Spectral Algorithms

Santosh Vempala

Georgia Tech

School of Computer Science

Algorithms and Randomness Center

Thanks to:

Dimitris Bertsimas
Margaret Bjarnodottir
Charlie Brubaker
David Cheng
Amit Deshpande
Petros Drineas
Nick Feamster
Alan Frieze
Ravi Kannan
Christos Papadimitriou
Luis Rademacher
Prabhakar Raghavan
Anirudh Ramachandran
Hisao Tamaki
Adrian Vetta
V. Vinay
Grant Wang

“Spectral Algorithm”??

- Input is a matrix or a tensor
- Algorithm uses singular values/vectors (principal components) of the input.

- Does something interesting!

Applications of Spectral Methods

- Indexing, e.g., LSI
- Embeddings
- Combinatorial optimization
- Learning
- Data mining

A book in preparation (joint with Ravi Kannan):

http://www.cc.gatech.edu/~vempala/spectral.pdf

Networks are matrices

- With entries indicating existence of links
- Or traffic or bandwidth or delay or …

- Tensors (multi-dimensional arrays) also arise naturally
- E.g., Nodes x Nodes x Time, with the (i,j,k)'th entry indicating the traffic between nodes i and j during time interval k

Some questions

How to

- detect anomalous behavior?
- learn network characteristics to use in routing etc.?
- understand the cause(s) of congestion/ failure?

Contents

- SVD basics
- Part I: Learning
- Part II: Clustering
- Part III: Sampling
- Bonus: Exercises!

Principal components

- A: m x n matrix of reals
- Singular value, left/right singular vectors:

$$Av=\sigma u \qquad u^T A= \sigma v^T$$

- u's are orthonormal, v's are orthonormal
- u=v=eigenvector if A is symmetric

Singular Value Decomposition

Real m x n matrix A can be decomposed as:

$$[\,A\,] = [\,U\,][\,D\,][\,V^T\,]$$

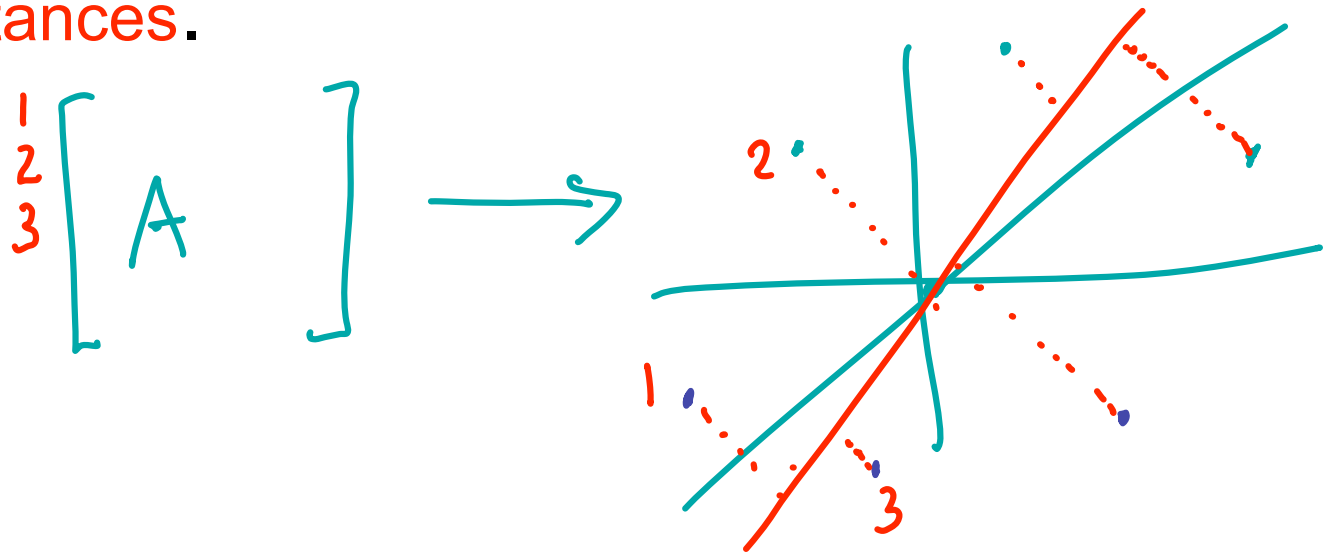
$$A = \sum_{i=1}^{n} \sigma_i \, u_i v_i^T$$

$\{u_i\}$, $\{v_i\}$ ORTHONORMAL

$$\sigma_1 \geq \sigma_2 \geq \cdots \sigma_n \geq 0$$

SVD in geometric terms

Rank-1 approximation is the projection to the line through the origin that minimizes the sum of squared distances.



Rank-k approximation is projection to k-dimensional subspace that minimizes sum of squared distances.

Later: Fast PCA with sampling

[Frieze-Kannan-V. ‘98]
Sample a “constant” number of rows/colums of input matrix.
SVD of sample approximates top components of SVD of full matrix.

[Drineas-F-K-V-Vinay]
[Achlioptas-McSherry]
[D-K-Mahoney]
[Deshpande-Rademacher-V-Wang]
[Har-Peled]
[Arora, Hazan, Kale]
[De-V]
[Sarlos]
…

Fast (nearly linear time) SVD/PCA appears practical for massive data.

Three problems

1. Learn a mixture of unknown Gaussians

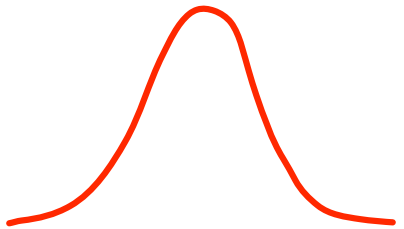
2. Cluster entities (e.g., IP's) using pairwise similarity.

3. Find the "important" subspace of a data set quickly

Problem 1

- Learn a mixture of Gaussians

  Classify samples from



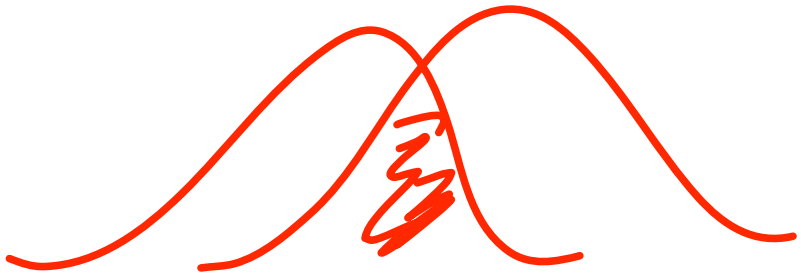
$$F = w_1 F_1 + \ldots + w_k F_k$$

where each F_i is an unknown Gaussian.

Mixture models

- Easy to unravel if components are far enough apart

- Impossible if components are too close



Distance-based classification

How far apart?

Let's look at $E(||X-Y||^2)$ for X,Y from the same component and from different components:

