Dimensionality Reduction Methods

Machine Learning Meet-Up

Brian Connolly

6/18/18

Ignignokt: You and your "third" dimension.

Frylock: What about it?

Ignignokt: Oh, nothing – it's cute. We have five.

Err: Thousand!

Ignignokt: Yes, five thousand.

Err: Don't question it!

Frylock: Oh yeah? Well, I only see two.

Ignignokt: Well, that sounds like a personal problem.

from *Aqua Teen Hunger Force*

Outline

- Overview of dimensionality reduction techniques
  - Supervised/unsupervised
- Unsupervised Methods
  - Principal Component Analysis
  - Multi-Dimensional Scaling
  - T-SNE
- Supervised Methods
  - Linear Discriminant Analysis
  - Neighborhood Component Analysis
- Example
  - Semi-supervised method for speaker identification

**** Going over a small sub-set of dimensionality reduction methods**

`sklearn.decomposition`: Matrix Decomposition

The `sklearn.decomposition` module includes matrix decomposition algorithms, including among others PCA, NMF or ICA. Most of the algorithms of this module can be regarded as dimensionality reduction techniques.

User guide: See the Decomposing signals in components (matrix factorization problems) section for further details.

`decomposition.DictionaryLearning` ([…])	Dictionary learning
`decomposition.FactorAnalysis` ([n_components, …])	Factor Analysis (FA)
`decomposition.FastICA` ([n_components, …])	FastICA: a fast algorithm for Independent Component Analysis.
`decomposition.IncrementalPCA` ([n_components, …])	Incremental principal components analysis (IPCA).
`decomposition.KernelPCA` ([n_components, …])	Kernel Principal component analysis (KPCA)
`decomposition.LatentDirichletAllocation` ([…])	Latent Dirichlet Allocation with online variational Bayes algorithm
`decomposition.MiniBatchDictionaryLearning` ([…])	Mini-batch dictionary learning
`decomposition.MiniBatchSparsePCA` ([…])	Mini-batch Sparse Principal Components Analysis
`decomposition.NMF` ([n_components, init, …])	Non-Negative Matrix Factorization (NMF)
`decomposition.PCA` ([n_components, copy, …])	Principal component analysis (PCA)
`decomposition.SparsePCA` ([n_components, …])	Sparse Principal Components Analysis (SparsePCA)
`decomposition.SparseCoder` (dictionary[, …])	Sparse coding
`decomposition.TruncatedSVD` ([n_components, …])	Dimensionality reduction using truncated SVD (aka LSA).
`decomposition.dict_learning` (X, n_components, …)	Solves a dictionary learning matrix factorization problem.
`decomposition.dict_learning_online` (X[, …])	Solves a dictionary learning matrix factorization problem online.
`decomposition.fastica` (X[, n_components, …])	Perform Fast Independent Component Analysis.
`decomposition.sparse_encode` (X, dictionary[, …])	Sparse coding

`sklearn.manifold`: Manifold Learning

The `sklearn.manifold` module implements data embedding techniques.

User guide: See the Manifold learning section for further details.

`manifold.Isomap` ([n_neighbors, n_components, …])	Isomap Embedding
`manifold.LocallyLinearEmbedding` ([…])	Locally Linear Embedding
`manifold.MDS` ([n_components, metric, n_init, …])	Multidimensional scaling
`manifold.SpectralEmbedding` ([n_components, …])	Spectral embedding for non-linear dimensionality reduction.
`manifold.TSNE` ([n_components, perplexity, …])	t-distributed Stochastic Neighbor Embedding.
`manifold.locally_linear_embedding` (X, … [, …])	Perform a Locally Linear Embedding analysis on the data.
`manifold.smacof` (dissimilarities[, metric, …])	Computes multidimensional scaling using the SMACOF algorithm.
`manifold.spectral_embedding` (adjacency[, …])	Project the sample on the first eigenvectors of the graph Laplacian.

`sklearn.discriminant_analysis`: Discriminant Analysis

Linear Discriminant Analysis and Quadratic Discriminant Analysis

User guide: See the Linear and Quadratic Discriminant Analysis section for further details.

`discriminant_analysis.LinearDiscriminantAnalysis` ([…])	Linear Discriminant Analysis
`discriminant_analysis.QuadraticDiscriminantAnalysis` ([…])	Quadratic Discriminant Analysis

Dimensionality Reduction Methods

- Generally used for ‘compressing’ high-dimensional information in a low dimensional space
- Visualization
  - Find patterns
  - Kernel

- Feature Engineering/Selection
  - Transform data into a bunch of variables that can best discriminate classes
  - Word embedding: forcing a language model into a low-dimensional space requires the ‘algorithm’ to consolidate the model into concepts
    - Extract meaning
- Data Compression (low-rank dimensionality reduction methods)
- Supervised classification
- Semi-supervised classification

- **Begin with unsupervised methods**
- **Learn by example: Principal Component Analysis (PCA)**

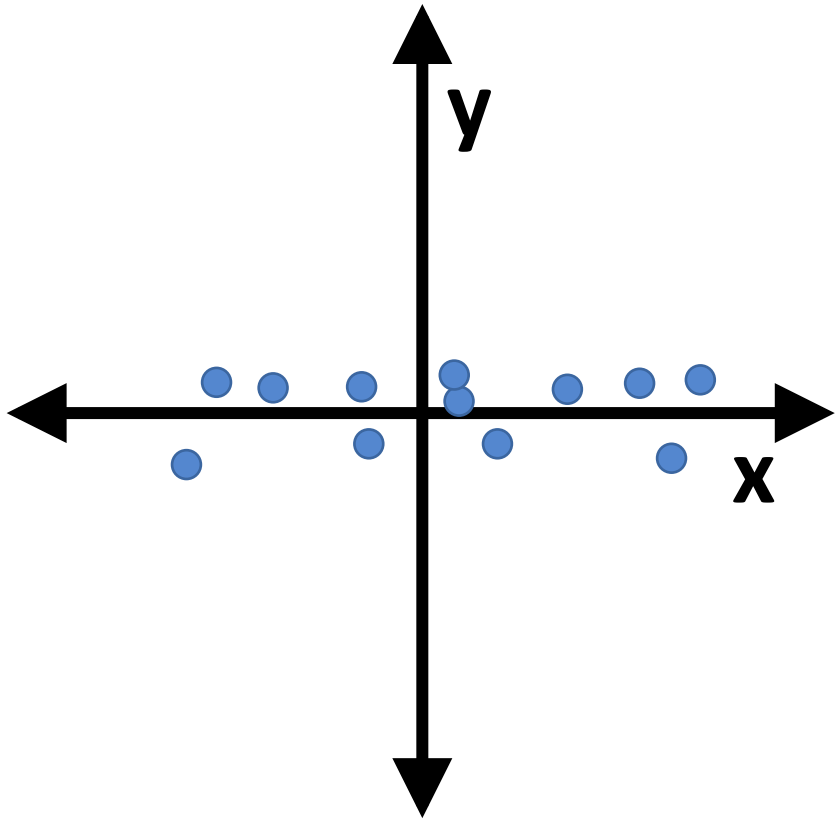
Principal Component Analysis

- Suppose you're stuck on a desert island to do an analysis on a two-dimensional data set (x and y), and you're told you can only take a single dimension with you?

- ***Which one would I choose?***

Principal Component Analysis

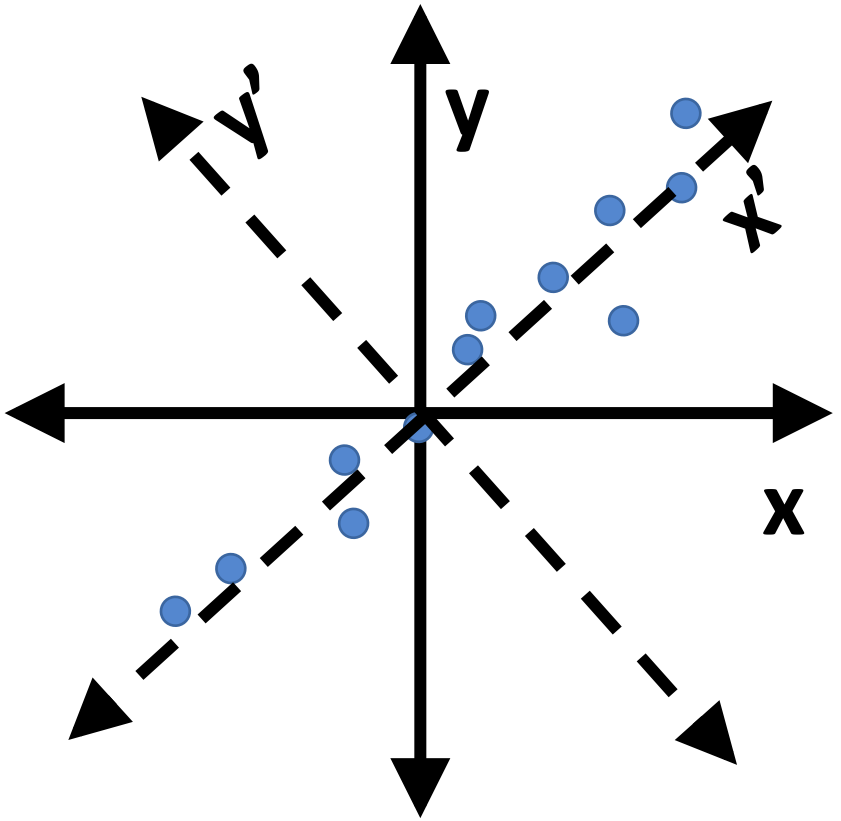
- Knowing nothing else, you'd choose the variable that varied the most from data point to data point
  - A variable whose value is always the same is not very interesting
- Find axis with the most variance
- X-axis!



