GEOMETRIC ASPECTS OF OPTIMIZATION AND APPLICATIONS TO SPECTRAL CLUSTERING

IPAM 2016

Mikhail Belkin, Ohio State University,
Computer Science and Engineering,
Department of Statistics

Joint work with Luis Rademacher and James Voss

The Spectral Theorem and the Power method

- A is a symmetric matrix.

$$F(u) = (Au \cdot u)$$

Theorem: there exists orthogonal basis $e_1, \dots, e_m$, such that

$$F(u) = \sum \lambda_i (u \cdot e_i)^2$$

How to find e_i, given access to F?

The Power Method: e_i are fixed points of the dynamical system on the sphere

$$u \to \frac{Au}{||Au||}$$

This talk: what, why and how

What: an algorithmic primitive, hidden basis recovery.

- Why: examples.
  - PCA, ICA, tensor decomposition, GMM learning, multi-way spectral clustering.
- How: gradient iteration algorithm. Dynamical system on the sphere.
  - Generalization of the power method for matrices/tensors.
- Analysis:
  - "Hidden convexity".
  - Perturbation analysis, generalization of Davis-Kahan theorem for matrices.
  - Fast convergence in clean and noisy settings.
- Applications:
  - ICA
  - Spectral clustering

Hidden Basis Recovery

- Orthonormal basis: $e_1, \dots, e_m$ [partial basis ok]
- Basis Encoding Function (BEF):

$$F(u) = \sum_{i=1}^{m} g_i(u \cdot e_i)$$

- Problem: given evaluation access to F and ∇F, recover e_i.

The Spectral Theorem

- $$F(u) = \langle u, Au \rangle$$

A is a symmetric matrix.

$$F(u) = \sum_i \lambda_i \, (u \cdot e_i)^2 = \sum_i g_i(u \cdot e_i)$$

with $g_i(t) = \lambda_i t^2$.

Eigenvlaues λ_i, *eigenvectors* e_i

Example: tensor decomposition

Orthogonal tensor decomposition (odeco tensors):
Given $T = T_{jlmt} = \sum_i w_i \, e_i \otimes \; e_i \otimes e \otimes e_i$, Basis
Encoding Function is

$$F(u) = T(u,u,u,u) = \sum_i w_i \, (u \cdot e_i)^4 = \sum_i g_i(u \cdot e_i)$$

with $g_i(t) = w_i t^4$.

E.g., [Anandkumar, Ge, Hsu, Kakade, Telgarsky *2013] for model recovery with tensors and using the tensor power method.*

Example: Independent component Analysis

- Independent Component Analysis
  Given samples from x given by $x = As$, with
  - x, s d-dim. random vectors,
  - s with independent coordinates,
  - A square invertible matrix.

  Goal: Recover A.
- After whitening/isotropy, can assume A is orthogonal.
- BEF: $F(u) = \kappa_4(u \cdot x) = \sum_i \kappa_4(s_i)(u \cdot A_i)^4$ with $g_i(t) = \kappa_4(s_i)t^4$.
  (κ_4 is the fourth cumulant, here $\kappa_4(T) = E(T^4) - 3$)

Example: Gaussian Mixture Models

- Parameter estimation for spherical Gaussian mixture model (cf. [Hsu Kakade 2012]).

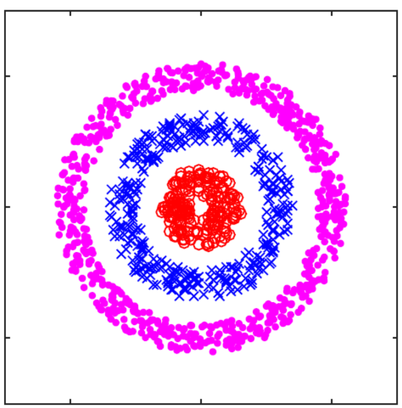
Directional third moment for a mixture can be rewritten in terms of a basis encoding function.

Spectral clustering

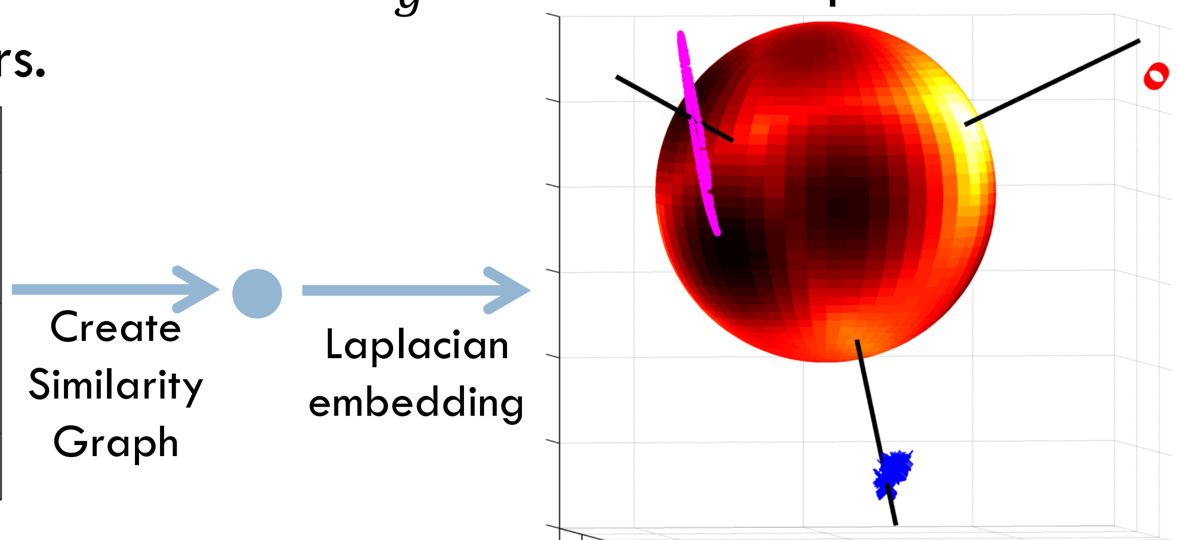
- We maximize an admissible contrast g over **directional projections** of the **embedded** data