$E(||X-Y||^2) =$

Thus, suffices to have

$||\mu_i-\mu_j||^2 >$

Distance-based classification

[Dasgupta '99] $\|\mu_i - \mu_j\| \geq \ldots \sqrt{n} \ldots$

[Dasgupta, Schulman '00] $n^{1/4}$

[Arora, Kannan '01] (more general) $n^{1/4}$

Spectral Projection

- Project to span of top k principal components of the data

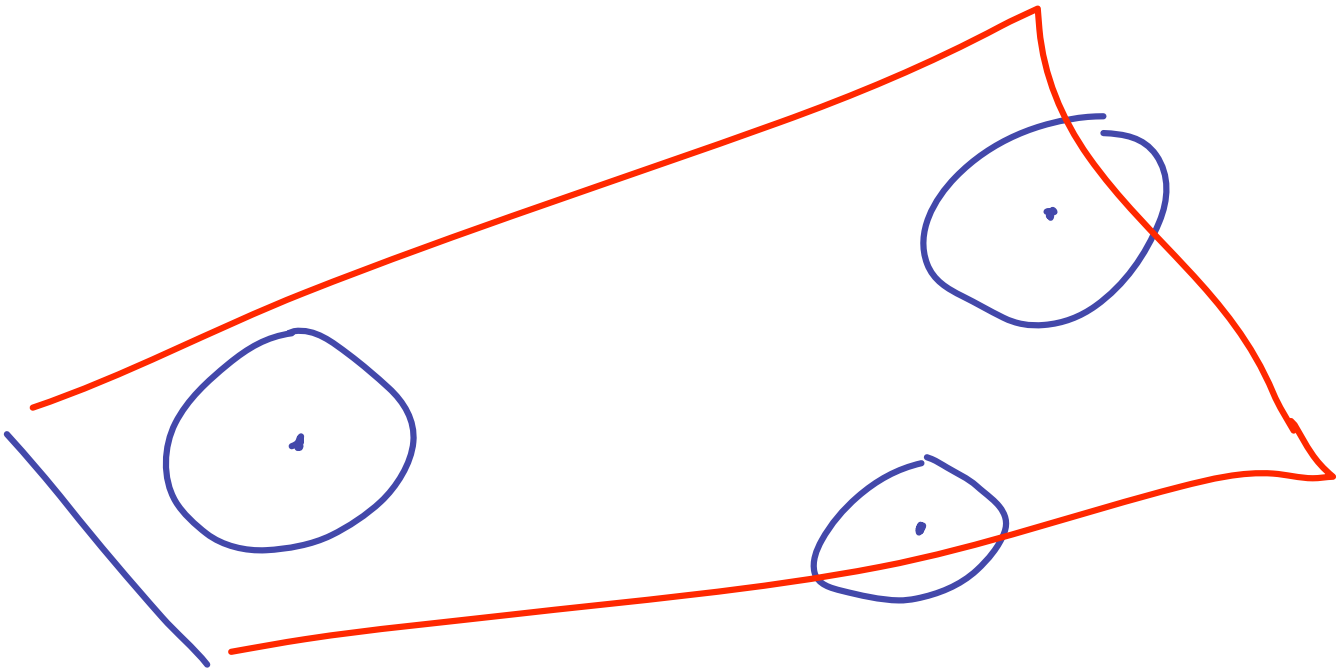
  Replace A with $A_k \approx \sum_{i=1}^{k} \sigma_i u_i v_i^T$

- Apply distance-based classification in this subspace

Main idea

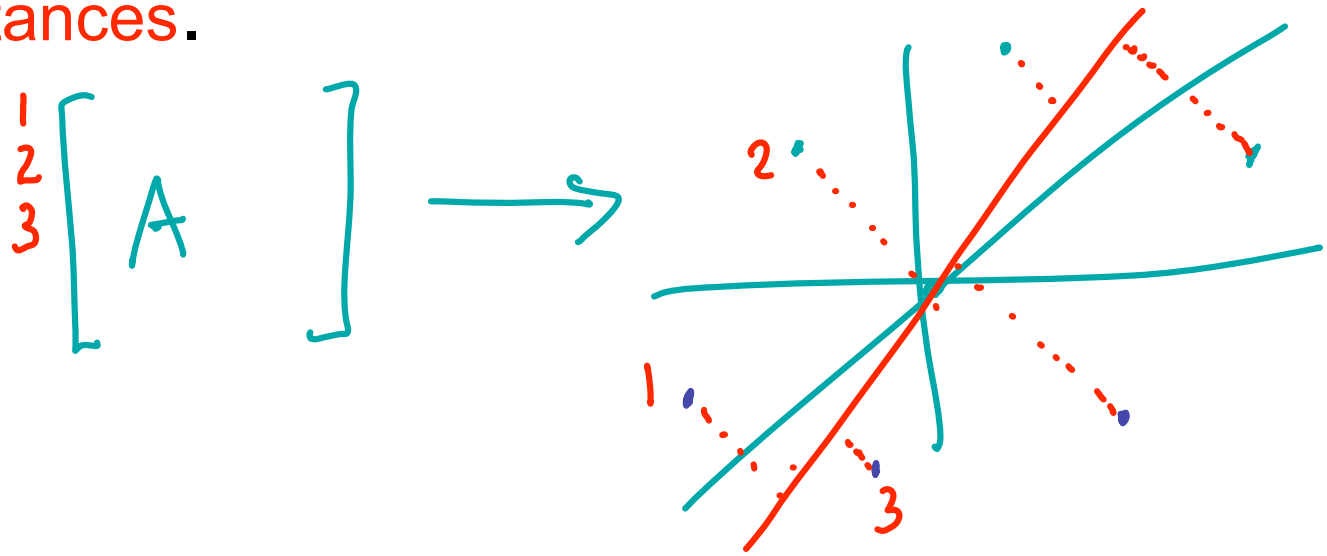
Subspace of top k principal components (SVD subspace)

spans the means of all k Gaussians



SVD in geometric terms

Rank 1 approximation is the projection to the line through the origin that minimizes the sum of squared distances.



Rank k approximation is projection k-dimensional subspace minimizing sum of squared distances.

Exercise 0

- Prove that for a set of points in n-space, the point X in space minimizing the sum of squared distances to the points in the set is their centroid.

Why?

- Best line for 1 Gaussian?
  - Line through the mean

- Best k-subspace for 1 Gaussian?
  - Any k-subspace through the mean

- Best k-subspace for k Gaussians?
  - The k-subspace through all k means!

How general is this?

Theorem[V.-Wang'02]. For any mixture of weakly isotropic distributions, the best k-subspace is the span of the means of the k components.

Covariance matrix = multiple of identity

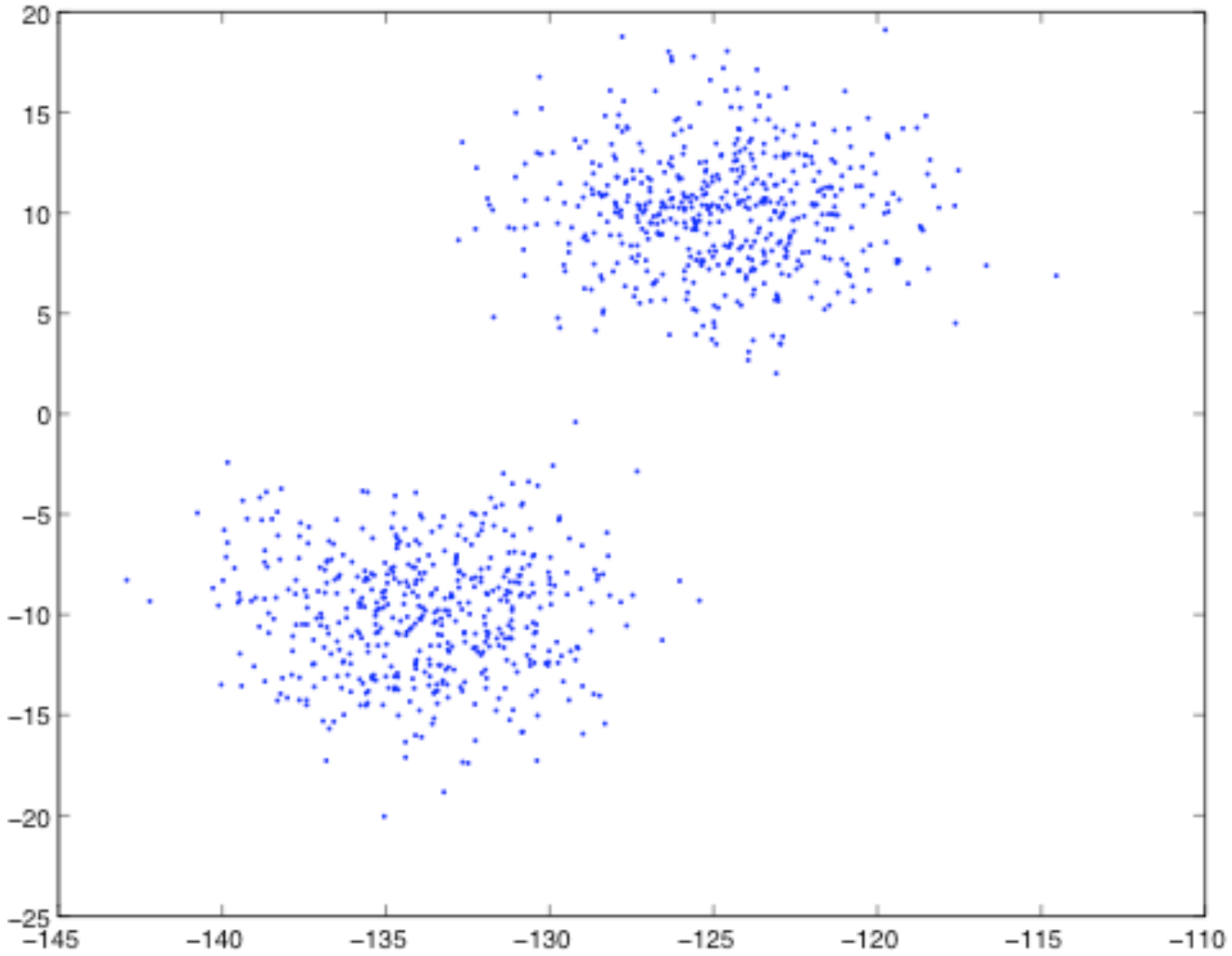
Ex 1: Prove cube is isotropic.