Principal Component Analysis

- If points not along x or y axis, might ‘invent’ an axis, **x’**, then give the coordinates relative to that new axis and carry them off to your island paradise

- By focusing on one axis (however it might be oriented) with the most variability, I have ***reduced*** the number of ***dimensions*** I have to worry about



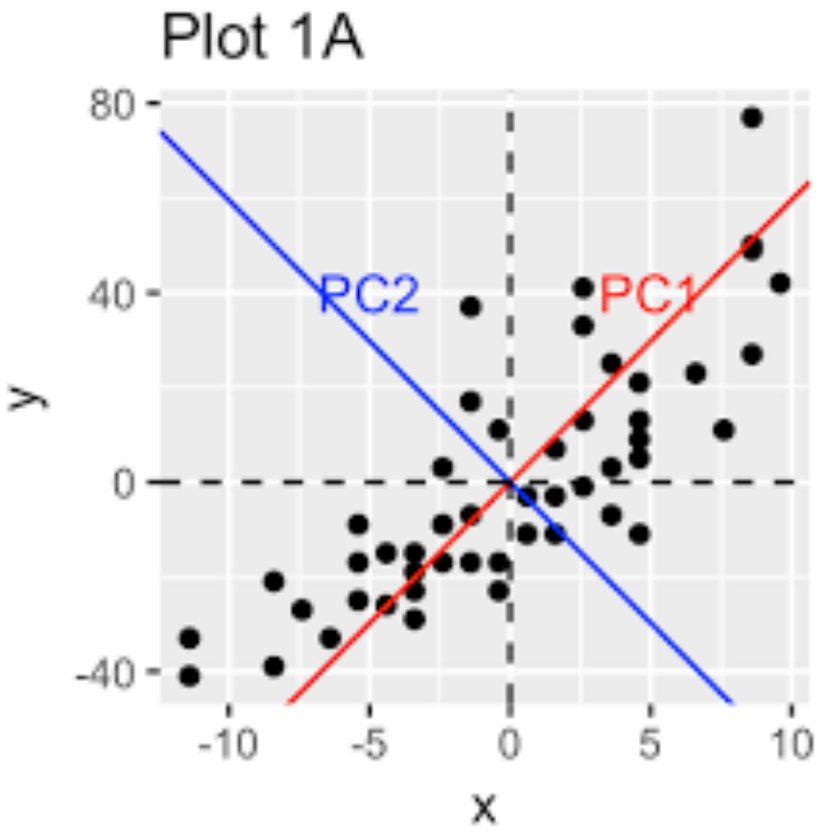
Principal Component Analysis

- Note even if I were not worried about reducing dimensions, I might want to do a PCA to 'rotate' my axis so that I can find the axes with the most variance (PC1), second most variance (PC2), etc.

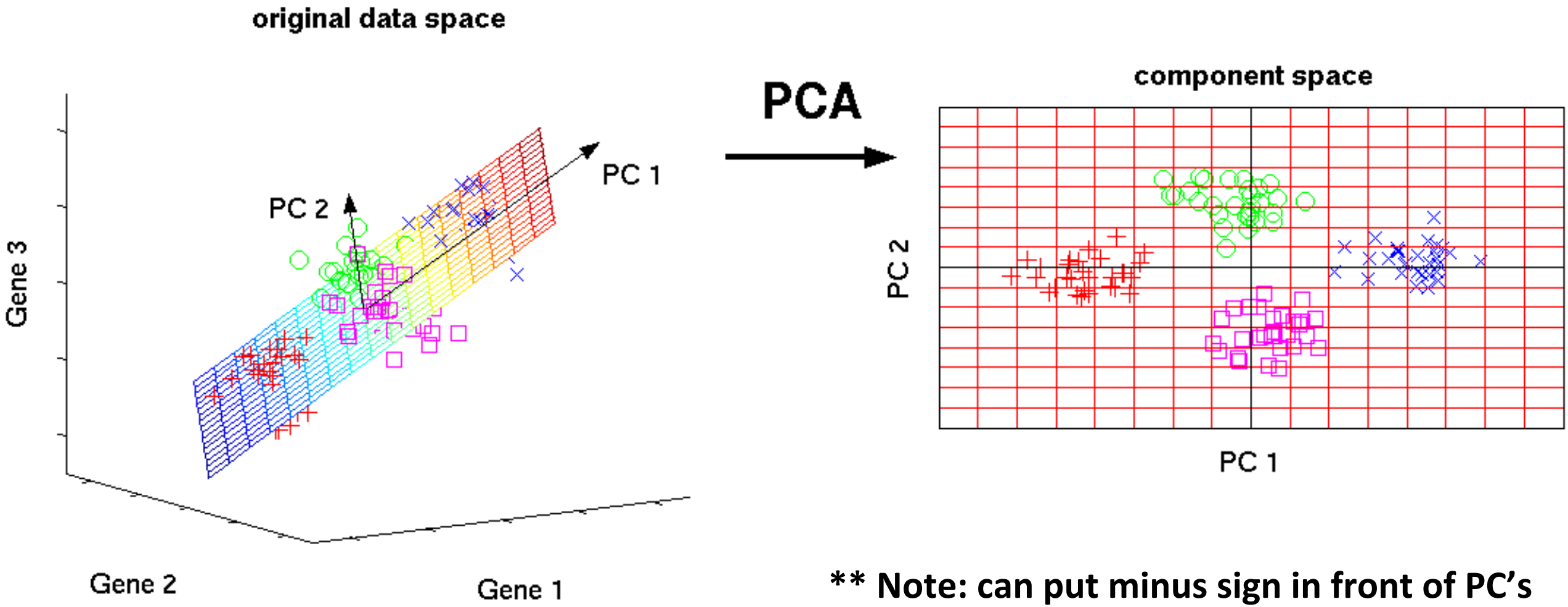
- In essence
  - PC1 = variable that is most important
  - PC 2 = variable that is second most important

- I can then use these new variables as input to my machine learning algorithm

- Strategy to visualize multi-dimensional data set: plot first and second components against each other to visualize how data are distributed
  - Put labels on points to determine if there is a separation between classes



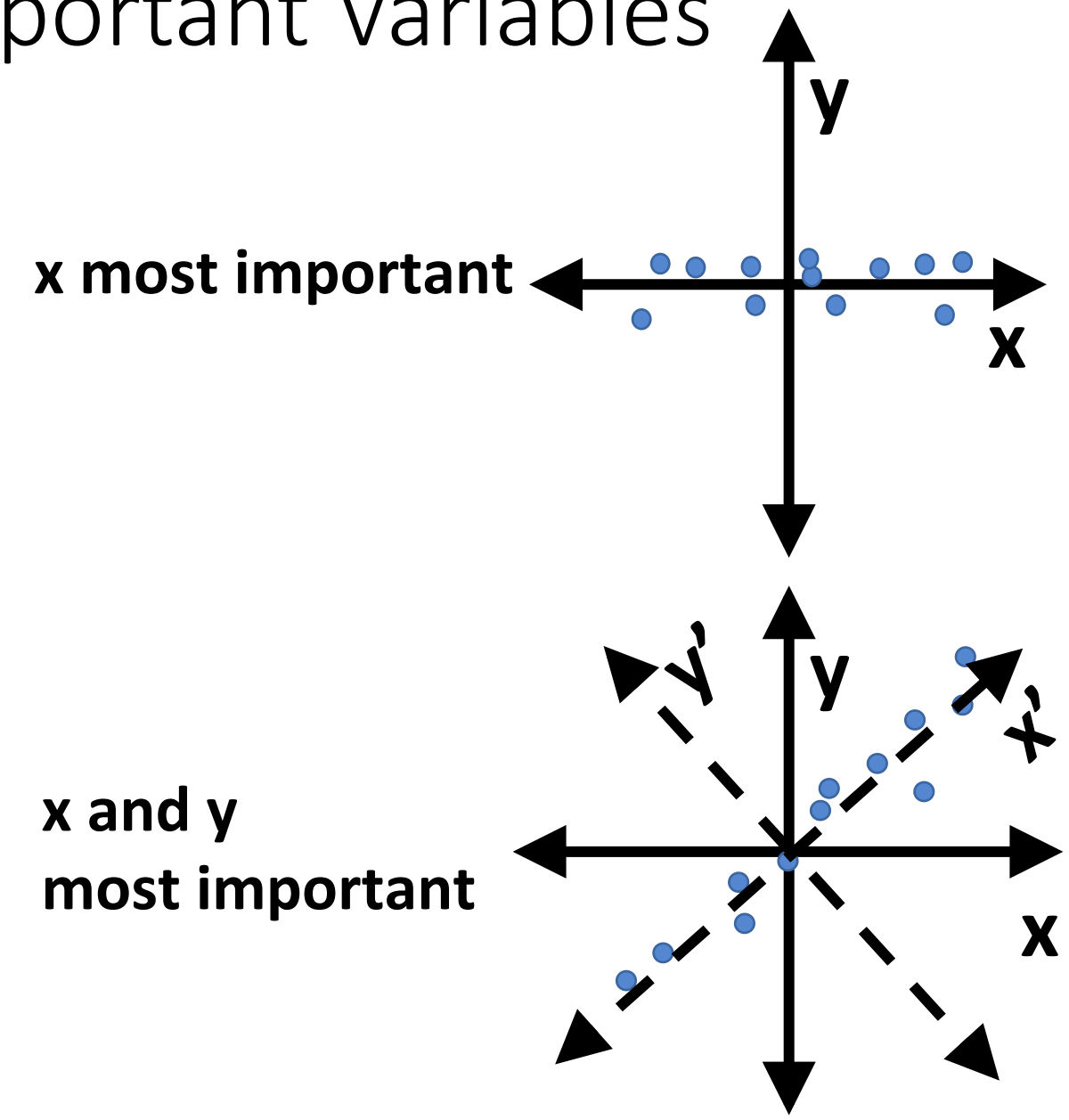
PCA: Looking at Separation of Classes



**** Note: can put minus sign in front of PC's**

PCA: Learn About Most Important Variables

- First few principal components tell you ‘directions’ (variables) with most variability