$$F_g(\mathbf{u}) = \frac{1}{n}\sum_{i=1}^{n} g(|\langle \mathbf{u}, \mathbf{x}_i \rangle|)$$

- Idea: The **local maxima** of F_g on $\mathbb{S}^{k-1}$ correspond to the desired clusters.



Original Data



Heat map of F_g evaluations on $\mathbb{S}^{k-1}$

Recovering the basis: “gradient iteration” algorithm

$$F(u) = \sum_{i=1}^{m} g_i(u \cdot e_i)$$

- “Gradient Iteration”: a fixed point iteration of the gradient:

$$u \rightarrow \frac{\nabla F(u)}{\|\nabla F(u)\|}$$

Repeat until convergence.

Generalization of the power method for matrices and tensors.

Gradient iteration

- "Gradient Iteration" is an extension of tensor power iteration to a functional setting without multi-linear algebra:

- For example: $F(u) = T(u,u,u,u)$, then tensor power iteration is $u \to \frac{T(u,u,u,\cdot)}{\|T(u,u,u,\cdot)\|}$
  Gradient iteration is $\mathrm{u} \to \frac{\nabla F(u)}{\|\nabla F(u)\|}$
  with $\nabla F(u) = c\, T(u,u,u,\cdot)$

Gradient iteration

$$F(u) = \sum_{i=1}^{m} g_i(u \cdot e_i)$$

- $h_i = g_i\left(\sqrt{|t|}\right)$
- "Gradient Iteration": [suppressing some signs]

Analog to λ_i

$$\mathrm{u} \rightarrow \nabla F(u) = 2\sum_{i=1}^{m} h_i'\left((u \cdot e_i)^2\right)(\mathrm{u} \cdot e_i)e_i$$

- compare to Power Iteration: $\mathrm{u} \rightarrow 2\sum \lambda_i(\mathrm{u} \cdot e_i)e_i$

Conditions on g_i

- "Contrast functions" g_i are either odd or even.
- $\pm g_i(\sqrt{x})$ is strictly convex on $[0,1]$
- $\frac{d}{dx}(g_i(\sqrt{x}))\Big|_{0+} = 0$

[All previous examples except PCA satisfy these]

Under the assumptions on contrasts g_i:

Thm 1 [Optimization point of view]: The set of $\{\pm e_i\}$, the hidden basis vectors, are the only local extrema of F on the *sph*

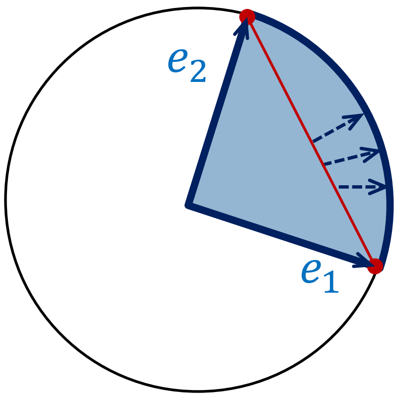
Lots of other critical (saddle) points.

Hidden convexity

Choose coordinates corresponding to the hidden basis e_i, $u = \sum x_i e_i$.

τ: $(x_1, \ldots, x_m) \rightarrow (\sqrt{x_1}, \ldots, \sqrt{x_m})$

sphere → (hidden) simplex



If $g_i(\sqrt{x})$ are convex,

$\mathrm{G}(u) = \sum_{i=1}^{m} g_i\left(\sqrt{\langle u, e_i \rangle}\right)$ is convex on the simplex (sum of convex functions).

Max of F over sphere ⇔ Max G over simplex.

Local maxima of convex functions are at extreme points, that is, e_i.

Maxima, not more usual minima.

Finding the hidden basis

Under the assumptions on contrasts g_i:

- Thm 2 [dynamical systems point of view]:

The set of stable fixed points of gradient iteration is exactly $\{\pm e_i\}$. Other fixed points (exponentially many) are unstable (hyperbolic).

- Thm 3: Gradient iteration will converge to a local extremum almost everywhere.

- Thm 4 [super-linear convergence]: If $g_i(\sqrt[r]{t})$ are convex, then convergence of gradient iteration is of order $r-1$.