Ex 2: Prove covariance = identity iff variance is 1 in every direction

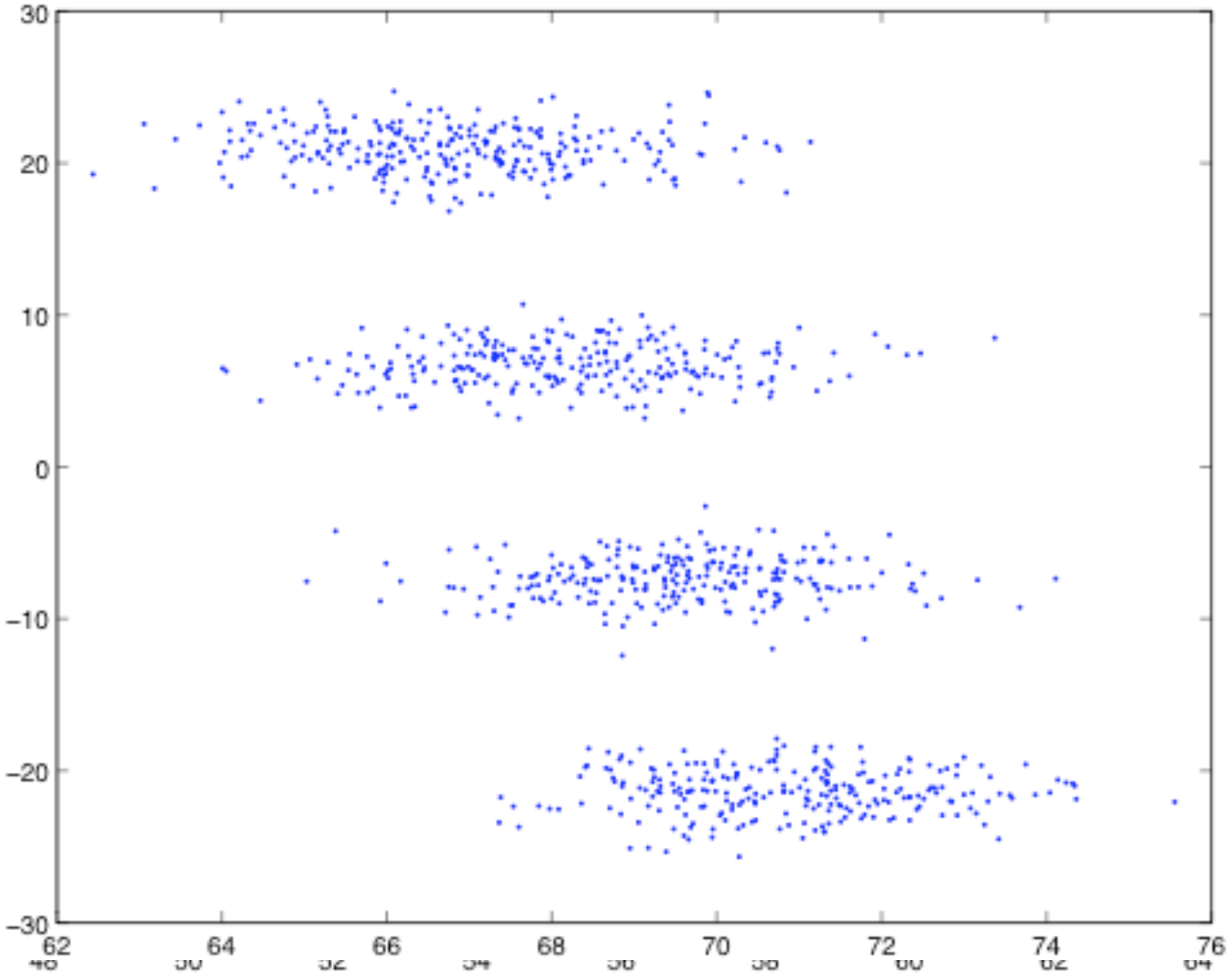
Sample SVD

- Sample SVD subspace is “close” to mixture’s SVD subspace.
- Doesn’t span means but is close to them.

2 Gaussians in 20 Dimensions



4 Gaussians in 49 Dimensions



Mixtures of logconcave Distributions

Theorem [Kannan, Salmasian, V, '04].

For any mixture of k distributions with SVD subspace V,

$$\sum_{i=1}^{k} w_i \, d(\mu_i, V)^2 \leq k \sum_{i=1}^{k} w_i \sigma_i^2$$

σ_i^2 : largest variance of F_i

Questions

1. Can Gaussians separable by hyperplanes be learned in polytime?

2. Can Gaussian mixture densities be learned in polytime?

Separable Gaussians

- PCA fails
- Even for “parallel pancakes”

- Separation condition that specifies distance between means *is not affine-invariant*, i.e., rotation and scaling can change the condition.
- Probabilistic separability is affine-invariant
- So is hyperplane separability.

Fisher criterion

- For a direction p,

$$J(p) = \frac{\text{intra-component variance along p}}{\text{total variance along p}}$$

- Overlap of a 2-component mixture:
  Min J(p) over all directions p.

- Small overlap => large separation along some direction

Fisher subspace for k > 2

- Overlap in a subspace S:

$$J(S) = \frac{\text{intra-component variance within S}}{\text{variance within S}}$$