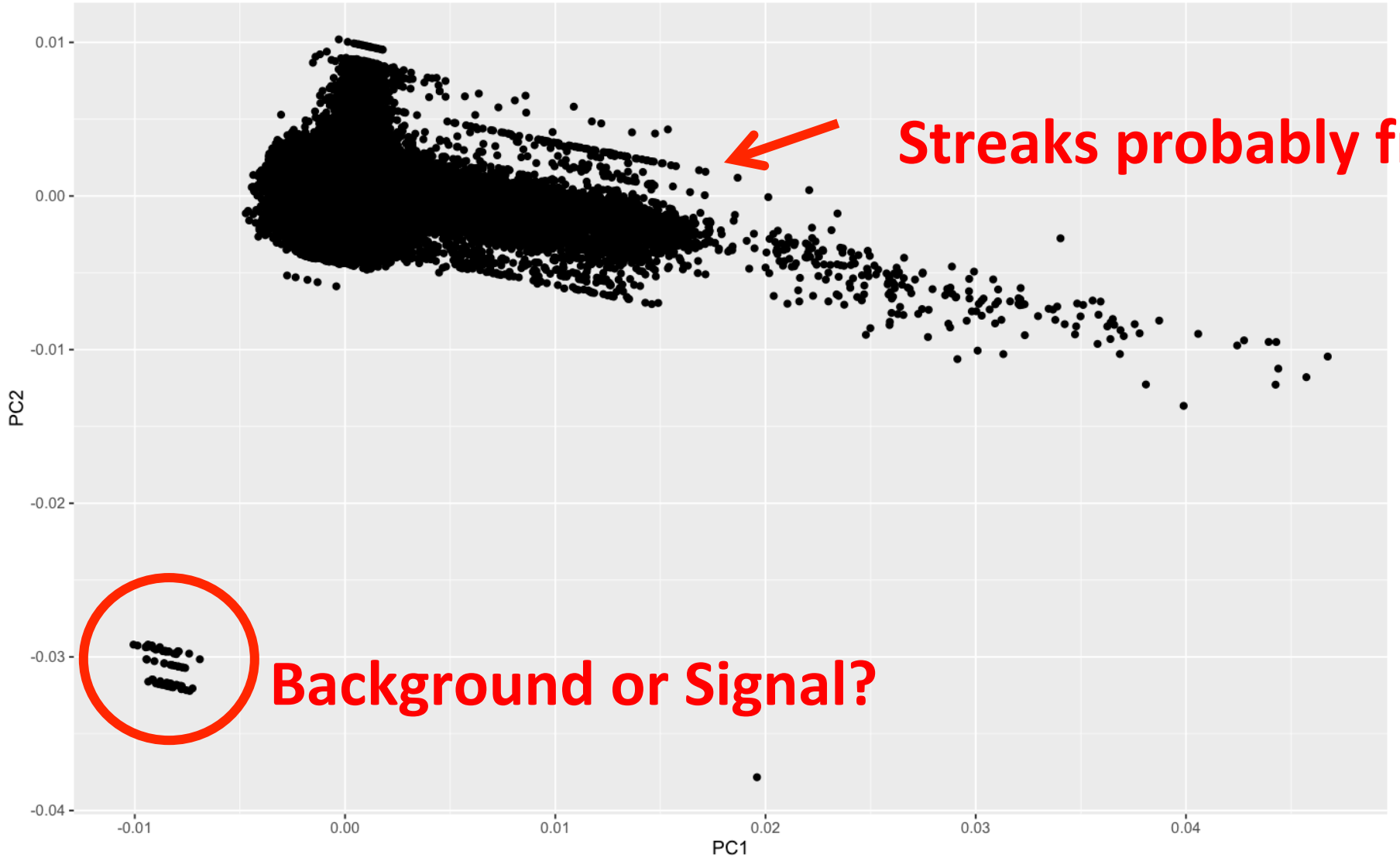


PCA: The Power of visualization

- Problems rejecting background noise in neutrino detectors
  - The **Antarctic Impulsive Transient Antenna** (**ANITA**) experiment has been designed to study ultra-high-energy(UHE) cosmic neutrinos by detecting the radio pulses emitted by their interactions with the Antarctic ice sheet.
  - Detect radio signals, but there is noise everywhere
  - Data come in the form of radio pulses and timing information spread over 14 dimensions

The ANITA-IV experiment in Antarctica, prior to being launched on a balloon.

PCA: The Power of visualization



PCA Summary

- Pros
  - Good first pass
  - Gives many insights into data
  - Everyone knows what PCA is, easy to understand
  - Good at telling what the variables that vary from data point to data point
  - Not very computationally expensive (easy to find packages that calculate PCA efficiently)

- Cons
  - Doesn't necessarily tell you what variables are best at discriminating between classes
  - Only tells you which variables have the most variability
  - Simple rotation (if transformation into 'interesting space' non-linear, not so great)
    - **Doesn't preserve non-linear boundaries as well as other techniques**

- Note: Singular Value Decomposition (SVD) is an extension of PCA

(Generalized) Multi-Dimensional Scaling

- Intuitive
- Better preserves non-linear boundaries
  - Important when determining/making sense of machine learning model
  - My SVM says a radial kernel works best – is this reasonable?
- Another technique to try if you don’t find anything with PCA

(Generalized) Multi-Dimensional Scaling

- Wikipedia: “In cases where the dissimilarities are distances on a surface and the target space is another surface, GMDS allows finding the minimum-distortion embedding of one surface into another.”

- Translation: connect every point in multi-dimensional space with spring and smash them into a lower-dimensional space (e.g., 2 dimensions)

(Generalized) Multi-Dimensional Scaling: Math

$\delta_{i,j} :=$ distance between i-th and j-th objects.

$$\Delta := \begin{pmatrix} \delta_{1,1} & \delta_{1,2} & \cdots & \delta_{1,I} \\ \delta_{2,1} & \delta_{2,2} & \cdots & \delta_{2,I} \\ \vdots & \vdots & & \vdots \\ \delta_{I,1} & \delta_{I,2} & \cdots & \delta_{I,I} \end{pmatrix}$$

$$\min_{x_1,\ldots,x_I} \sum_{i<j} (\|x_i - x_j\| - \delta_{i,j})^2$$

Goal: Find x_i and x_j in lower dimensional space s.t.

$$\|x_i - x_j\| \gtrsim \delta_{i,j}$$

Distance between points in lower-dimensional space

As you can imagine, by doing its best to preserve distances between points, this method gives more accurate representation of what boundary between points looks like

(Generalized) Multi-Dimensional Scaling

- Pros:
  - All the pros of PCA + preserves boundary
- Cons:
  - Can be computationally intensive
  - Not many people know about it
  - Still doesn’t determine best space to discriminate classes

t-SNE: Yet Another Unsupervised Dimensionality Reduction Method

- t-Distributed Stochastic Neighbor Embedding

- Won Merck Visualization challenge, so it's kind of a thing now
  - Merck Challenge: identify the best statistical techniques for predicting biological activities of different molecules, both on- and off-target, given numerical descriptors generated from their chemical structures.

- Sort of like MDS, but instead of trying its best to preserve distances, it tries to preserve probabilities that that objects are 'similar'

t-SNE Details

1. Calculate probability two points are similar

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)} \qquad p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

2. Calculate ‘distance’ between points in lower dimensional space

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$

3. Calculate how different p_{ij} is from q_{ij}

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Neat Application: Topic Modeling and t-SNE Visualization

- SHUAI'S AI & DATA BLOG
- 20 Newsgroups dataset
  - Various articles selected from users of Usenet
- Ignored the new groups, wanted to cluster the topics
- Used Latent Dirichlet Model to fuzzy cluster the articles, then transform the word frequency matrix into probabilities that the documents fell into 20 topics
- Then use t-SNE to knock that matrix down to two dimensions for visualization

Word Frequencies

Documents	pan	the	castle	...	megasaurus
	0	1	5	...	0
	.	.			.
	.	.			.
	.	.			.
	3	0	1	...	1

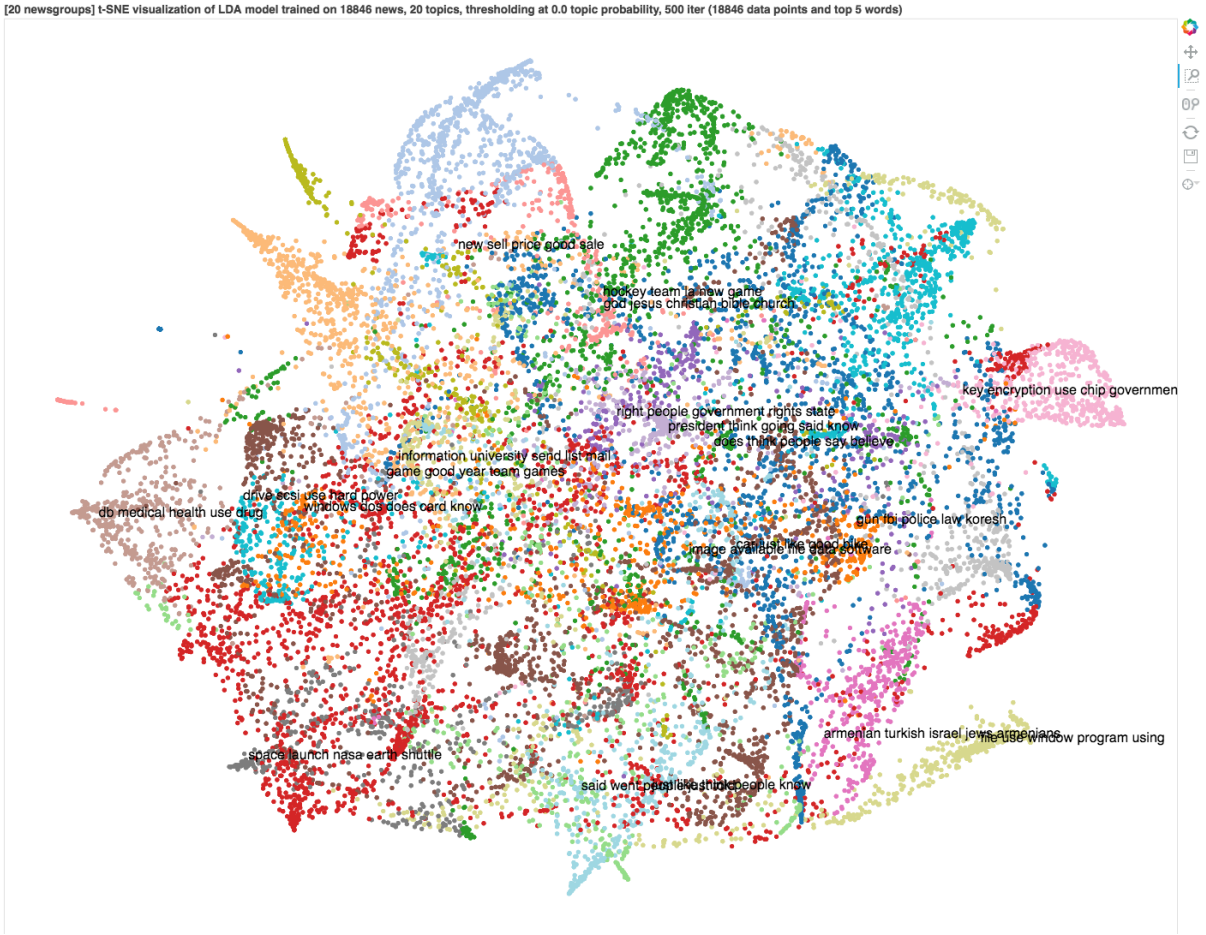
Neat Application: Topic Modeling and t-SNE Visualization