Perturbation Analysis: model

Additive noise model for F.

$$\hat{F} = F + \mathrm{E} = \sum_{i=1}^{m} g_i(u \cdot z_i) + E$$

Control up to second derivative $\left\|\nabla\,(\mathrm{F} - \hat{F})\right\|_\infty + \left\|\mathcal{H}(\mathrm{F} - \hat{F})\right\|_\infty < \epsilon$

Need to quantify convexity of $g_i(\sqrt{x})$ on [0,1]:

$$\beta x^{\delta-1} \le (g_i(\sqrt{x}))'' \le \alpha x^{\gamma-1}, \qquad \alpha, \beta, \delta, \gamma > 0$$

E.g., $g_i(x) = x^{2+0.01}$ works.

Perturbation Analysis

$\beta x^{\delta-1} \le (g_i(\sqrt{x}))'' \le \alpha x^{\gamma-1}, \qquad \alpha, \beta, \delta, \gamma > 0$

Sufficiently small perturbation size ϵ.

More general perturbation model.

Thm 5: "Gradient iteration" recovers $e_1, \ldots, e_m$ up to error $4\sqrt{2}\delta m^{\delta}\epsilon/\beta$.

Cf. Davis-Kahan: m eigenvectors of ϵ -perturbed matrix error ϵ/λ.

E.g. for $g_i(x) = x^3$,we have $3\sqrt{2}m^{0.5}\,\epsilon$.

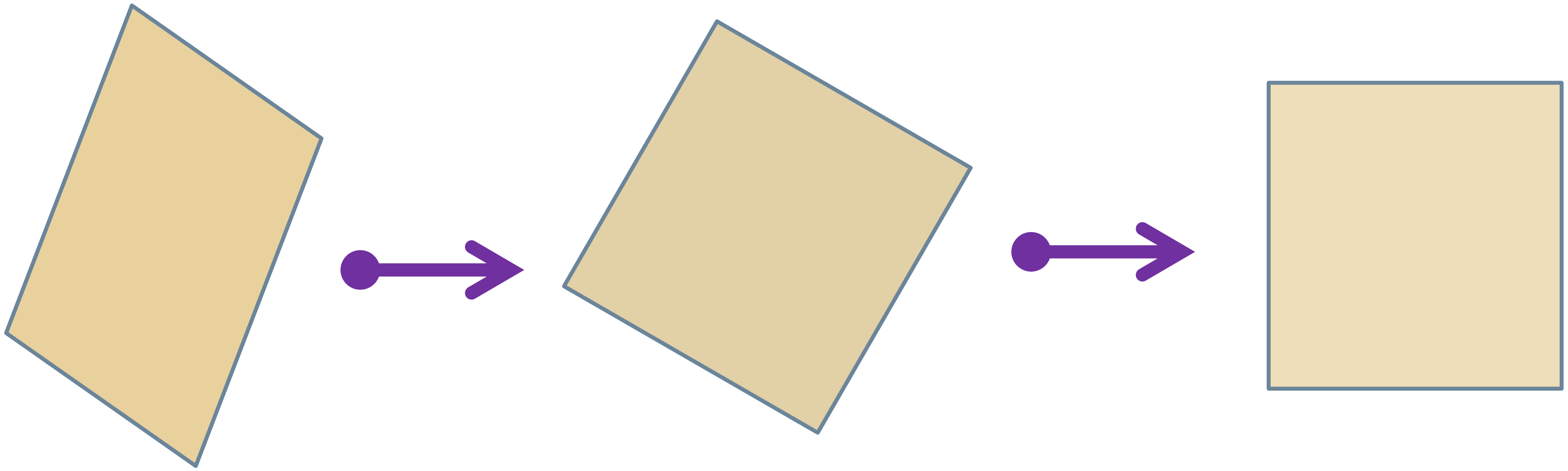
Thm 6 [Fast convergence]:

Need $\mathrm{N} = 4\left[\log_{1+2\gamma}\,\log_2\frac{\beta}{\delta\epsilon} + C\,\right]$ iterations.

E.g. for $g_i(x) = x^3$, $4\left[\log_2\,\log_2\frac{3}{2\epsilon} + C\,\right]$

Application 1:Independent Component Analysis

Y = AS   Recover independent variables by observing linear combinations.
(Cocktail party problem.)



Step 1. Whitening: normalizing covariance to I. (Use PCA).

Step 2. ICA: Recovering the rotation.