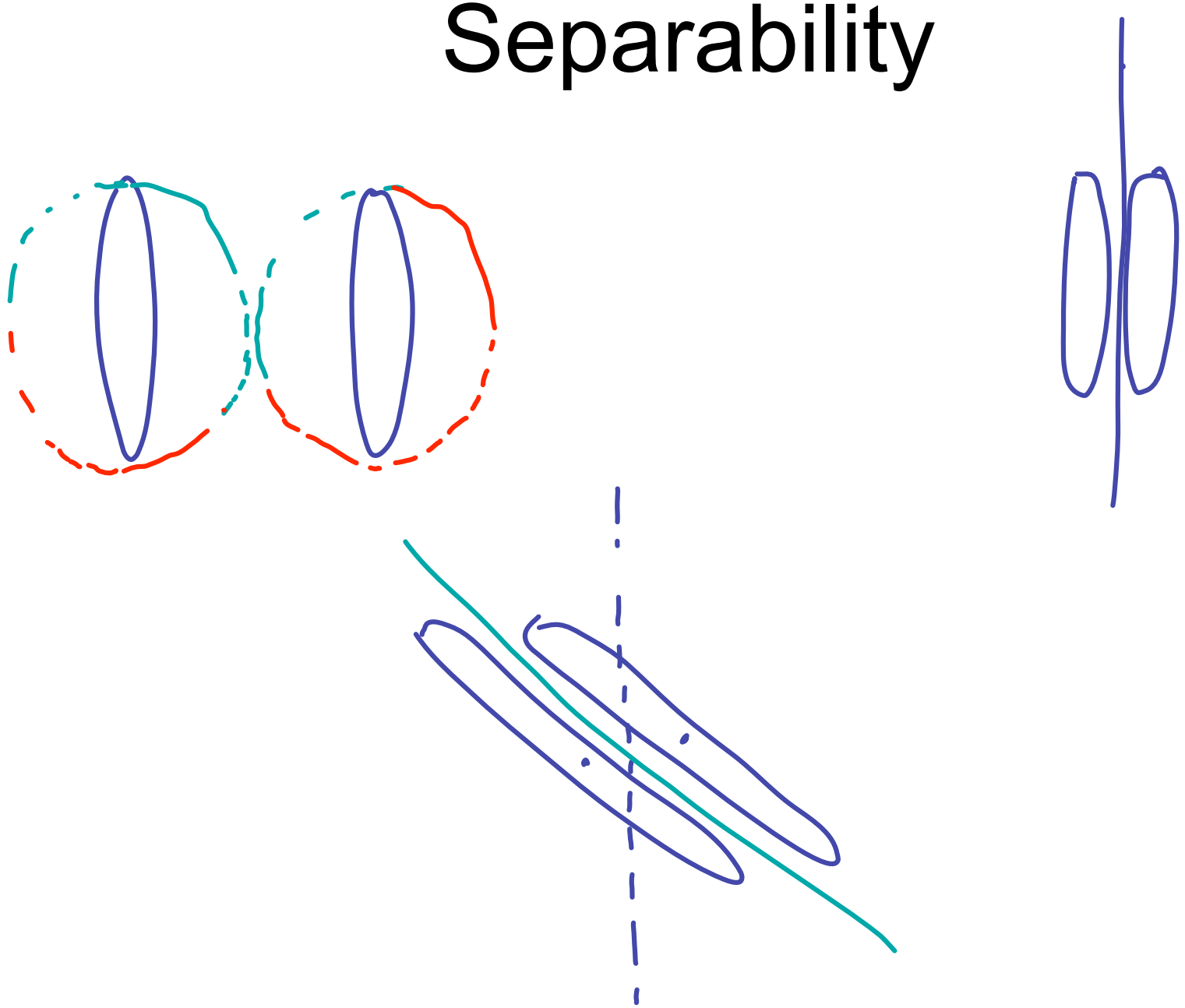
- Overlap for k-component mixture

  Min J(S), S: k-1 dim subspace

  This parameter is affine-invariant. In fact,

  For an isotropic mixture, Fisher subspace is the span of the component means!

Separability



How to find the Fisher subspace?

- Make isotropic: the mean of the mixture the origin and the variance in every direction equal (to 1).
- Moves parallel pancakes apart.

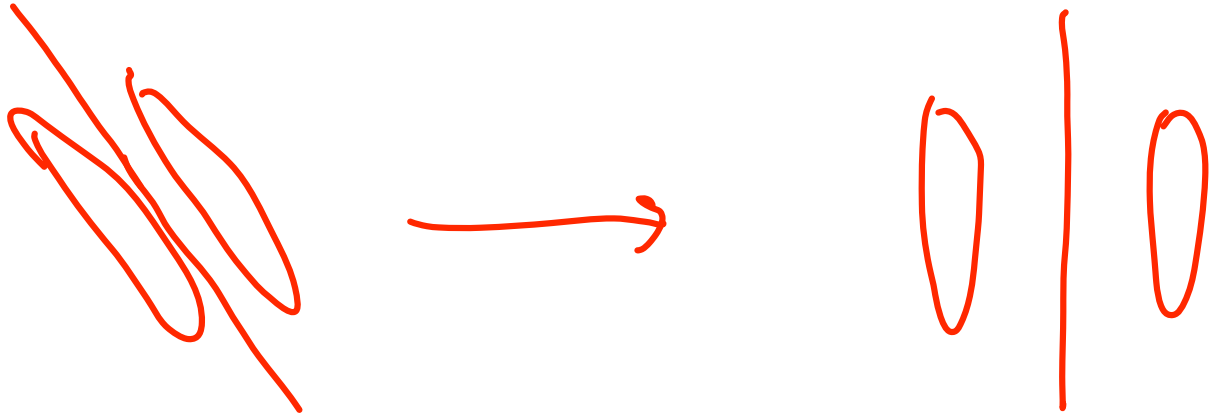
- But, all singular values are equal, so PCA finds nothing!

Idea: Rescale and Reweight

- Apply an isotropic transformation to the mixture.
- Then reweight using the density of a spherical Gaussian centered at zero.
- Now find the top principal component.

Two parallel pancakes

- Isotropy pulls apart the components



- If one is heavier, then overall mean shifts along the separating direction
- If not, principal component is along the separating direction

Unraveling Gaussian Mixtures

Unravel(k)

- Make isotropic
- Reweight
- If mixture mean shifts significantly, use that direction to partition and recurse
- Else use top principal component to partition and recurse.

Unraveling Gaussian Mixtures

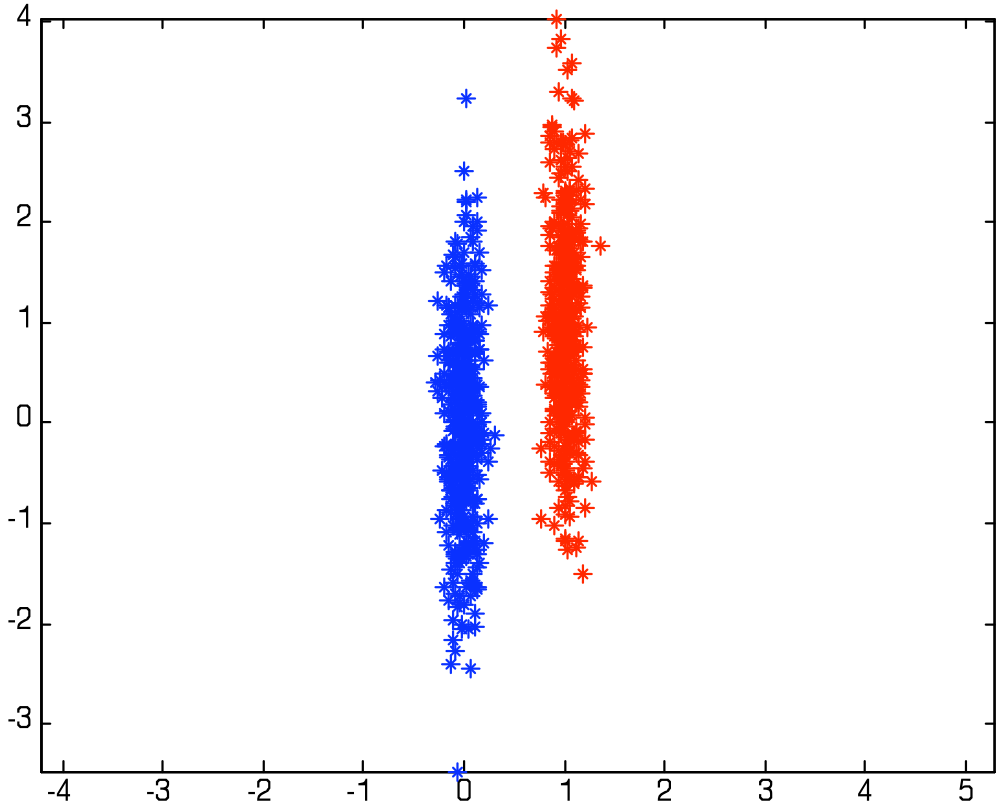
Theorem [Brubaker-V. 08]

The algorithm correctly classifies samples from two arbitrary Gaussians separable by a hyperplane with high probability.

Mixtures of k Gaussians

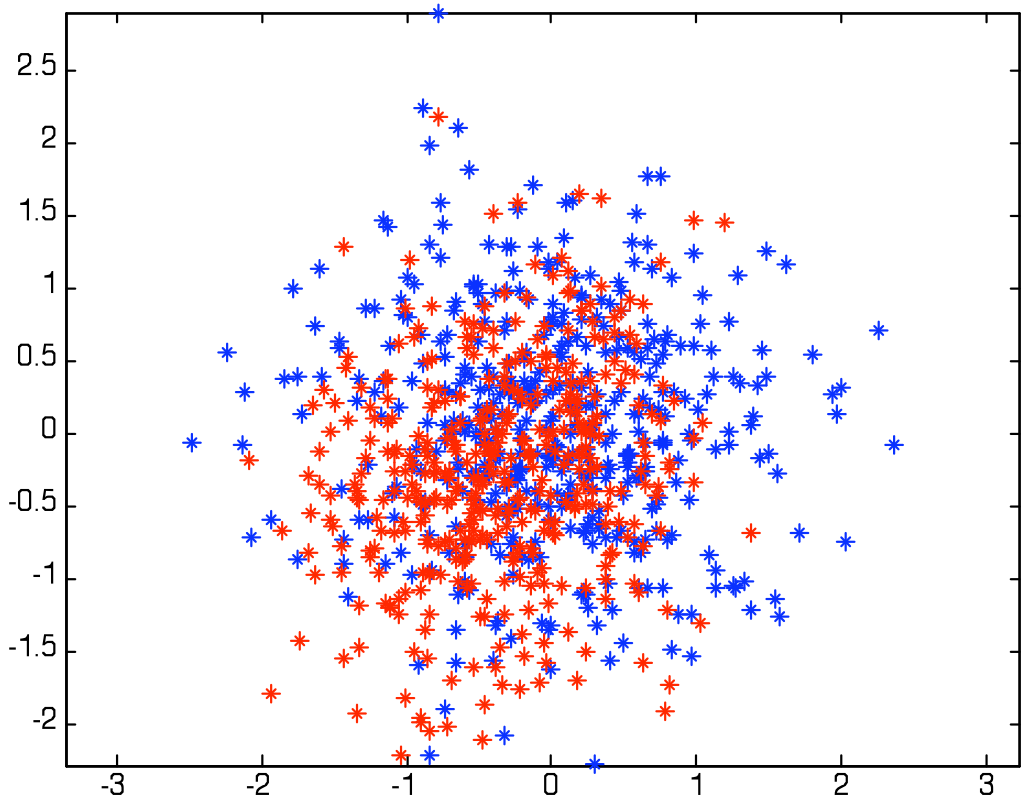
Theorem [B-V 08] For a k-Gaussian mixture with overlap at most $1/k^3$, the algorithm classifies correctly whp using poly(n) samples.