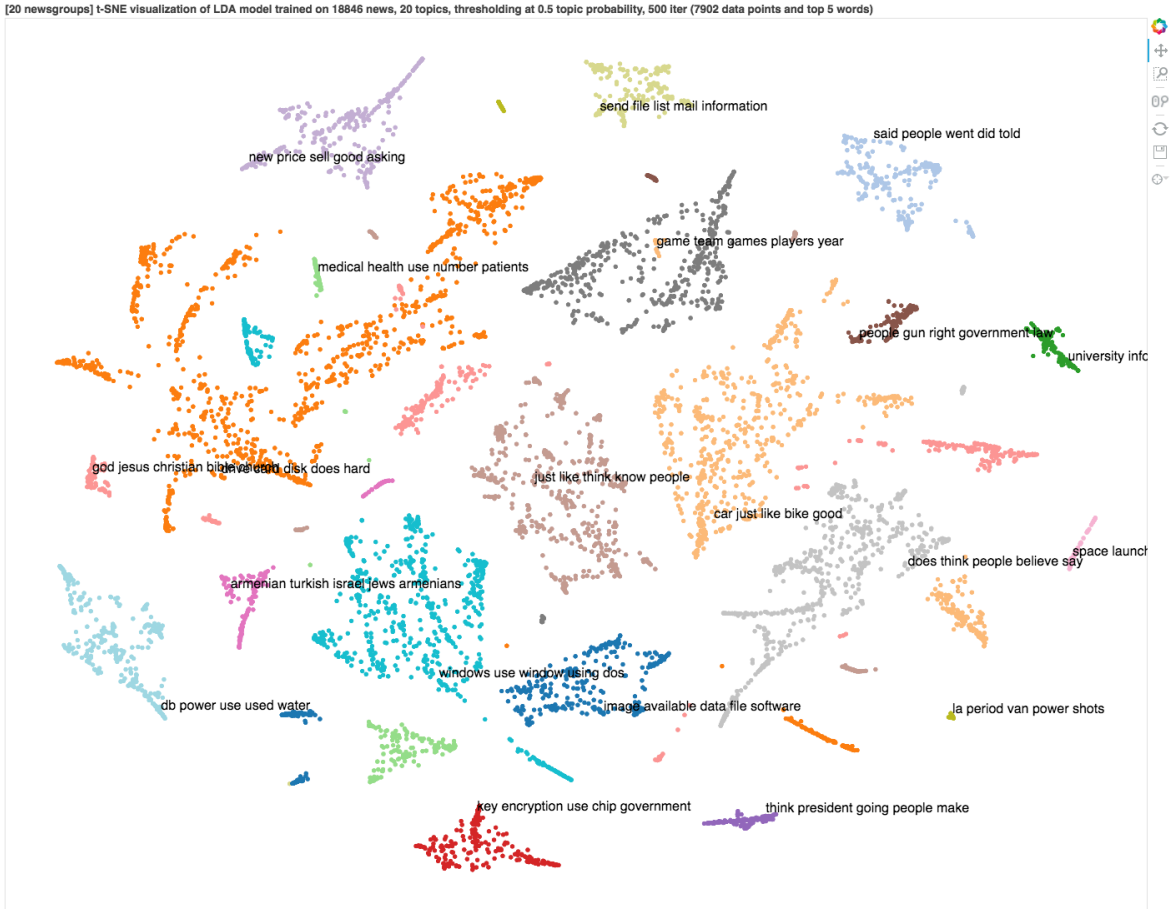
- SHUAI'S AI & DATA BLOG
- 20 Newsgroups dataset
  - Various articles selected from users of Usenet
- Ignored the new groups, wanted to cluster the topics
- Used Latent Dirichlet Model to fuzzy cluster the articles, then transform the word frequency matrix into probabilities
- Then use t-SNE to knock that matrix down to two dimensions for visualization

Probabilities of Topics

Documents	Topic1	Topic 2	.....	Topic 20
	0.0	0.1	0.1 0.3 …	0.5
	.	.		.
	.	.		.
	.	.		.
	0.5	0.1	0.3 0.0 …	0.0

Neat Application: Topic Modeling and t-SNE Visualization





t-SNE: Pros/Cons

- Pros:
  - Obviously good for clustering a very complex data set
  - Mathematical formalism makes good sense

- Cons:
  - Hand, David J. "Classifier technology and the illusion of progress." *Statistical science* (2006): 1-14.
  - From my arm chair: aspects seems a little hackish – can imagine it might be limited in some way
  - Can be computationally expensive
  - Problems with t-SNE arise when intrinsic dimensions are higher i.e. more than 2-3 dimensions. t-SNE has the tendency to get stuck in local optima like other gradient descent based algorithms
  - Have to tune “Perplexity”
  - If you’re doing a preliminary look at the data might be overkill?

Supervised Methods

Supervised Methods

- Have looked at unsupervised methods
  - Don’t use labels in any way

- Turn to supervised methods where we want to transform the data in a (lower-dimensional space) where you can best differentiate classes

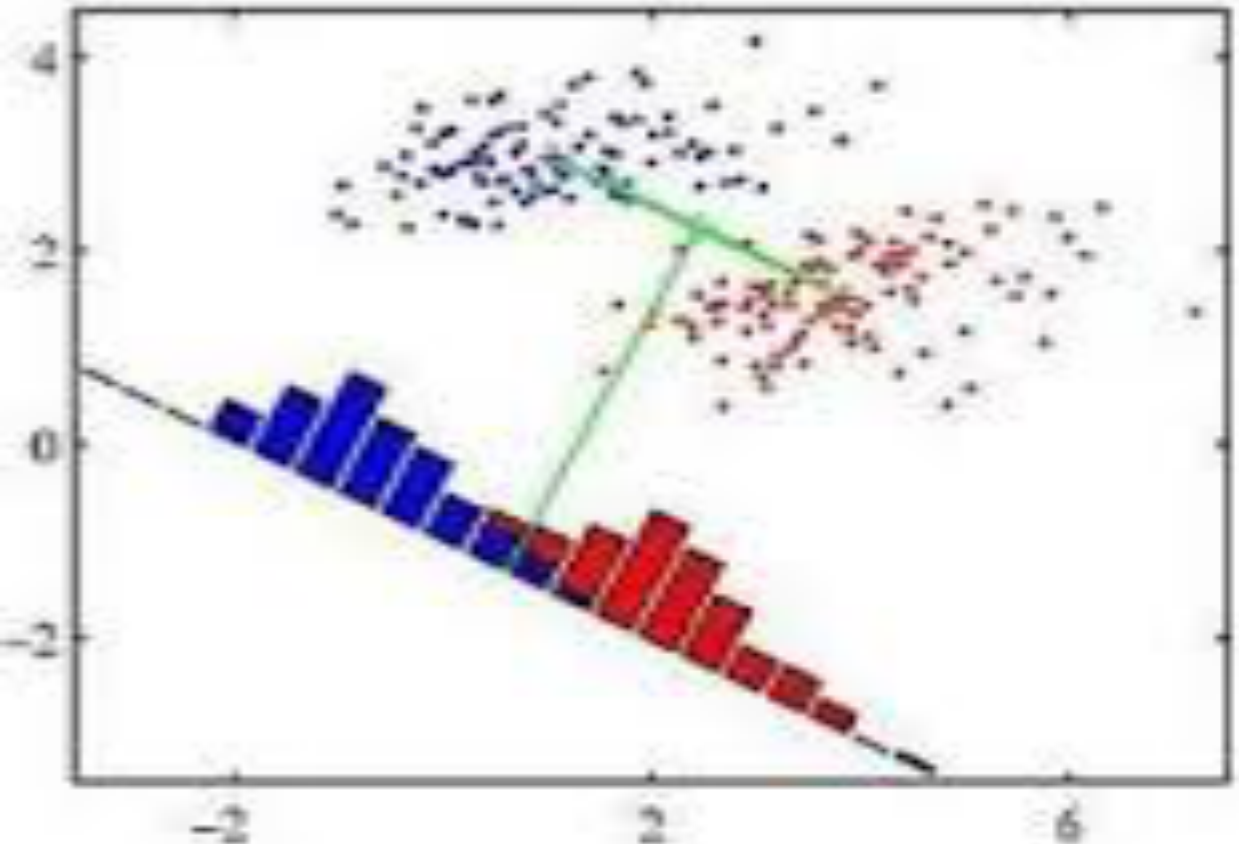
First Such Method: Linear Discriminant Analysis