Cumulants

- Cumulant generating function $\mathrm{h(t)} = \log(E \exp(tx))$
- $h(t) = \sum \frac{1}{l!} k_l t^l$
- Polynomial in moments:

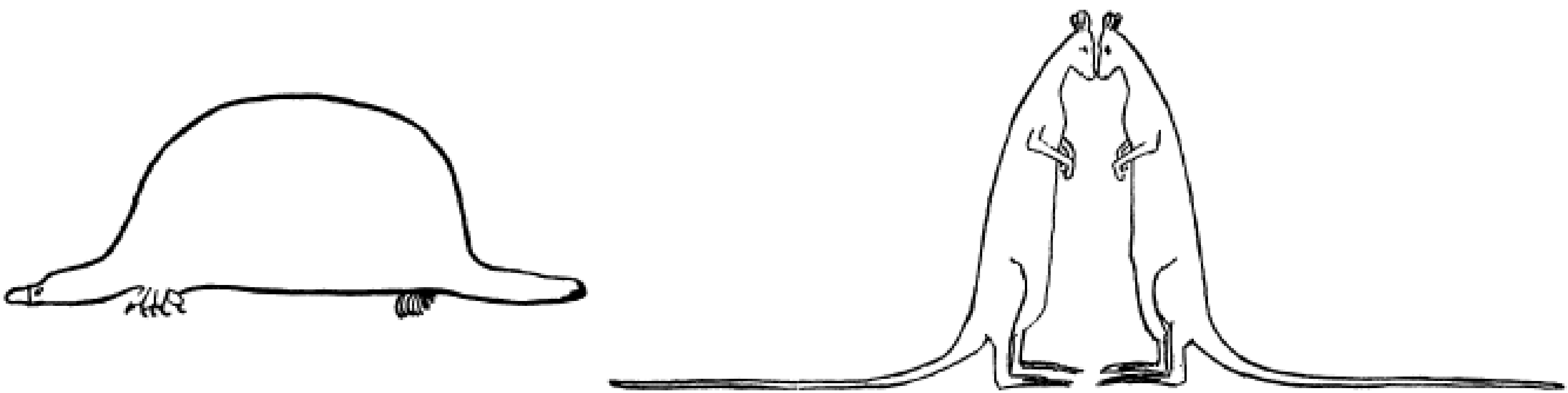
$$k_2 = \mu_2, k_3 = \mu_3, k_4 = \mu_4 - 3\mu_2 \ldots$$

- Key property: $k_l(aX + bY) = a^l k_l(X) + b^l k_l(Y)$ for independent X, Y.

Recent rebirth of moment/cumulant methods in Theoretical CS and Machine Learning. E.g. Hsu, Kakade,12 for learning Gaussian mixtures.

Kurtosis k_4

* In case any of my readers may be unfamiliar with the term "kurtosis" we may define mesokurtic as "having β_2 equal to 3," while platykurtic curves have $\beta_2 < 3$ and leptokurtic > 3. The important property which follows from this is that platykurtic curves have shorter "tails" than the

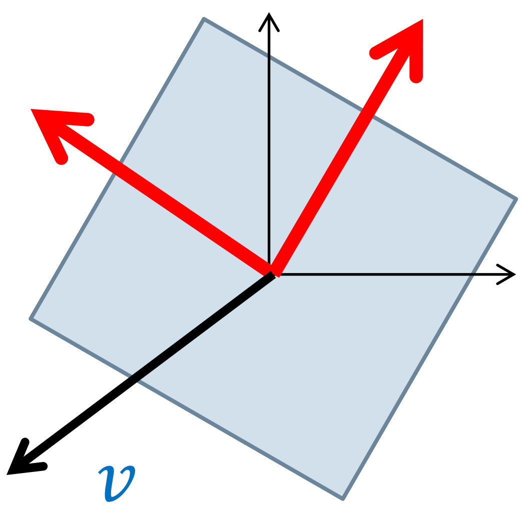


normal curve of error and leptokurtic longer "tails." I myself bear in mind the meaning of the words by the above *memoria technica*, where the first figure represents platypus, and the second kangaroos, noted for "lepping," though, perhaps, with equal reason they should be hares!

Student's drawing , 1927 (taken from the web site of K. Wuensch).

Independent Component Analysis (Step 2).

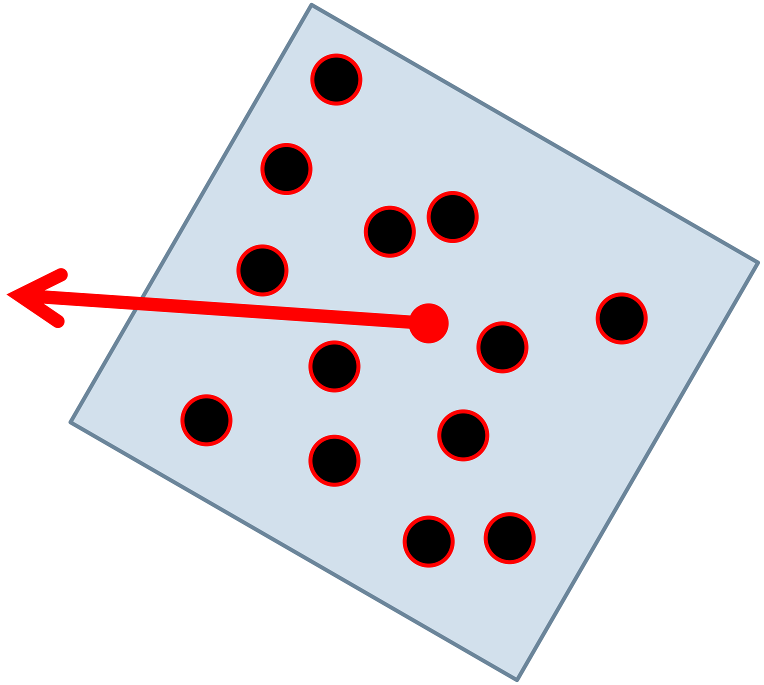
- Cumulant generating function $\mathrm{h(t)} = \log(E \exp(tx))$
- $h(t) = \sum \frac{1}{l!} k_l t^l$
- Define $f(v) = E_x \, k_l(\langle \mathrm{v}, x \rangle), \mathrm{l} > 2$



Theorem: the only maxima of $|f(v)|$ correspond to the original coordinate directions.