Original Data

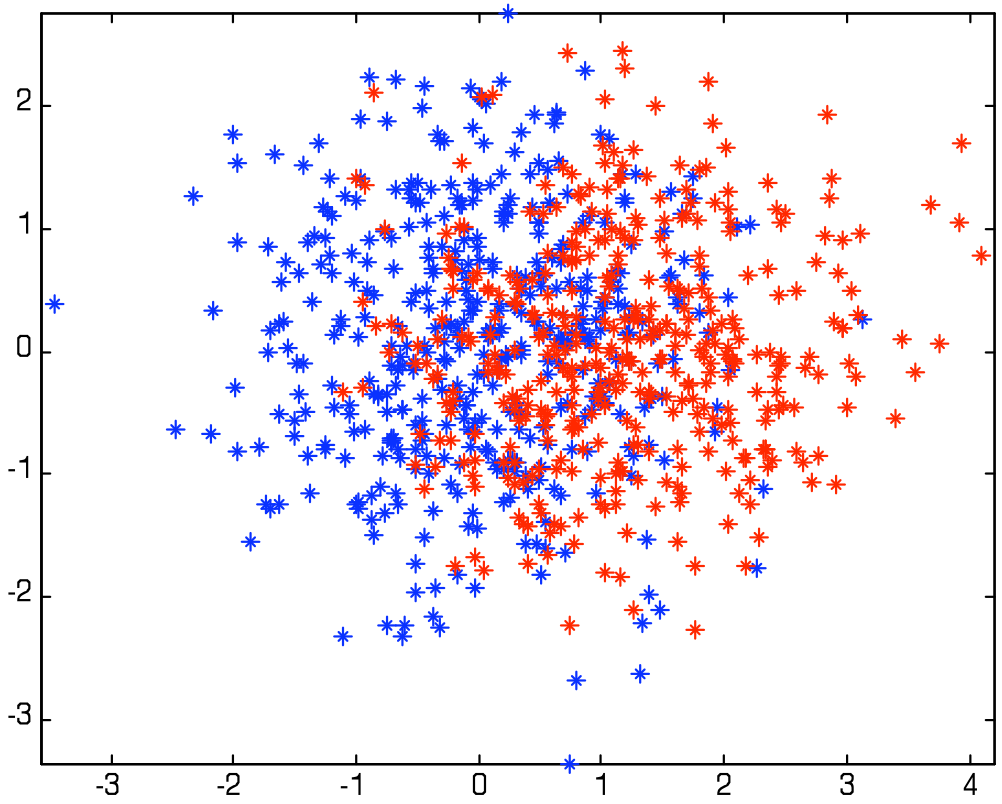


- 40 dimensions, 8000 samples (subsampled for visualization)
- Means of (0,0) and (1,1).

