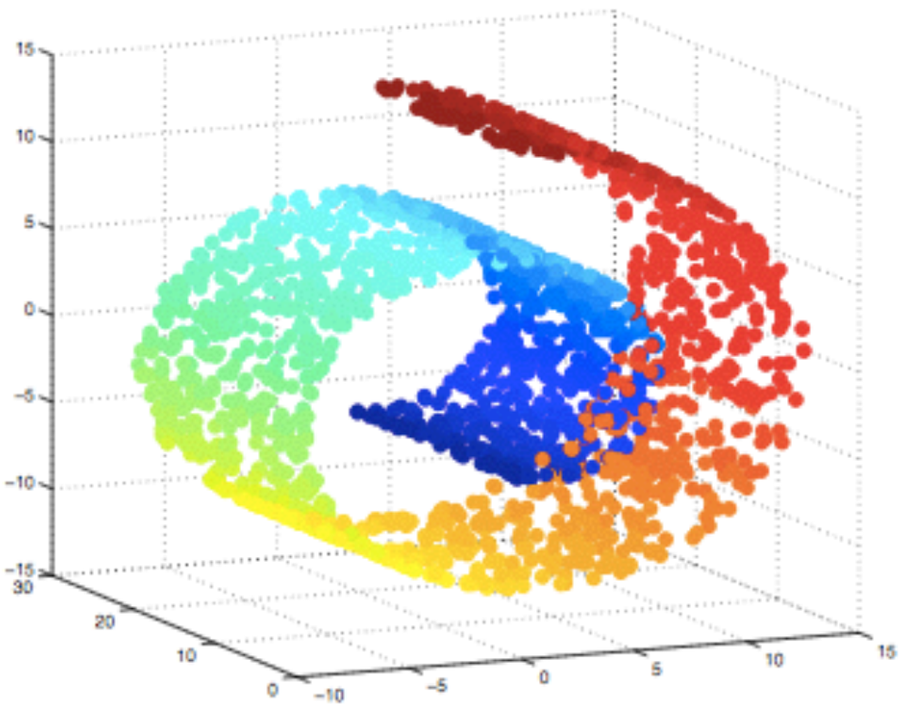


Why do linear models fail?

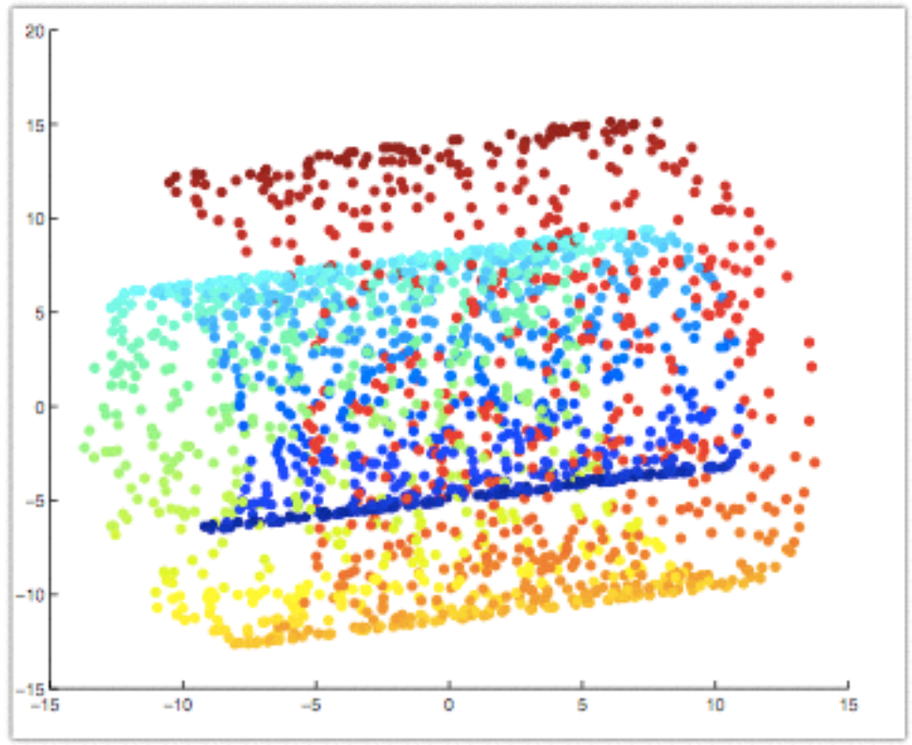


Why do linear models fail?