Estimating from data

- $f(v) = E_x\, k_l(\langle \mathrm{v}, x \rangle) \approx \frac{1}{n} \sum k_l(\langle \mathrm{v}, x_i \rangle)$



Other contrast functions are also used in practice but only cumulants are guaranteed to work.

ICA as basis encoding

From cumulant properties: $k_l(v) = k_l(\sum \mathrm{a}_i e_i) = \sum \mathrm{a}_i^l k_l(e_i)$

Put $g_i(x) = w_i x^l$, $w_i = k_l(e_i)$, $Z_i = e_i$

$F(v) = \sum_{i=1}^{k} g_i(\langle \mathrm{v}, Z_i \rangle)$.

Gradient iteration – GI-ICA algorithm (Voss, Rademacher, Belkin, NIPS13)

Recovery under Gaussian noise model.

The basis recovery theorem guarantees ICA recovery.

Stability of ICA with arbitrary (small) noise

[Belkin, Rademacher, Voss, 15]

Spectral clustering: Laplacian Embedding

- W – Weighted adjacency matrix for n-vertex graph G (a.k.a., the **similarity matrix**)
- D – **Degree matrix** with $D_{ii} = \sum_{j=1}^{n} W_{ij}$
- $L := D - W$ is the **graph Laplacian**
  - We can also handle the normalized Laplacians.
- X – Columns form the lowest k eigenvectors of L scaled to have $\sqrt{n}$-norm.
- Row expansion: $X = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$.

$\mathbf{x}_1, \dots, \mathbf{x}_n$ are the embeddings of the n vertices of G

Multi-way clustering with k-means

$$\text{Graph} \rightarrow \mathbb{R}^k$$
$$\phi\colon\ x_i \rightarrow ((e_1)_i, (e_2)_i, \ldots, (e_k)_i)$$

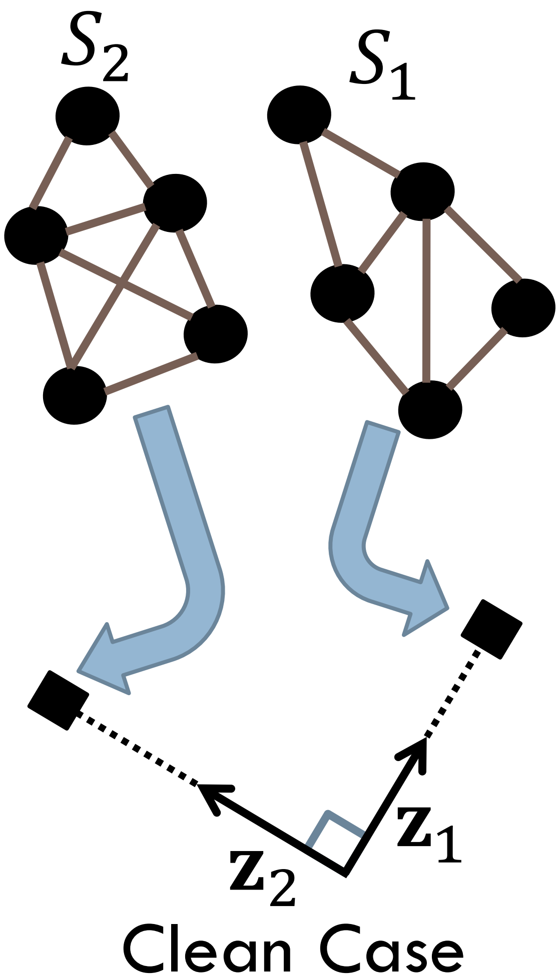
- Apply k-means in the embedding space.

(Shi, Malik 00, Ng, et al, 01, Yu, Shi 03, Bach, Jordan, 06…)

Can be justified as a relaxation of a partition problem.

However, initialization dependent and the objective function has certain peculiarities.