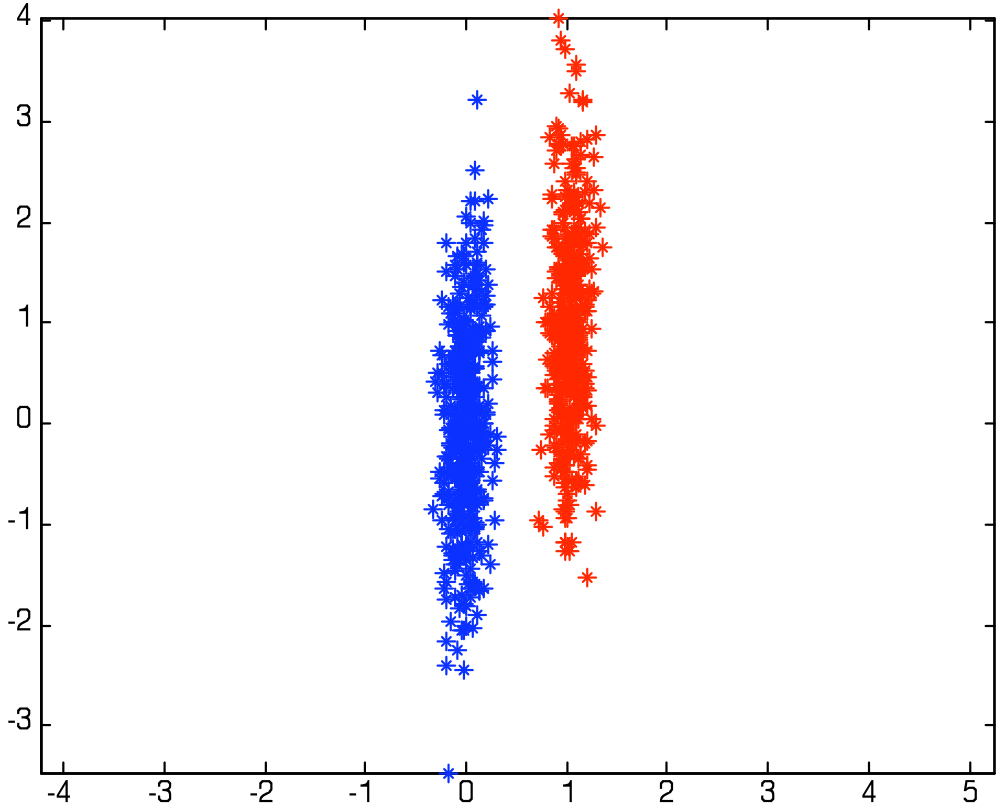
Random Projection



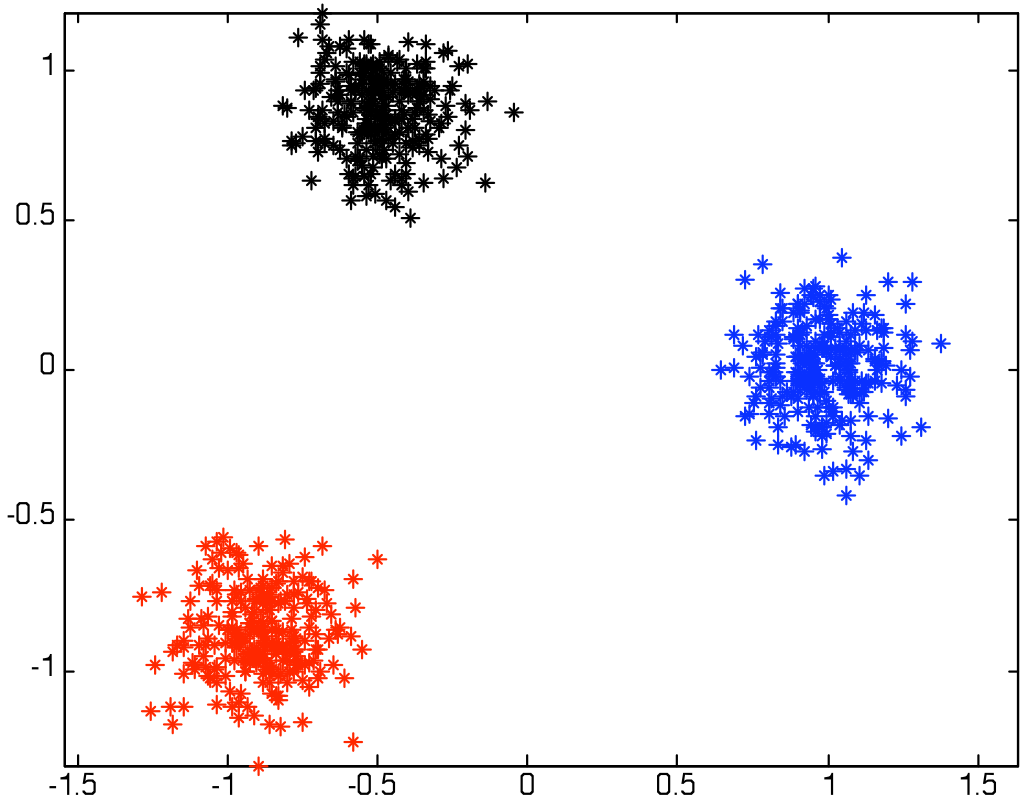
PCA



Isotropic PCA

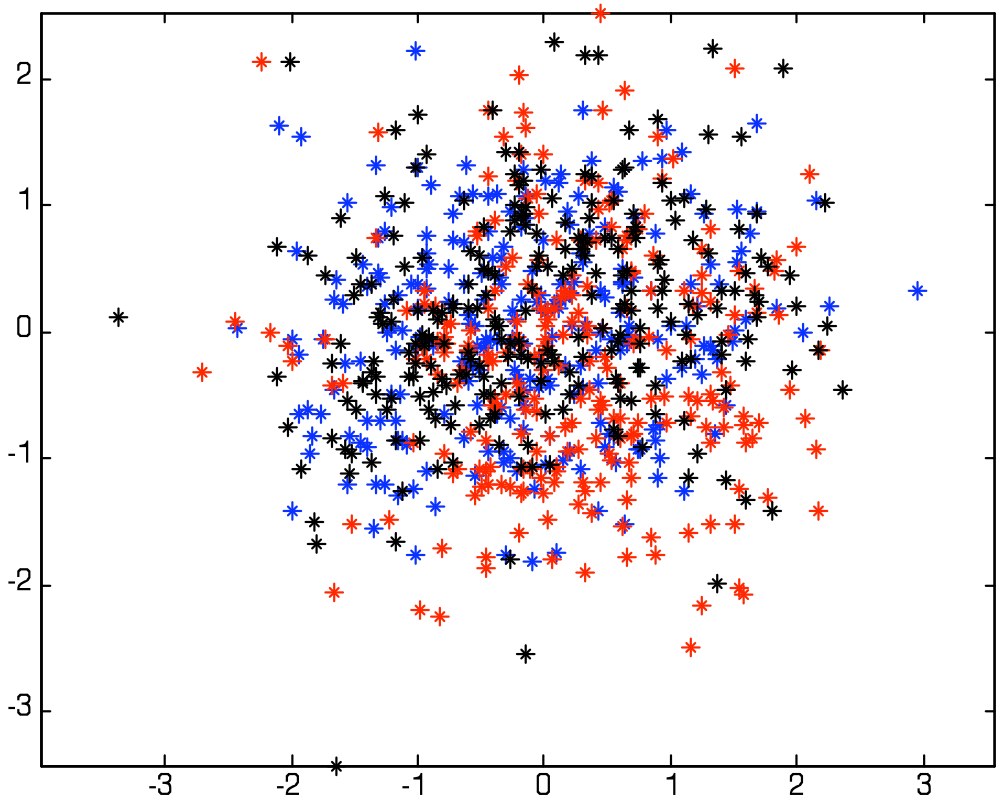


Original Data (k=3)

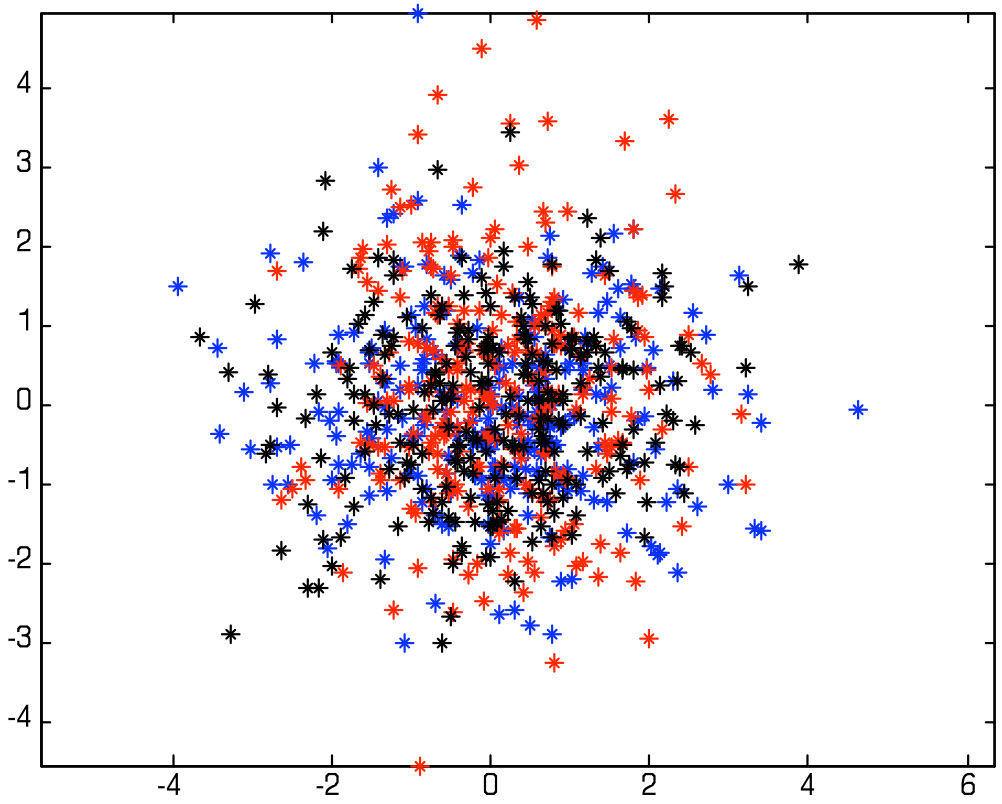


- 40 dimensions, 15000 samples (subsampled for visualization)

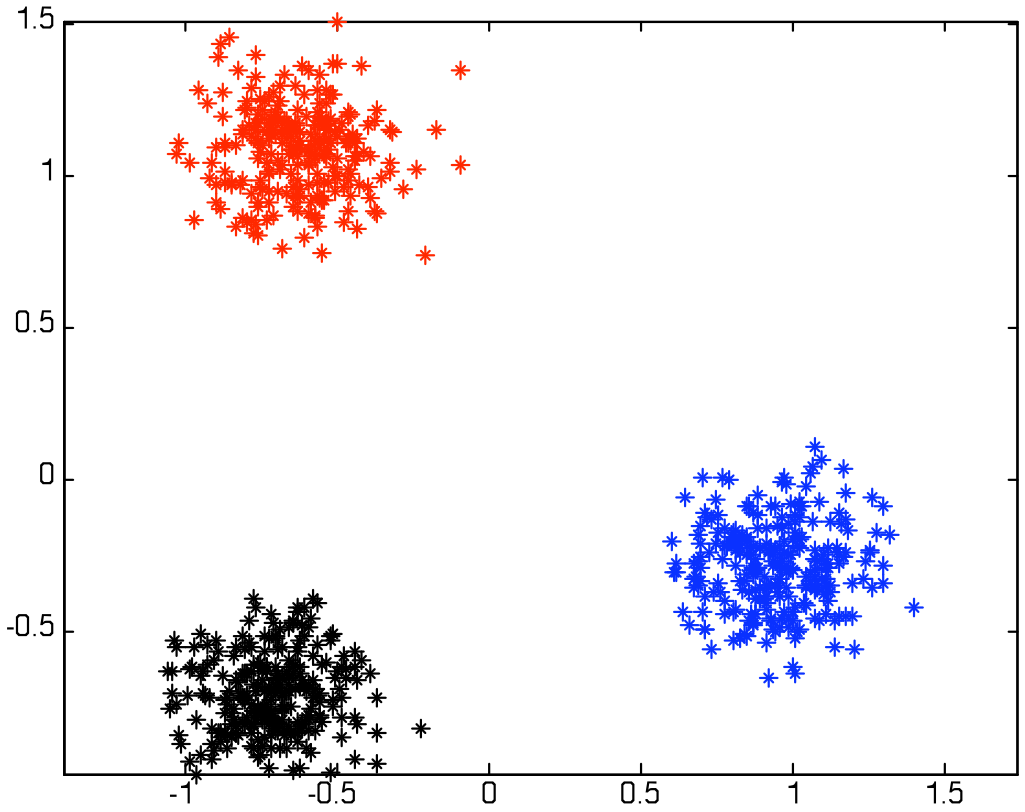
Random Projection



PCA



Isotropic PCA



Questions

- Is it hard to learn the density of a mixture when the components are allowed to overlap significantly?