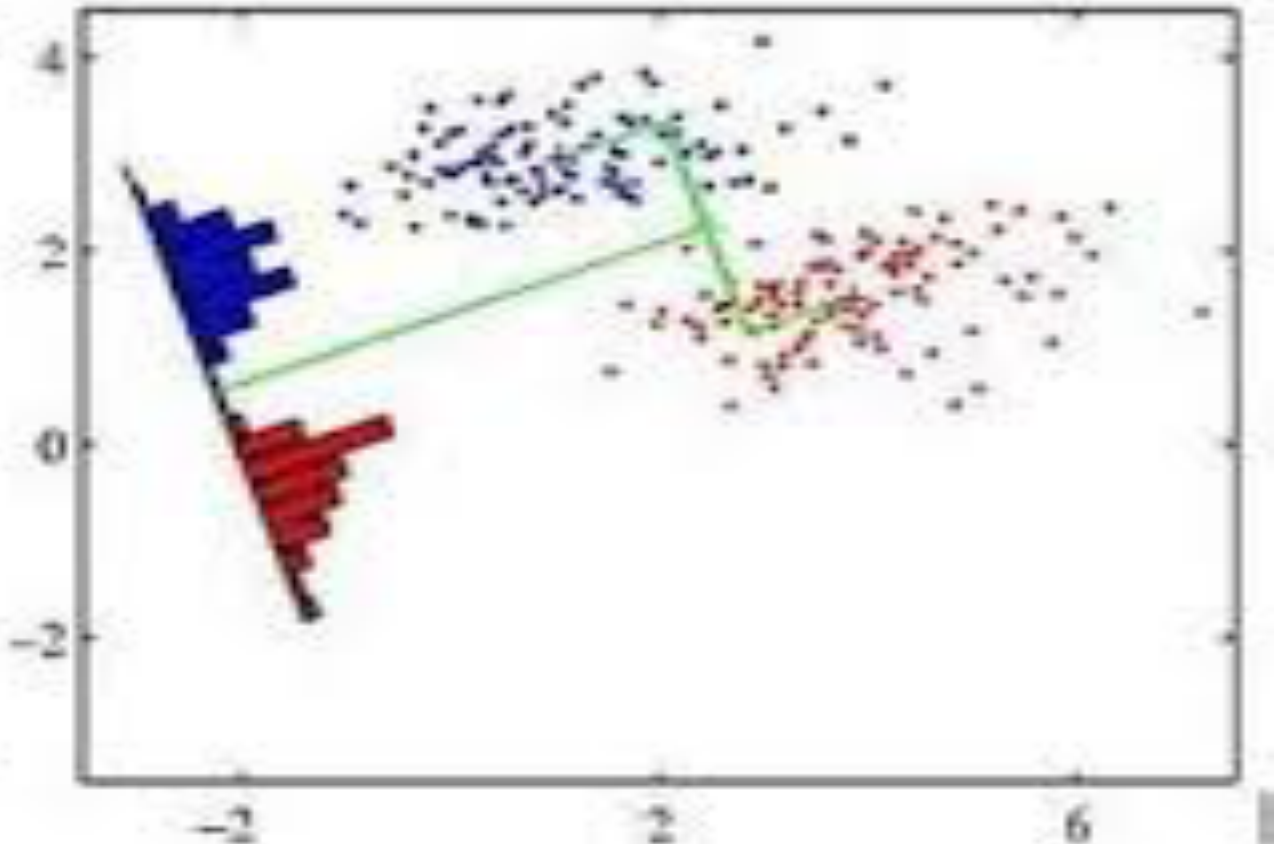
- Invented by Ronald Fisher in the 30's, really first (machine learning) classifier
- Called LDA not to be confused with Latent Dirichlet Allocation (also known as LDA)
- Idea: find a way to transform the data so that it minimizes the within-class variability and maximize the distance between the mean values of the classes

Whuh?