Spectral Embedding's Basis

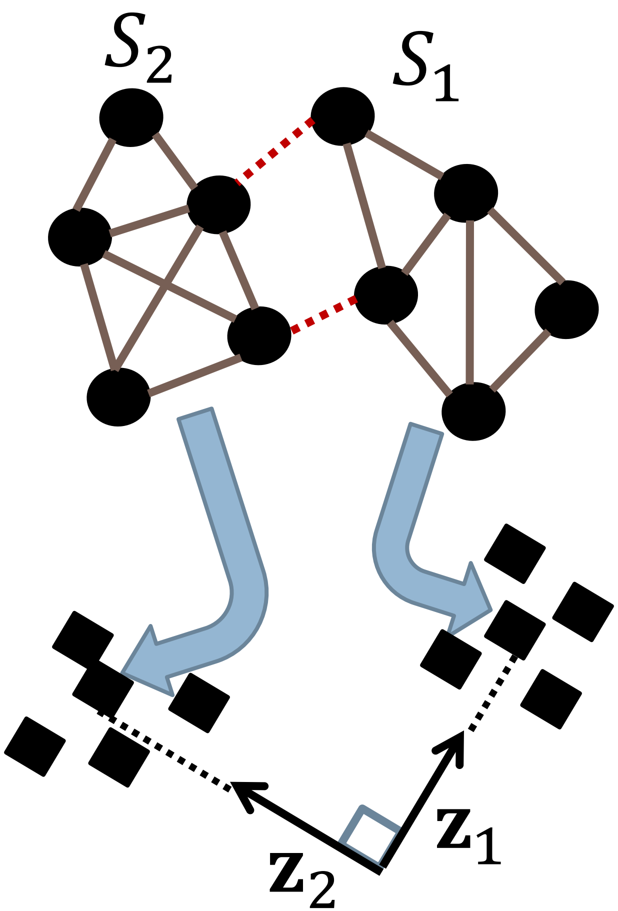


- Vertices embedded into $\mathbb{R}^k$ using the spectral embedding.
- $\mathbf{x}_i$ is the i^{th} embedded point

Fact. If G has k connected components $S_1, \ldots, S_k$, then there exists $\mathbf{z}_1, \ldots, \mathbf{z}_k$ an **orthonormal basis** of $\mathbb{R}^k$ such that

$$\mathbf{x}_i = \left|S_j\right|^{-1/2} \mathbf{z}_j \text{ for all } i \in S_j$$

Spectral Embedding's Basis



Realistic Case

- The basis structure persists under realistic conditions.

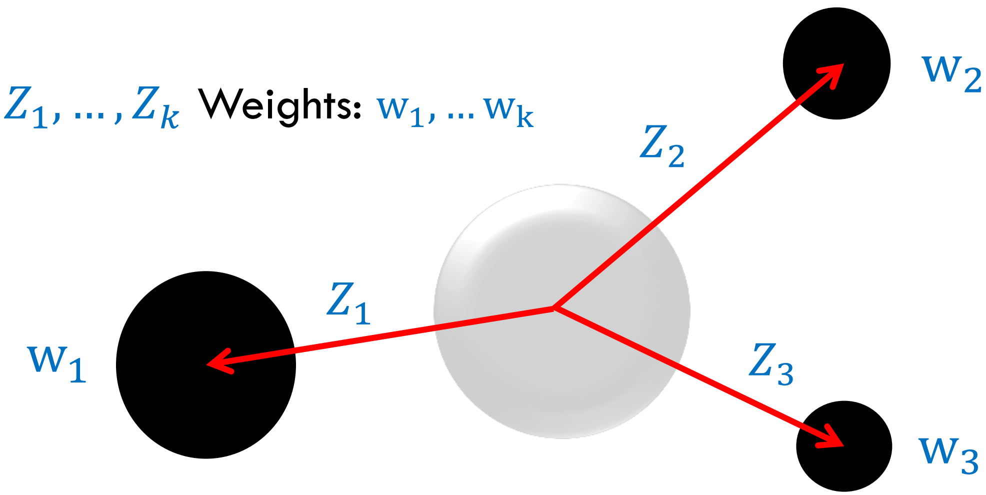
Lemma (Informal). If G has k "clusters" $S_1, \dots, S_k$ with low weight cross-edges, then there exists $\mathbf{z}_1, \dots, \mathbf{z}_k$ an **orthonormal basis** of $\mathbb{R}^k$ such that

$$\mathbf{x}_i \text{ is near } \left|S_j\right|^{-1/2}\mathbf{z}_j \text{ for all } i \in S_j$$

Spectral clustering as hidden basis recovery

Weighted basis vectors.

Basis vectors: $Z_1, \dots, Z_k$ Weights: $\mathrm{w}_1, \dots \mathrm{w}_\mathrm{k}$



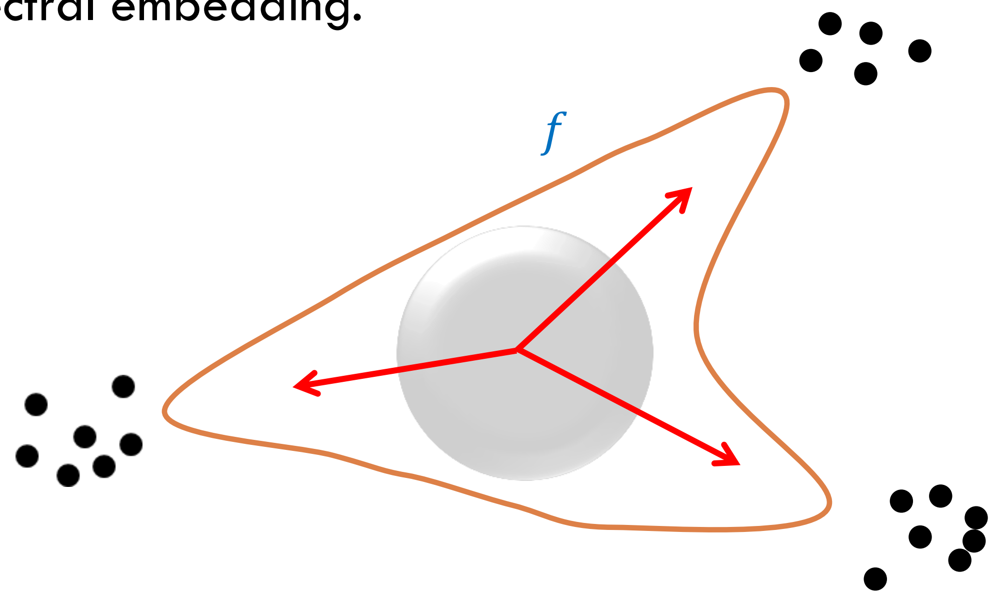
Key identity (choose g):