- Other applications of Iso-PCA? E.g., it can distinguish a cylinder from two parallel disks. What does it do in general?

Part II: Spectral Clustering

- Generic Algorithm:

  -Project to top k principal components

  -Map each point to component to which it has largest projection

- Q. When do singular vectors identify clusters?

Spectral Clustering

- Motivating example: block-diagonal matrix with k blocks, 1 per cluster.
- Top eigenvalue is repeated k times, with 1 eigenvector per cluster, the top eigenvector of each block
- To identify clusters, map each row to the eigenvector to which it has the highest projection
- Works also if A = B+E where B is block-diagonal and E is a perturbation of small norm [PRTV, FKMS]

Spectral Clustering

- Planted clique problem:

  Find a large hidden clique in a random graph

- More generally: planted partitions

  [Bopanna, Alon-Kahale, McSherry, Dasgupta-Hopcroft-Kannan-Mitra, Kannan]

Planted clique

- A: adjacency matrix of a random graph, with a planted clique of size k.
- Reporting highest degree vertices works when k > ??
- Reporting largest component vertices of the top eigenvector works for k > ??
- Finding smaller planted cliques is a major open problem

Clustering from pairwise similarities

Input:

A set of objects and a (possibly implicit) function on pairs of objects.

Output:

1. A flat clustering, i.e., a partition of the set
2. A hierarchical clustering
3. (A weighted list of features for each cluster)

Typical approach

Optimize a “natural” objective function
E.g., k-means, min-sum, min-diameter etc.

Using EM/local search (widely used) OR
a provable approximation algorithm

Issues: quality, efficiency, validity.
Reasonable functions are NP-hard to optimize

Divide and Merge

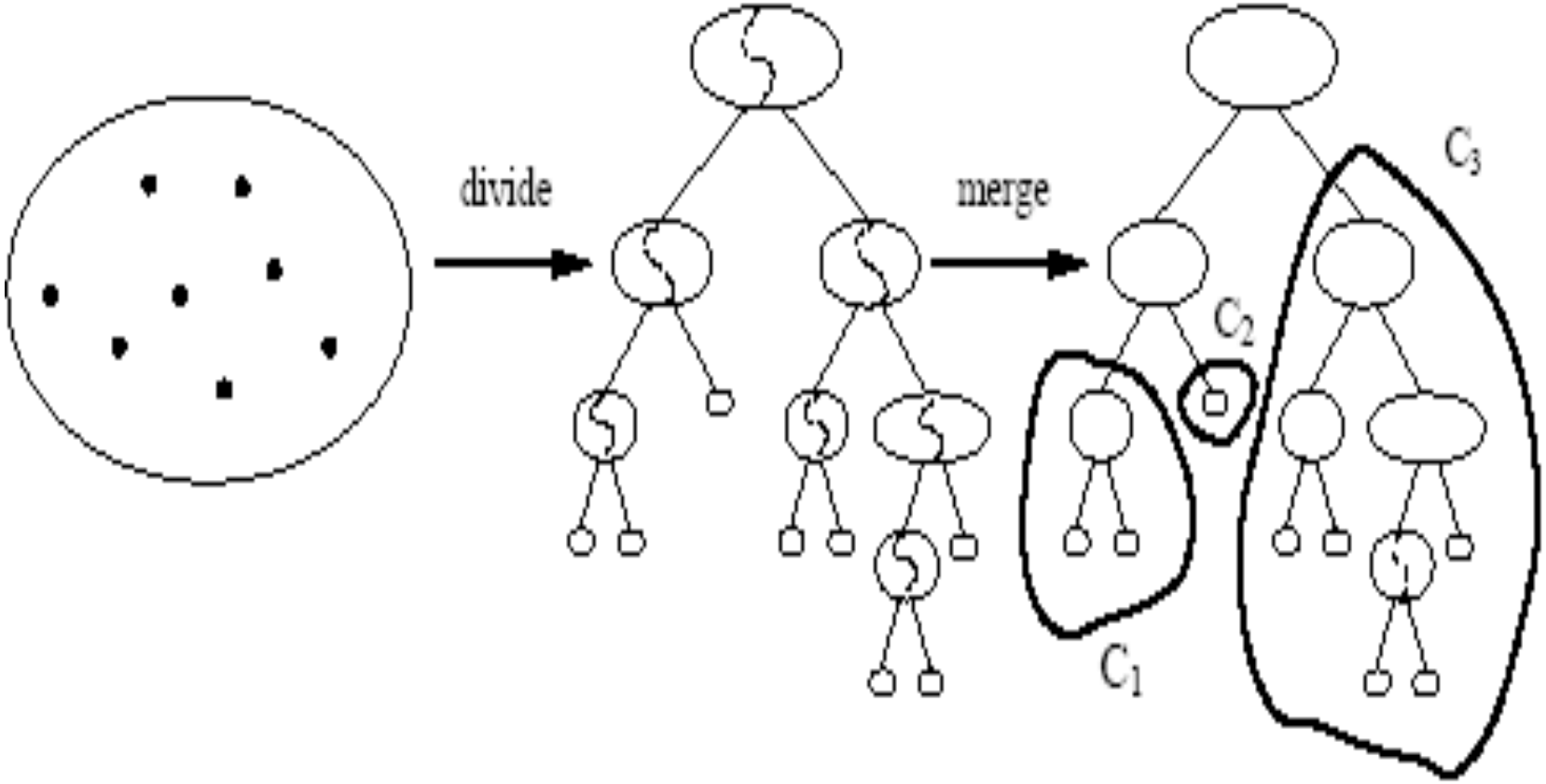
- Recursively partition the graph induced by the pairwise function to obtain a tree

- Find an “optimal” tree-respecting clustering

Rationale: Easier to optimize over trees;

k-means, k-median, correlation clustering all solvable quickly with dynamic programming

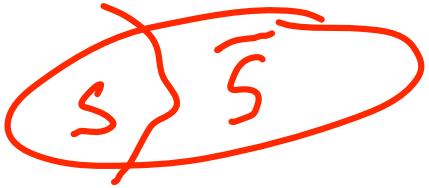
Divide and Merge



How to cut?

Min cut? (in weighted similarity graph)

Min conductance cut [Jerrum-Sinclair]



$$\phi(S) = \frac{w(S, \bar{S})}{\min\{w(S), w(\bar{S})\}}$$

Sparsest cut [Alon]

Normalized cut [Shi-Malik]

Many applications: analysis of Markov chains, pseudorandom generators, error-correcting codes...

How to cut?

Min conductance/expansion is NP-hard to compute.

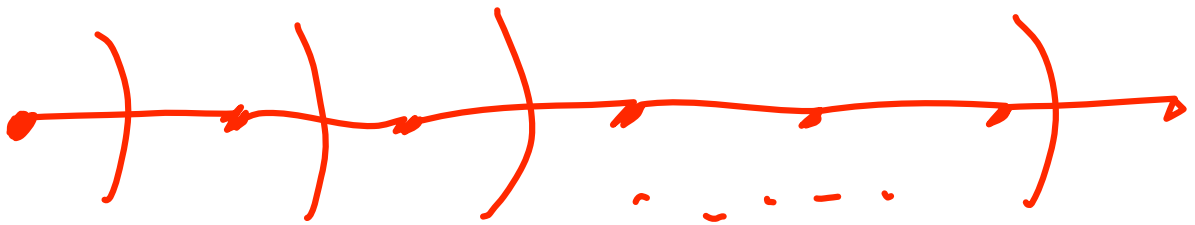
- Leighton-Rao
  Arora-Rao-Vazirani

$O(\log n)$

$O(\sqrt{\log n})$

- Fiedler cut: Minimum of n-1 cuts when vertices are arranged according to component in 2nd largest eigenvector of similarity matrix.

$O(\sqrt{OPT})$



Divide phase

- Normalize similarity matrix
- Order according to components in second eigenvector
- Choose best of the n-1 cuts in this ordering
- Recurse on the two parts

Merge phase

- Clusters are subtrees
- Use dynamic programming to find optimal tree-respecting clustering for specified objective function

- Ex. 1: Maximize min conductance while keeping number of clusters at most k.
- Ex. 2: Minimize fraction of inter-cluster edges while keeping min conductance at least α.