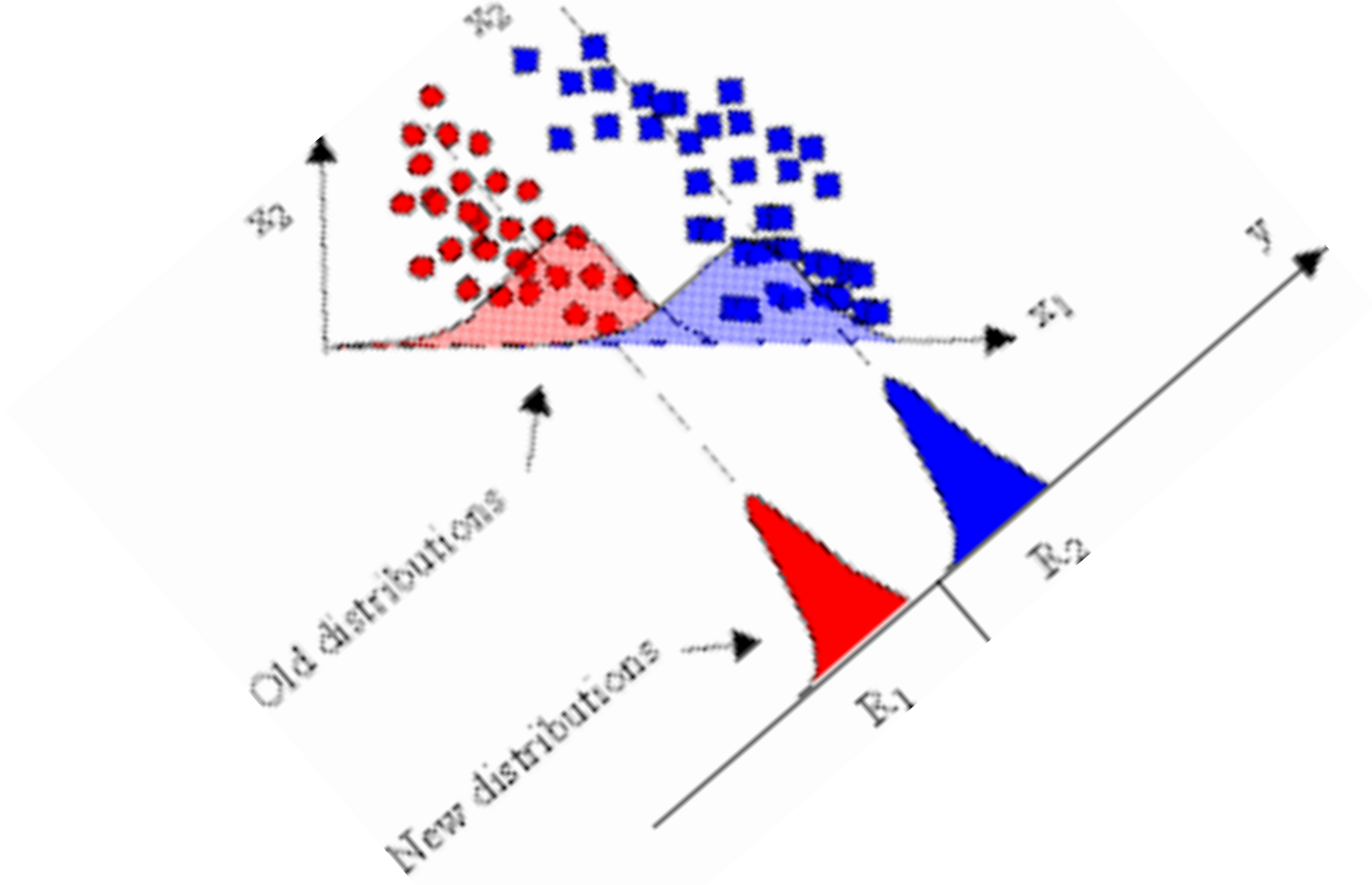
Linear Discriminant Analysis

- Find a boundary (plane or vector in 2d) that

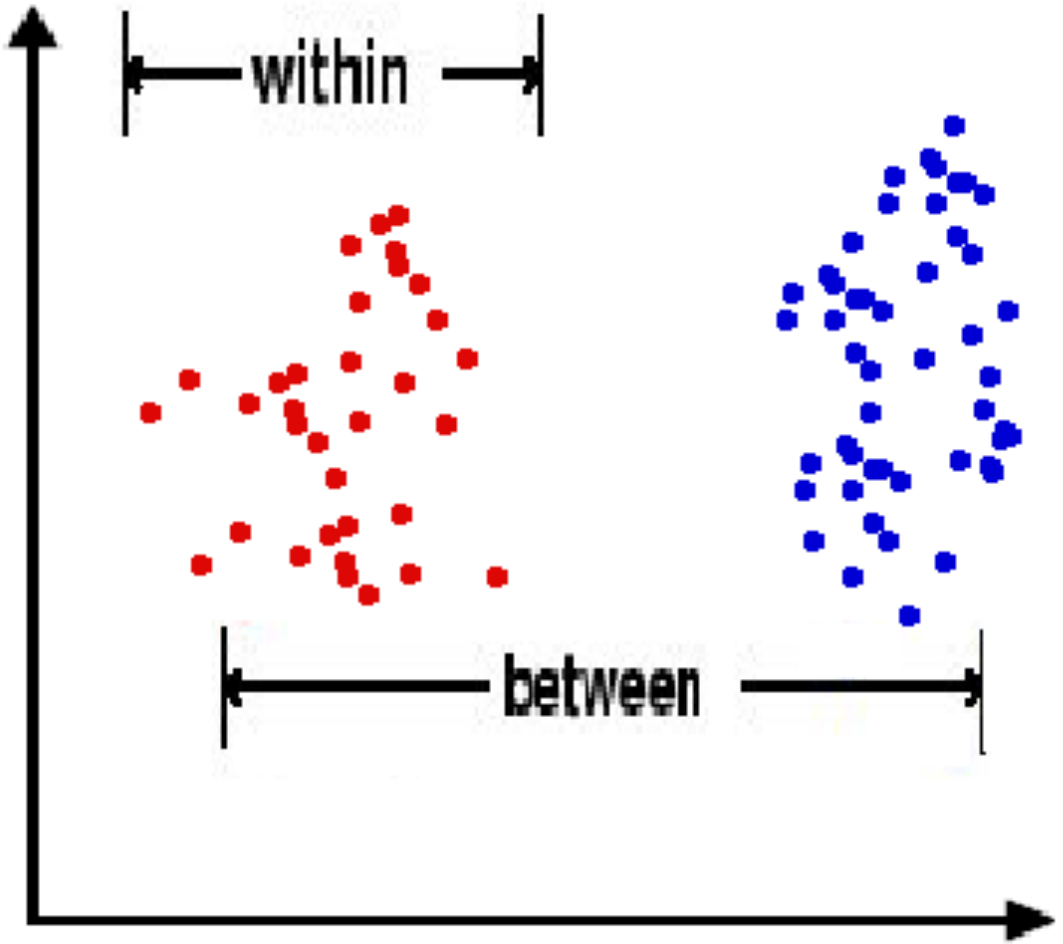




Linear Discriminant Analysis



Linear Discriminant Analysis

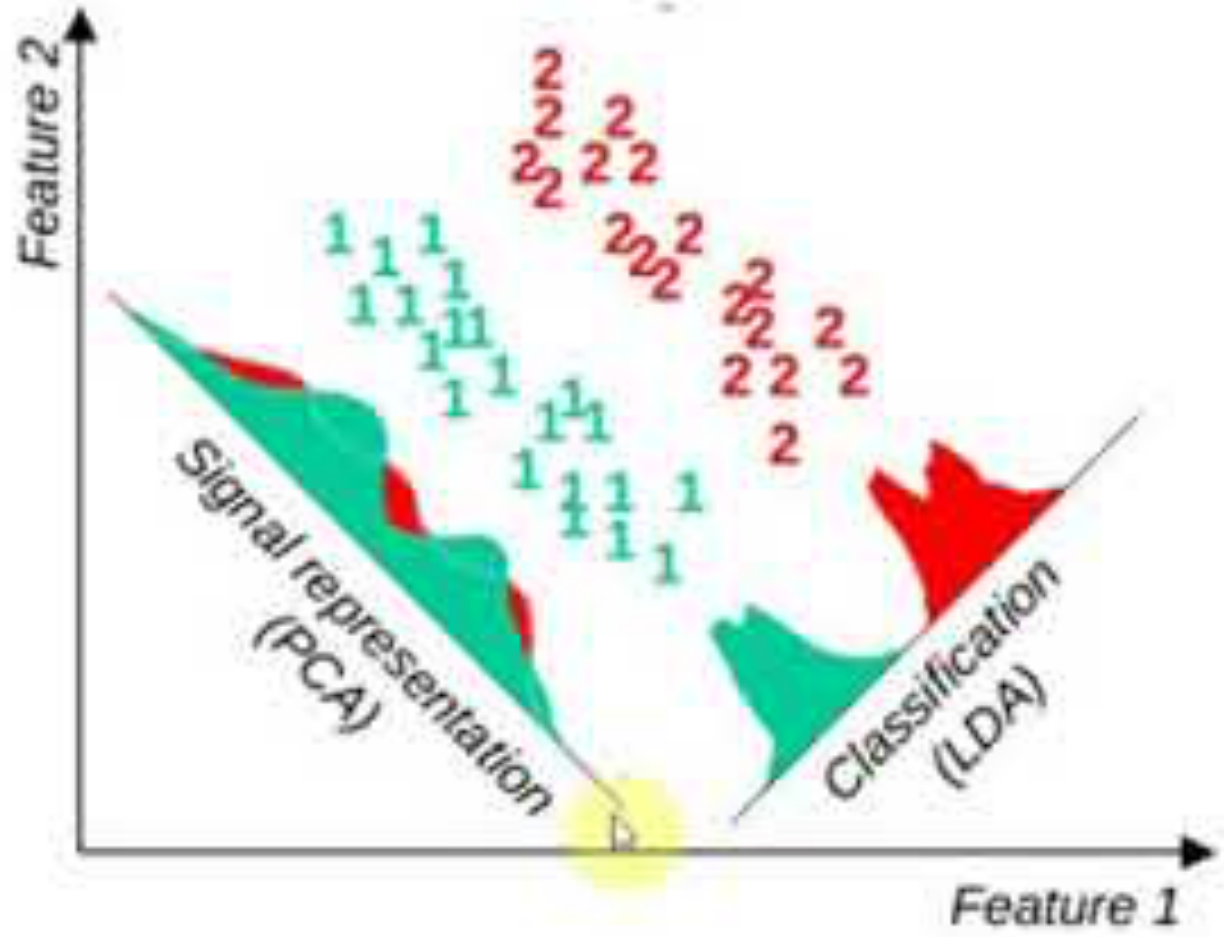


Linear Discriminant Analysis: Quiz

- Question:
  - PCA is about maximizing the variance
  - LDA minimizes variance but maximizes distance between classes
  - In practice, will the LDA and PCA transformations really differ that much?

- Answer: Heck yeah!

Linear Discriminant Analysis: Quiz



http://stackoverflow.com/questions/33576963/dimensions-reduction-in-matlab-using-pca

Linear Discriminant Analysis

- Pros
  - Basic, computationally inexpensive method of deriving a space (and boundary) between classes
  - Complement to PCA
    - PCA characterizes patterns and find important features over entire data set
    - LDA characterizes patterns and finds important features used to differentiate classes

- Cons:
  - “Linear” boundary
    - E.g., quadratic discriminant analysis

Neighborhood Component Analysis

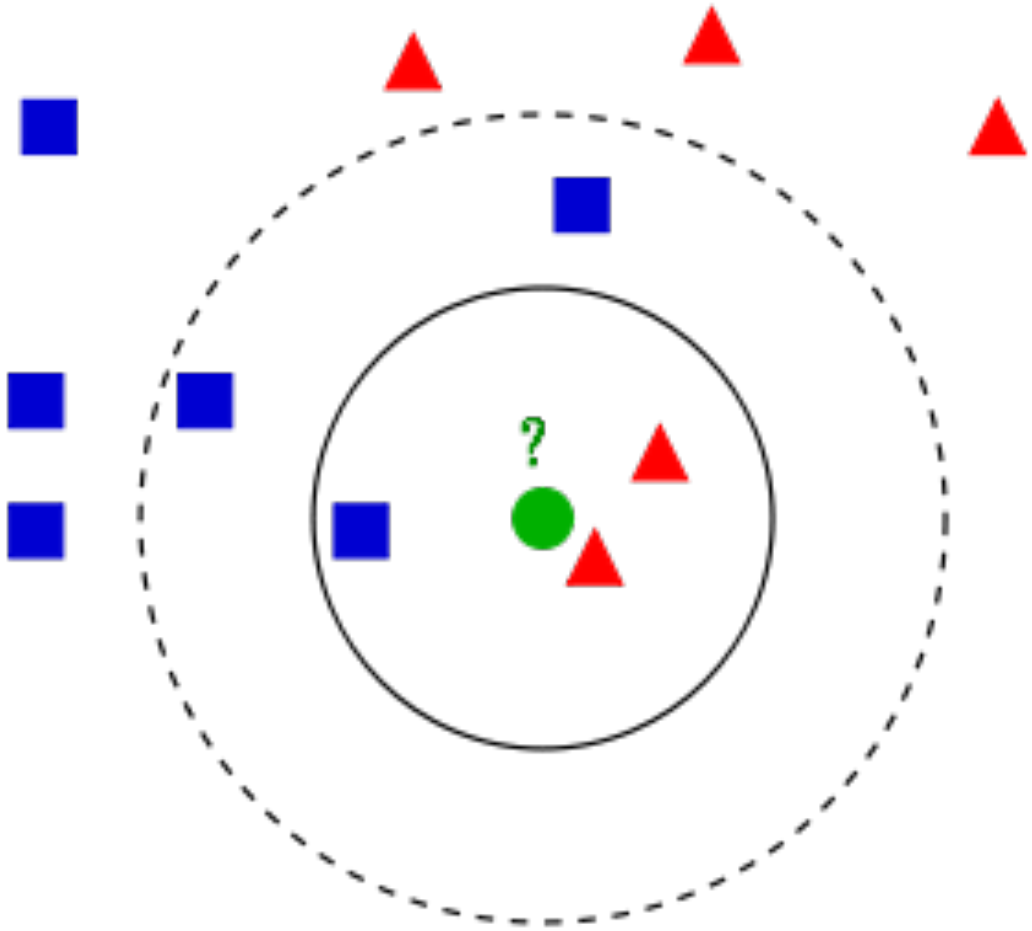
- J. Goldberger, G. Hinton, S. Roweis, R. Salakhutdinov. (2005) *Neighbourhood Components Analysis*. Advances in Neural Information Processing Systems. 17, 513-520, 2005.