- Classic "Swiss Roll" example



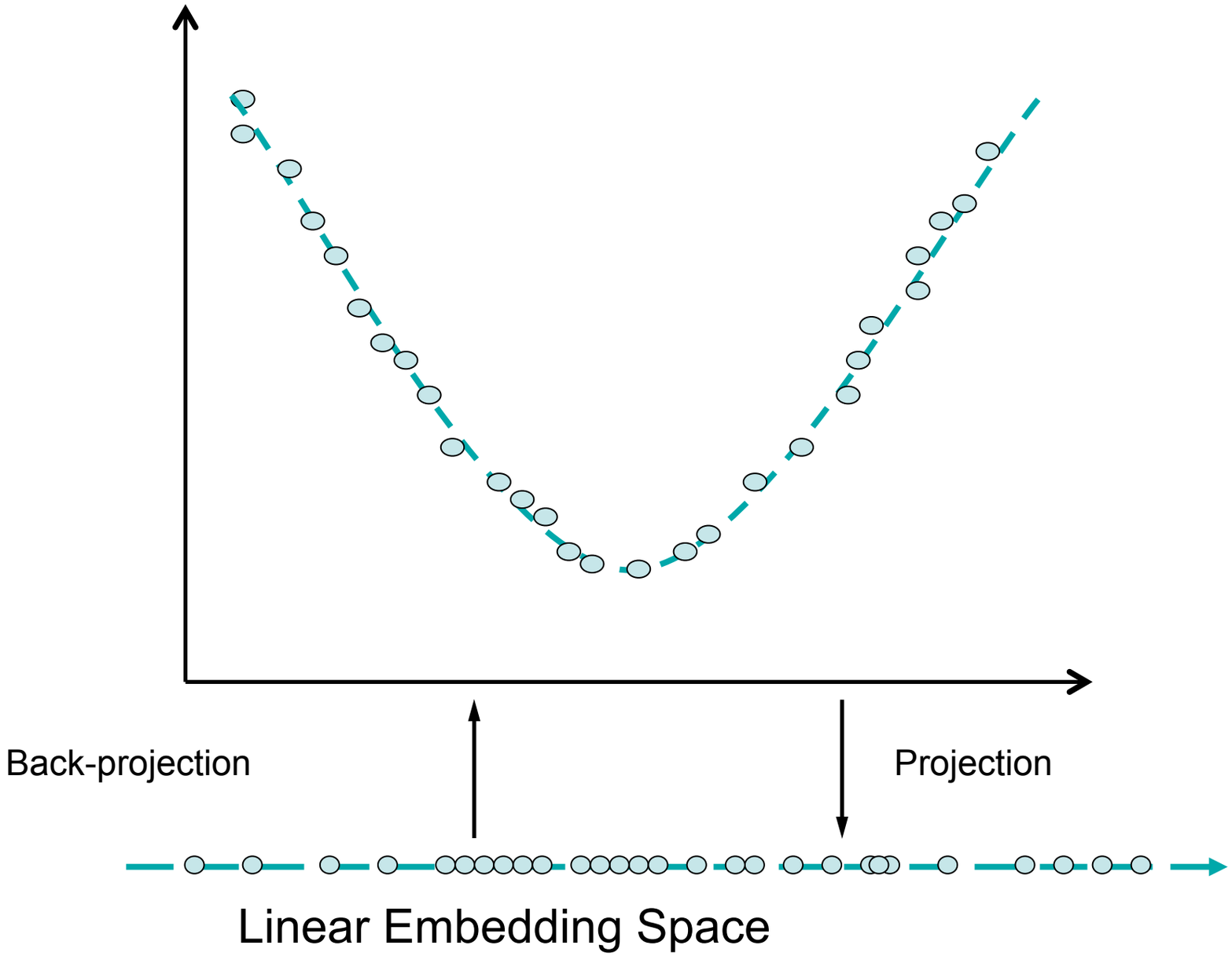
x_i



PCA

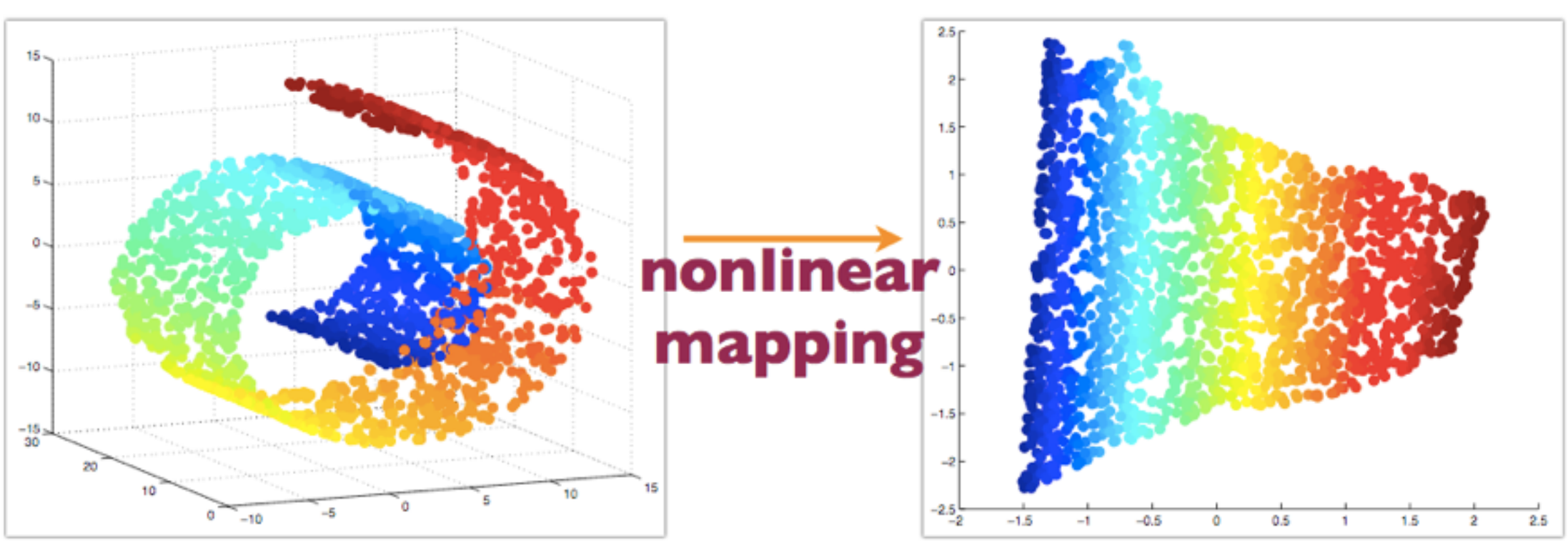
Based on slide from F. Sha

Non-Linear Manifold Methods



Non-Linear Manifold Methods

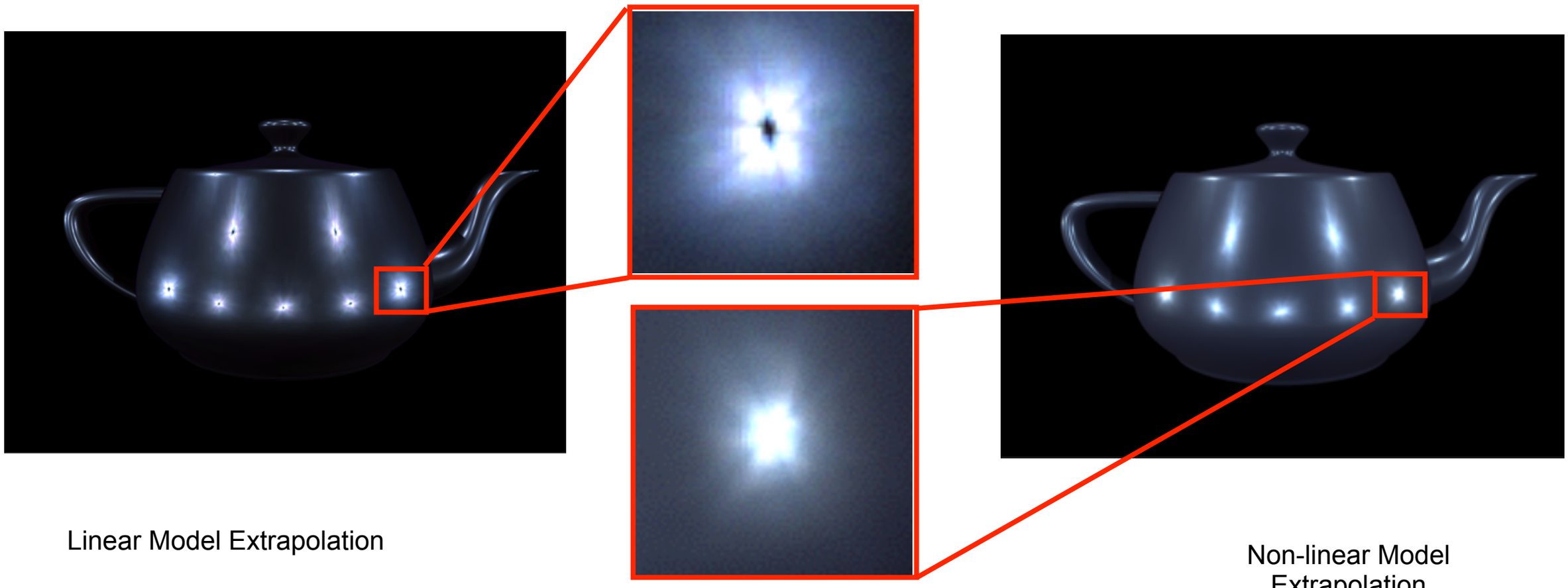
- Intuition: Distortion in local areas, but faithful in the global structure



Based on slide from F. Sha

Non-Linear BRDF Model

- 15-dimensional space (instead of 45 PCs)
- More robust - allows extrapolations



Linear Model Extrapolation

Non-linear Model Extrapolation

Dimensionality Reduction

- Linear methods:
  - Principal Component Analysis (PCA) – Hotelling[33]
  - Singular Value Decomposition (SVD) – Eckart/Young[36]
  - Multidimensional Scaling (MDS) – Young[38]
- Nonlinear methods:
  - IsoMap – Tenenbaum[00]
  - Locally Linear Embeddings (LLE) – Roweis[00]

Further Reading

CS231n Convolutional Neural Networks for Visual Recognition

These notes accompany the Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition. Feel free to ping @karpathy if you spot any mistakes or issues, or submit a pull request to our git repo.
We encourage the use of the hypothes.is extension to annote comments and discuss these notes inline.

Assignments

Assignment #1: Image Classification, kNN, SVM, Softmax

Assignment #2: Neural Networks, ConvNets I

Assignment #3: ConvNets II, Transfer Learning, Visualization

Module 0: Preparation

Python / Numpy Tutorial

IPython Notebook Tutorial

Terminal.com Tutorial

http://cs231n.github.io

Module 1: Neural Networks

Image Classification: Data-driven Approach, k-Nearest Neighbor, train/val/test splits

L1/L2 distances, hyperparameter search, cross-validation

Linear classification: Support Vector Machine, Softmax