$$\mathrm{F}(v) = \frac{1}{n}\sum_{i=1}^{n} g(|\langle \mathrm{v}, \phi(x_i)\rangle|) = \sum_{i=1}^{k} w_i g(|\langle \mathrm{v}, Z_i\rangle|)$$

BEF: $$g_i(t) = w_i g(t/\|Z_i\|)$$

Spectral clustering as hidden basis recovery

Data after spectral embedding.



Choose allowable "contrast function" $g: R_+ \to R$.

Define $f: \mathcal{S}^{k-1} \to \mathbb{R}$ by $\mathrm{F}(v) = \sum_{i=1}^{n} g(|\langle v, \phi(x_i)\rangle|)$

(a sort of "generalized moment")

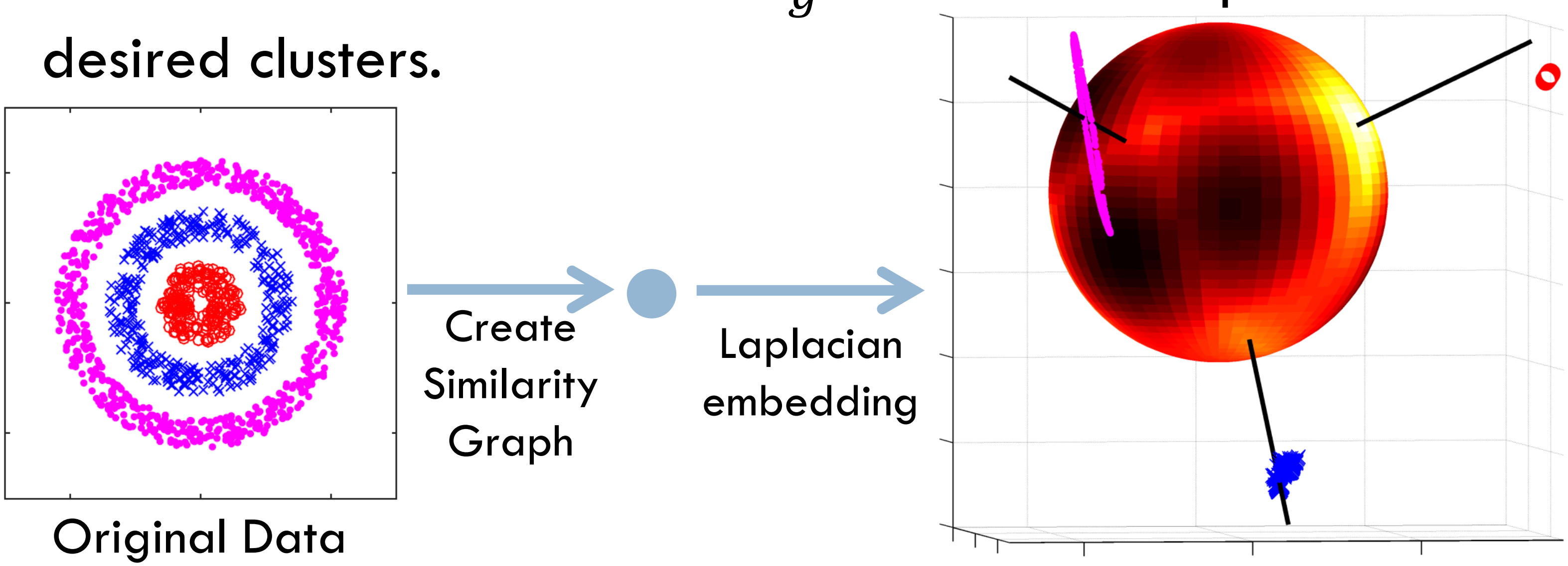
Claim: all local maxima of F "point" at the clusters.

Basis Recovery for Clustering

- We maximize an admissible contrast g over **directional projections** of the **embedded** data

$$F_g(\mathbf{u}) = \frac{1}{n}\sum_{i=1}^{n} g(|\langle \mathbf{u}, \mathbf{x}_i \rangle|)$$

- Idea: The **local maxima** of F_g on $\mathbb{S}^{k-1}$ correspond to the desired clusters.