- Like k-nearest neighbor algorithm

Neighborhood Component Analysis: k-nearest neighbor algorithm

- Classify a point based on how many of the k nearest points fall into one class or another

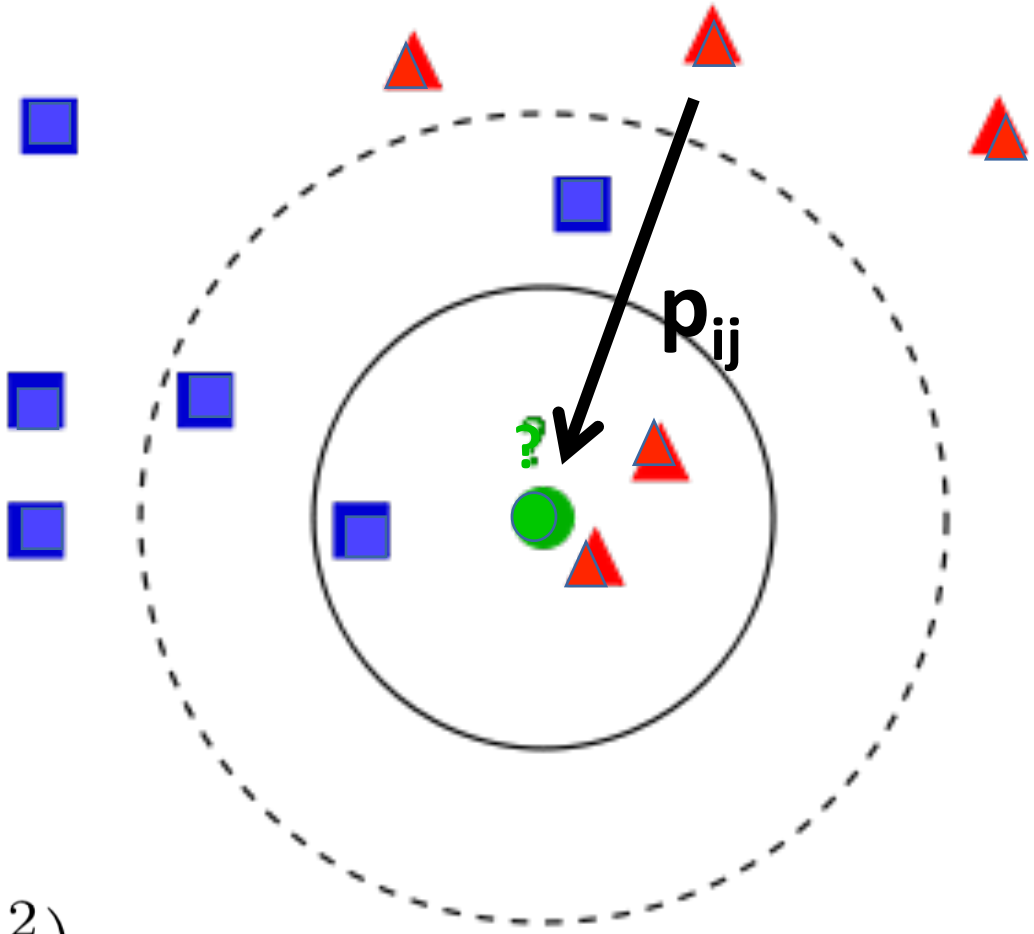
- Typically tune k (e.g., using cross validation)



Neighborhood Component Analysis: The Idea

- Naïve tweak to k-NN: transform your data points into a space that maximizes your classification accuracy

- Less obvious tweak:
  - Forget about k
  - Transform the points in some space
  - Assume the probability that a point is correctly classified based on the label of another point falls off with distance



$$p_{ij} = \frac{\exp(-\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i} \exp(-\|Ax_i - Ax_k\|^2)}$$

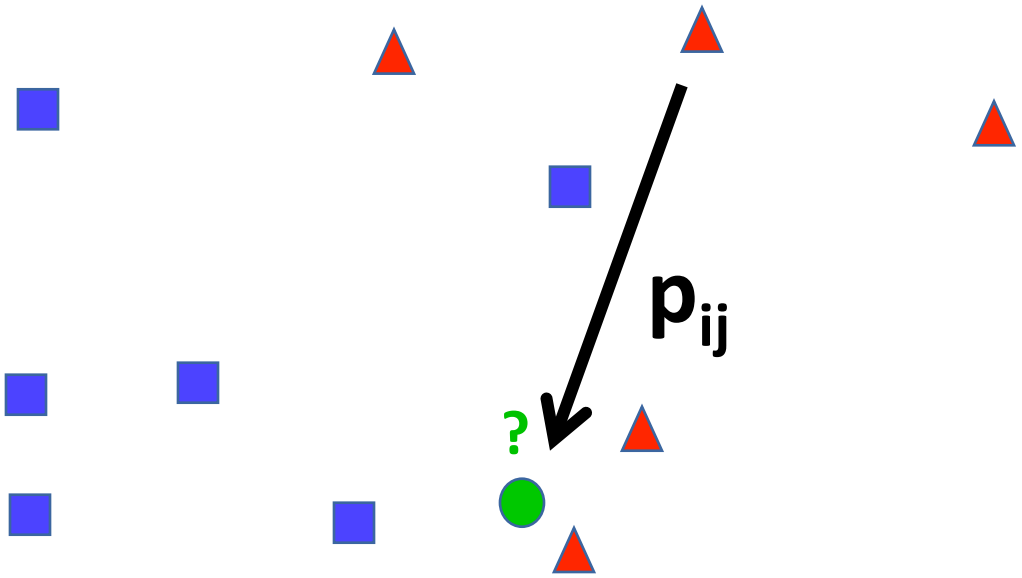
Neighborhood Component Analysis: The Idea

$$p_{ij} = \frac{\exp(-\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i} \exp(-\|Ax_i - Ax_k\|^2)}$$

ith point

jth point

transformation



Neighborhood Component Analysis

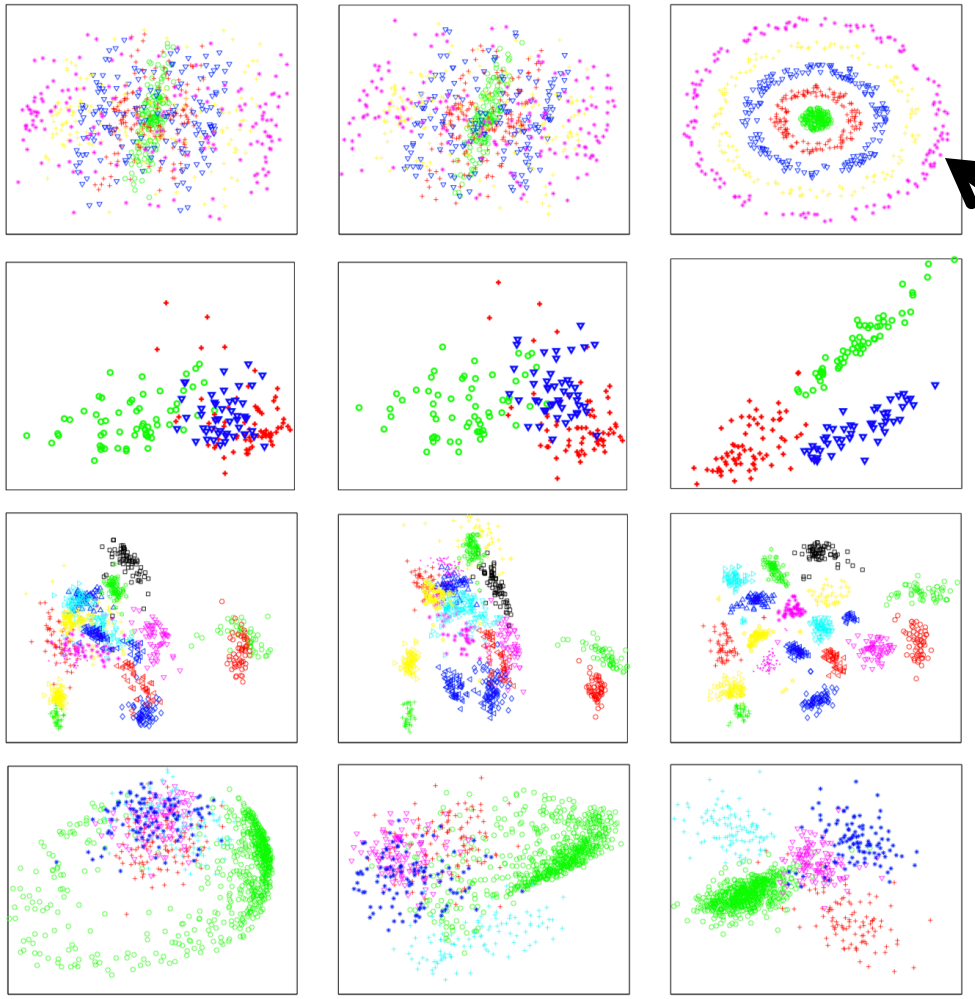


Figure 2: Dataset visualization results of PCA, LDA and NCA applied to (from top) the "concentric rings", "wine", "faces" and "digits" datasets. The data are reduced from their original dimensionalities (D=3,D=13,D=560,D=256 respectively) to the d=2 dimensions show.

Have to think: how often do I run into this problem?

Neighborhood Component Analysis

- Pros:
  - Put you in a space that best differentiates data even when boundaries are highly non-linear
  - I use it to determine if there is any hope in differentiating classes before applying any more robust, cross-validated ML method

- Cons:
  - Very computationally expensive

Neighborhood Component Analysis: Semi-Supervised Classification

- There are cases where you don't entire trust that new data points will exactly match the (simulated) data points you trained on
- Example:
  - I record two people talking
  - I record the frequency spectra of their voices (over many points in time)
  - Based on previous recording of one of those people talking, I want to know who is talking when
  - Problem: previous recording in
    - Different context
    - Different voice dynamics
    - Different noise levels
    - Different devices!

Neighborhood Component Analysis: Example

- I record two people talking
- I record the frequency spectra of their voices (over many points in time)
- Based on previous recording of one of those people talking, I want to know who is talking when

- Want to transform data in a space where you have the best shot of associating the previous recording with the correct speaker

Neighborhood Component Analysis: Strategy

- Transform ***recording with two speakers*** into a space that best differentiates ***two speakers***
- Transform ***previous recording*** into that space
- See which set of points the previous recording overlaps with which speaker

Semi-Supervised Approach for Identifying Speakers Using NCA: The Data

- ~10 minutes of my daughter and I trading places reading Calvin and Hobbes cartoons
- ~10 minutes of me reading footnotes from Tolstoy's Diaries
- Used pyAudioAnalysis package to split the speakers
- Used the extracted acoustic features every 0.2 second over 2 second windows

Feature ID	Feature Name	Description
1	Zero Crossing Rate	The rate of sign-changes of the signal during the duration of a particular frame.
2	Energy	The sum of squares of the signal values, normalized by the respective frame length.
3	Entropy of Energy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.
4	Spectral Centroid	The center of gravity of the spectrum.
5	Spectral Spread	The second central moment of the spectrum.
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames.
7	Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale.
22-33	Chroma Vector	A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
34	Chroma Deviation	The standard deviation of the 12 chroma coefficients.