Worst-case guarantees

- Suppose we can find a cut of conductance at most $a.C^{\nu}$ where C is the minimum.

Theorem [Kannan-V.-Vetta '00].

If there exists an (α, ε)-clustering, then the algorithm is guaranteed to find a clustering of quality $\left(\frac{\alpha^{1/\nu}}{a \log n}, a\varepsilon^{\nu} \log n \right)$

Experimental evaluation

- Evaluation on data sets where true clusters are known
  - Reuters, 20 newsgroups, KDD UCI data, etc.
  - Test how well algorithm does in recovering true clusters – look an entropy of clusters found with respect to true labels.

- Question 1: Is the tree any good?

- Question 2: How does the best partition (that matches true clusters) compare to one that optimizes some objective function?

data set	Spectral	p-QR	p-Kmeans	K-means
alt.atheism/comp.graphics	93.6 ± 2.6	89.3 ± 7.5	89.6 ± 6.9	76.3 ± 13.1
comp.graphics/comp.os.ms-windows.misc	81.9 ± 6.3	62.4 ± 8.4	63.8 ± 8.7	61.6 ± 8.0
rec.autos/rec.motorcycles	80.3 ± 8.4	75.9 ± 8.9	77.6 ± 9.0	65.7 ± 9.3
rec.sport.baseball/rec.sport.hockey	70.1 ± 8.9	73.3 ± 9.1	74.9 ± 8.9	62.0 ± 8.6
alt.atheism/sci.space	94.3 ± 4.6	73.7 ± 9.1	74.9 ± 8.9	62.0 ± 8.6
talk.politics.mideast/talk.politics.misc	69.3 ± 11.8	63.9 ± 6.1	64.0 ± 7.2	64.9 ± 8.5

Table 1: 20 newsgroups data set (Accuracy)

data set	Spectral	BEX02	LA99	NJM01
8,654 articles	.713	.57	.63	N/A
6,575 articles	.733	N/A	N/A	.665

Table 2: Reuters data set (F-measure)

data set	Spectral	Dhillon 2001
MedCran	.032	.026
MedCisi	.092	.152
CisiCran	.045	.046
Classic3	.090	.089

(a) SMART data set (Entropy)

data set	Spectral	B97
J1	.77	.69
J2	.81	1.12
J3	.54	.85
J4	1.12	1.10
J5	.81	.74
J6	.81	.83
J7	.63	.90
J8	.84	.96
J9	.65	1.07
J10	1.77	1.17
J11	.90	1.05

(b) Webpage data set (Entropy)

Table 3: SMART and Webpage data sets

Clustering medical records

Medical records: patient records (> 1 million) with symptoms, procedures & drugs

Goals: predict cost/risk, discover relationships between different conditions, flag at-risk patients etc.. [Bertsimas-Bjarnodottir-Kryder-Pandey-V.-Wang]

Clusters for “diabetes”:

Cluster 44: [938]
64.82%: Antidiabetic Agents, Misc..
51.49%: Ace Inhibitors & Comb..
49.25%: Sulfonylureas.
48.40%: Antihyperlipidemic Drugs.
36.35%: Blood Glucose Test Supplies.
23.24%: Non-Steroid/Anti-Inflam. Agent.
22.60%: Beta Blockers & Comb..
20.90%: Calcium Channel Blockers&Comb..
19.40%: Insulins.
17.91%: Antidepressants.

Clustering medical records

Cluster 97: [111]
100.00%: Mental Health/Substance Abuse.
58.56%: Depression.
46.85%: X-ray.
36.04%: Neurotic and Personality Disorders.
32.43%: Year 3 cost - year 2 cost.
28.83%: Antidepressants.
21.62%: Durable Medical Equipment.
21.62%: Psychoses.
14.41%: Subsequent Hospital Care.
8.11%: Tranquilizers/Antipsychotics.

Cluster 48: [39]
94.87%: Cardiography - includes stress testing.
69.23%: Nuclear Medicine.
66.67%: CAD.
61.54%: Chest Pain.
48.72%: Cardiology - Ultrasound/Doppler.
41.03%: X-ray.
35.90%: Other Diag Radiology.
28.21%: Cardiac Cath Procedures
25.64%: Abnormal Lab and Radiology.
20.51%: Dysrhythmias.

Other domains

Clustering genes of different species to discover orthologs – genes performing similar tasks across species.

Eigencluster to cluster search results

Compare to Google

[Cheng, Kannan,Vempala,Wang]

Behavioral blacklisting of IP's with SpamTracker

[Ramachandran, Feamster, Vempala]

Exercise 3

1. Find data set on the web

2. Apply eigencluster, choose α (http://arc2.cc.gatech.edu)

3. Interpret results

What next?

- Move away from explicit objective functions? E.g., feedback models, similarity functions

  [Balcan-Blum-Vempala]

- Efficient regularity-style quasi-random clustering: partition into a small number of pieces so that edges within a piece or between pieces appear random.

- Tensors: using relationships of small subsets; Tensor PCA?

  [Frieze-Kannan, FKKV]

- ?!

Part III: Sampling

- Approximate data in a low-dimensional space
- Preserve “important” structure

- Important = ?
- Pairwise distances?
- Hidden model (relevant subspace)?

An illustrative problem

- Suppose data is close to a line or a plane.
- How to tell by sampling only a few points?

Sampling method I: Uniform sampling

- Pick rows (or entries) uniformly at random. Deduce approximation to full matrix from these samples

Matrix-vector product

$$Av = \sum_{j=1}^{n} A^{(j)} v_j$$

Idea: sample terms of this summation

Q. From what distribution?

$p_1, p_2 \dots p_n \qquad \sum p_j = 1$

Matrix-vector product

$$Av = \sum_{j=1}^{n} A^{(j)} v_j$$

$$X = \frac{A^{(j)} v_j}{p_j} \quad \text{w.p.} \quad p_j$$

$$E(X) = Av, \qquad E\left(\|X - Av\|^2\right) \le \|A\|_F^2 \|v\|^2$$

Matrix-vector product

What is optimal sampling distribution?
i.e., one that minimizes Var(X).

Sampling 2: Length-squared sampling

- Pick rows (or entries) with probability proportional to their squared length.

- Exercise 4: Prove that length-squared sampling is optimal.
- Exercise 5: Given N numbers arriving in arbitrary order, show how to pick one from the LS distribution, while using only O(1) memory.

Matrix product

$$AB = \left(AB^{(1)}, AB^{(2)}, \ldots, AB^{(p)}\right)$$

$m \times n$
$n \times p$

Pick index j with probability p_j

$$Y = \frac{1}{s} \sum_{t=1}^{s} \frac{A^{(j_t)} B_{(j_t)}}{p_{j_t}}$$

$$E(Y) = AB, \qquad \mathrm{Var}(Y) \le \frac{1}{s} \|A\|_F^2 \|B\|_F^2$$

Low-rank approximation

Algorithm: Fast-SVD

1. Sample s columns of A from the length-squared distribution to form a matrix C.

2. Find $u_1 \ldots u_k$ the top k left singular vectors of C.

3. Output $\sum_{j=1}^{k} u_j u_j^T A$

as a rank-k approximation to A.

Low-rank approximation

Theorem [FKV, DFKVV].

Let $\tilde{A}$ be the approximation found by algorithm Fast-SVD. Then,

$$E\left(\|A-\tilde{A}\|_F^2\right) \leq \|A-A_k\|_F^2 + \frac{2k}{s}\|A\|_F^2$$

$$\leq OPT + \varepsilon\|A\|_F^2$$

$$\left(\text{for } s=\frac{k}{4\varepsilon^2}\right)$$

Low-rank approximation

- [FKV]: Any matrix contains a constant-sized submatrix from which a nearly optimal low-rank approximation can be constructed.

- Error is additive and cannot be avoided for a fixed sampling scheme.
- Q. What about relative error?

Sampling 3: Volume sampling

- Pick subset of rows with probability proportional to the squared volume of the simplex they induce with the origin.

Sampling 4: Isotropic subspace

- Isotropic RP: Pick random vectors from a Gaussian with the same covariance matrix as the data. Project to the span of the sample.

Spectral Algorithms

- Insightful
- Fast
- Widely applicable

- Notes: http://www.cc.gatech.edu/~vempala/spectral.pdf
- Please send feedback!