Heat map of F_g evaluations on $\mathbb{S}^{k-1}$

Allowable contrast functions

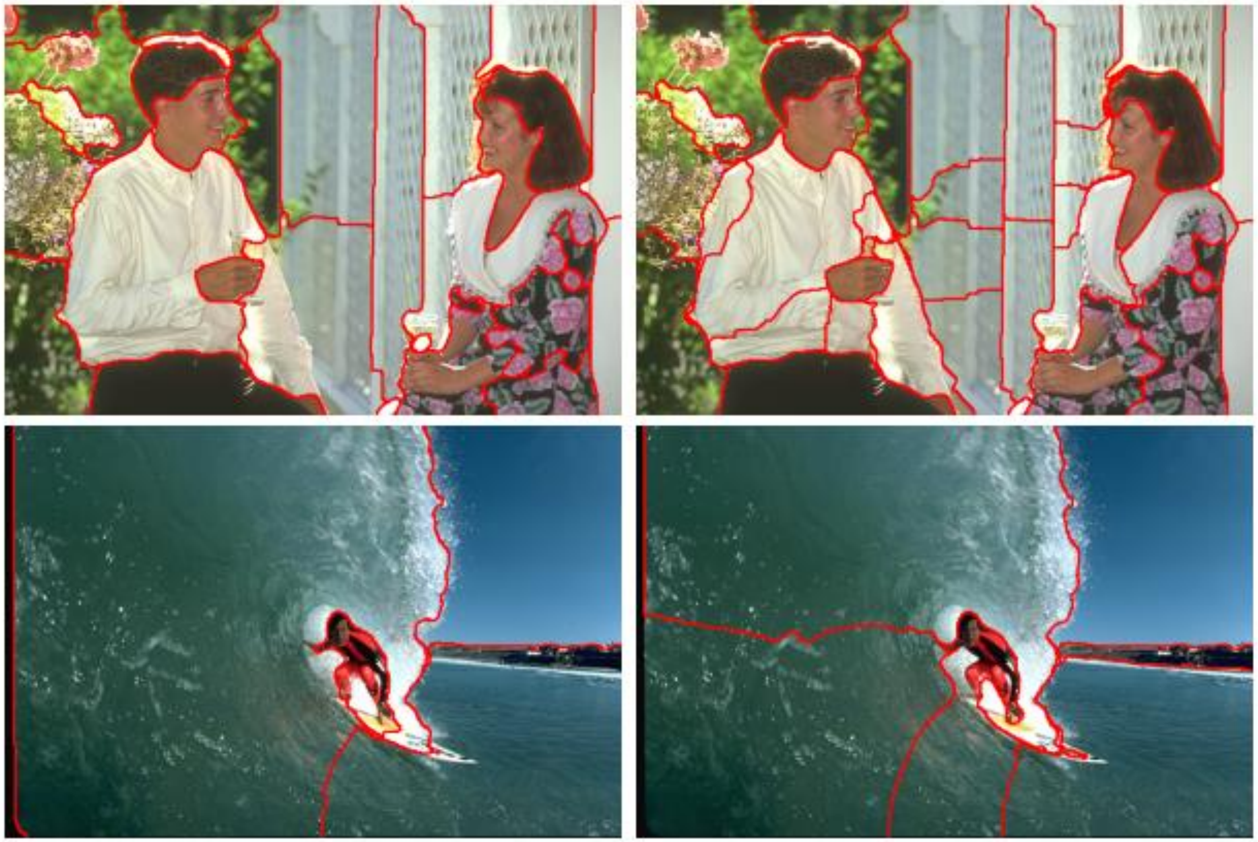
Conditions:

- $g(\sqrt{x})$ is strictly convex on $[0, \infty)$.
- $\frac{d}{dx}\left(g(\sqrt{x})\right)\Big|_{0+}$ is 0 or $+\infty$

Some examples:

- $-|x|$
- $|x^p|, p > 2$
- $\exp(-x^2)$
- $\log(\cosh x)$ [from Independent Component Analysis]

Image segmentation



Our method (left) vs k-means (right)

Stochastic block model

Stochastic block model with three unbalanced clusters + between-cluster noise.

Accuracy: 99.9% vs 42.1% for k-means.

Explanation: K-means objective function likes to split big blocks.

Clustering Accuracy Comparison

	k-means (baseline)	Choice of contrast g		
		$-\|x\|$	$\|x\|^3$	$\log\cosh(\|x\|)$
E. coli	69.0	80.9	79.3	**81.2**
flags	33.1	**36.8**	36.6	**36.8**
glass	46.8	**47.0**	**47.0**	**47.0**
thyroid	80.4	**82.4**	82.2	82.2
car eval	36.4	**37.0**	36.3	35.2
cell cycle	62.7	64.3	63.8	**64.5**

- Clustering accuracy (%) comparison of UCI data sets.
  - Compares unsupervised clusters with true data labels.
- Similarity matrices constructed via Gaussian kernel.

Summary

- Non-linear (and non-tensorial) generalization of the classical spectral decomposition and power iteration.
  - Lots of (harmless) saddle points but all local maxima are "good".
- An efficient algorithmic primitive + theoretical analysis.
- An alternative for spectral clustering.

A non-convex yet efficient optimization technique.

Should we look for "hidden convexity" elsewhere?