NCA: Recordings

My Daughter and I

reading

Calvin and Hobbes

Me

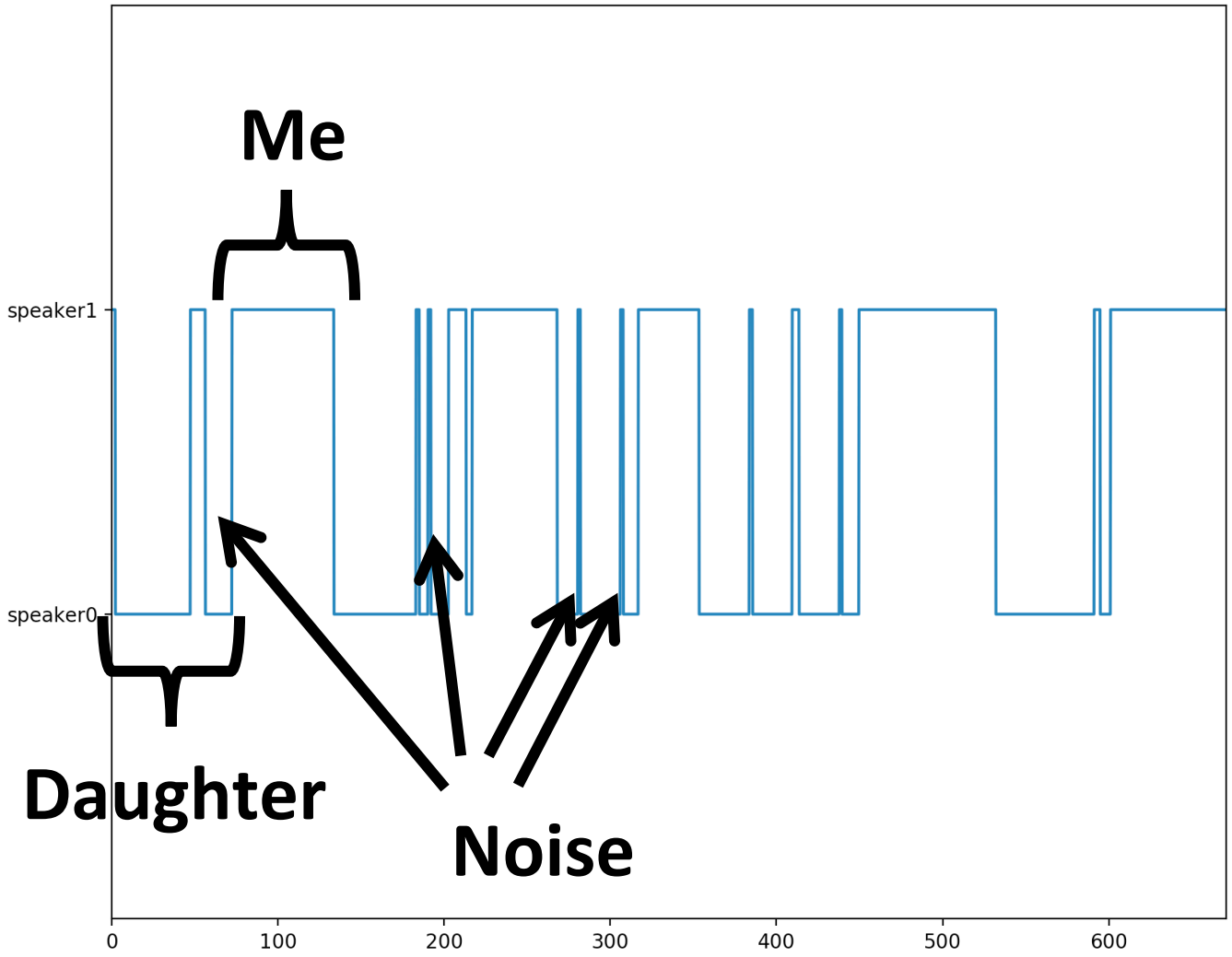
reading

Tolstoy Diaries

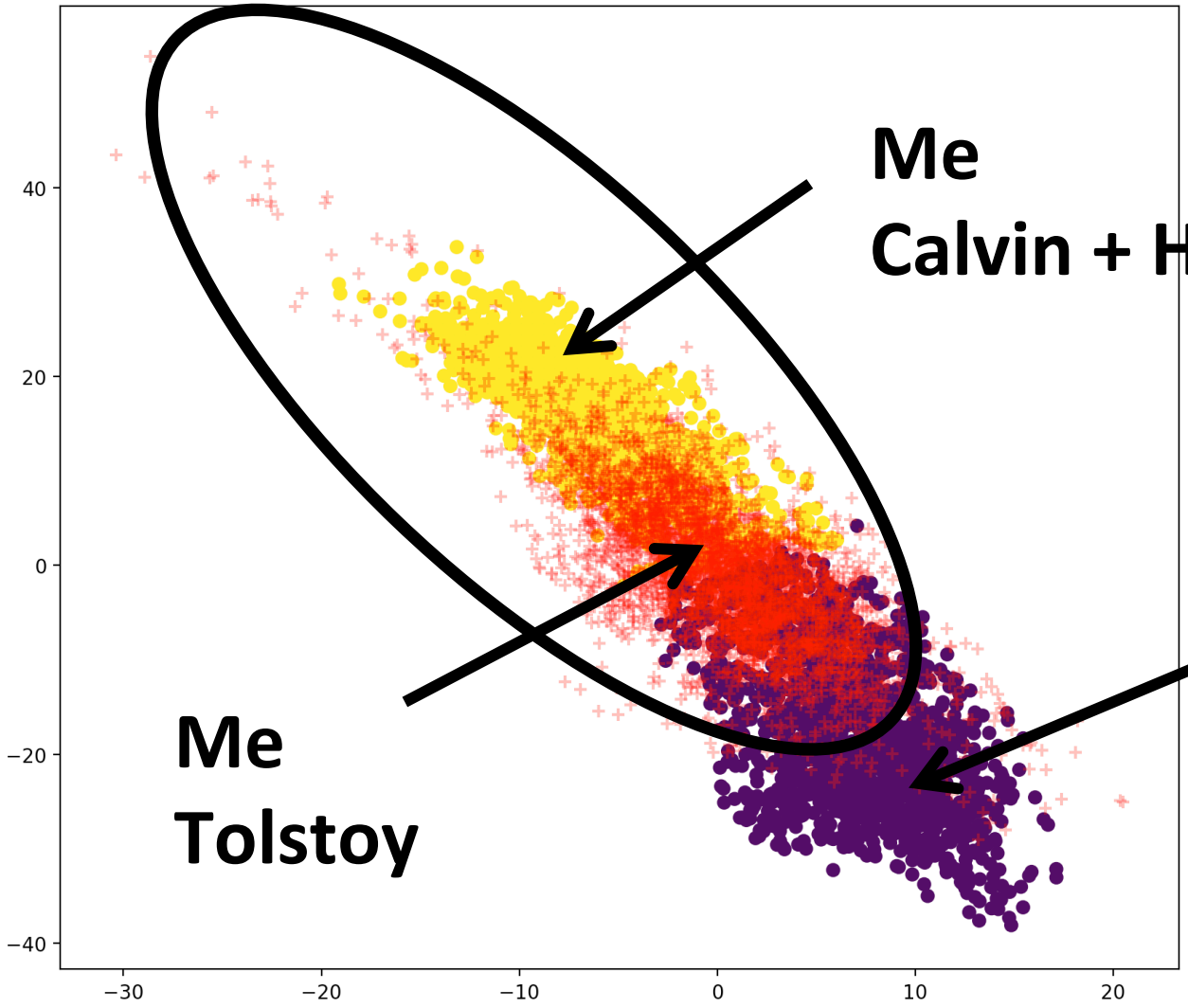
Semi-Supervised Approach for Identifying Speakers Using NCA: Strategy

- Find NCA transformation that best differentiates speaker 1 and speaker 2 from Calvin and Hobbes me+daughter .wav file
- Then transform me reciting Tolstoy into that ‘space’
- See if my acoustic points overlap the most with speaker 1 or speaker 2

Semi-Supervised Approach for Identifying Speakers Using NCA: Speaker Diarization



Semi-Supervised Approach for Identifying Speakers Using NCA: NCA-transformed Data



Me
Calvin + Hobbes

Daughter
Calvin + Hobbes

Me
Tolstoy

NCA Semi-Supervised Classification: Other Uses

- Rejecting background noise in neutrino detectors
  - Have simulation that gives a rough idea of what signal and background noise look like
  - Have a few background events identified from data
  - Strategy:
    - Use NCA to put points in space where you have best shot of discriminating Signal (high energy neutrinos) and radio background
    - Cluster the points (using k-NN, Dirichlet, whatever) assuming two groups
    - Fit each cluster with a density function
    - Figure out which group corresponds to signal and background based on proximity
    - ***Measure the strength of your signal***
  - *In this strategy have let the data tell you almost everything about what signal/background look like*

The ANITA-IV experiment in Antarctica, prior to being launched on a balloon.

Summary

- Discussed unsupervised methods: PCA, MDA, and t-SNE
- Discussed supervised methods: LDA and NCA
- Discussed how one might use a supervised method to perform a semi-supervised classification task
- Only covered a small bit of dimensionality reduction

Things I Didn’t Have Time to Cover

- Singular Value Decomposition
- How word embedding fits into all of this
- Extensions of these techniques (e.g., multi-class LDA, quadratic discriminant analysis, kernel PCA, etc.)
- Compare/contrast these techniques with standard statistical techniques like regression/factor analysis
- Low-Rank Dimensionality Reduction (used for collaborative filtering)
- Autoencoding

Resources

- A Tutorial on PCA, Lindsay I. Smith

- J. Goldberger, G. Hinton, S. Roweis, R. Salakhutdinov. (2005) *Neighbourhood Components Analysis*. Advances in Neural Information Processing Systems. 17, 513-520, 2005.

- Background on LDA: Hand, David J. "Classifier technology and the illusion of progress." *Statistical science* (2006): 1-14.