Avoiding the curse of dimensionality: Computational efficiency in high dimensional inference

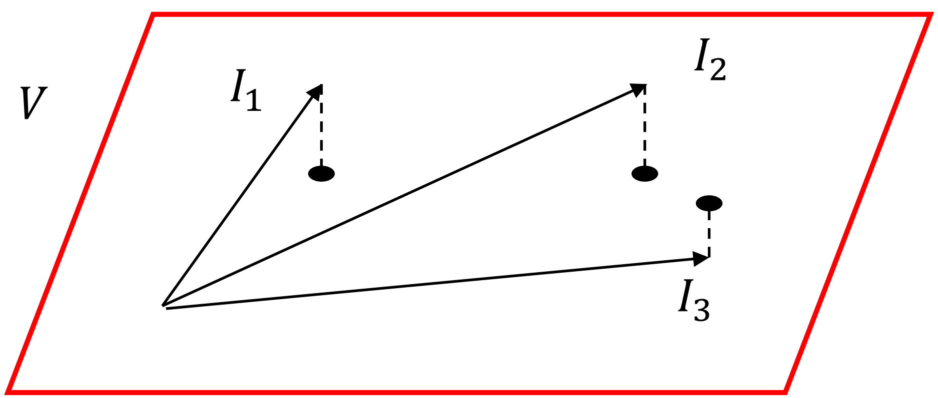
Luis Rademacher

Computer Science and Engineering

The Ohio State University

Inference helps us understand data

- Example: Principal Component Analysis
- Geometrically, given a set of n-dimensional vectors, determine whether there is a k-dimensional subspace such that they are close to it.



Algorithmic lens: Does a model have a provably efficient algorithm?

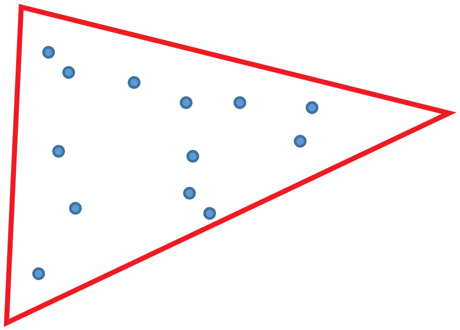
- Algorithm = Turing machine or similar formalization
- Efficient = polynomial time
  = polynomial number of steps as a function of problem size.

Example 1: Nash equilibrium

- An example of a superb model without a provably efficient algorithm.
- Given interacting agents, Nash equilibrium is a prediction of how they will act.
- An efficient algorithm is a requisite of a sound model
  - “If your laptop can’t find it, then neither can the market…” (Kamal Jain)

Example 2: Simplex learning

- Simplex learning problem [Frieze Jerrum Kannan ‘96]:
Given uniformly random points from a simplex in R^n, estimate the simplex.
- Maximum Likelihood Estimator (MLE): minimum volume simplex containing sample.
- **Theorem:** For MLE to be within constant L_1 distance, $O^*(n^2)$ samples are enough
Proof: follows from the theory of empirical processes [Vapnik Chervonenkis]
- But finding the minimum volume simplex containing a given set of points is an NP-hard problem [Packer]. (No efficient algorithm unless P=NP)



Linear feature extraction

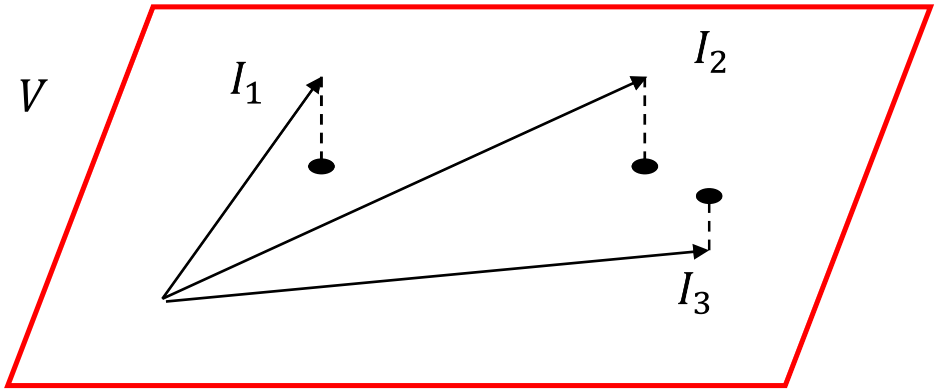
Given n-dimensional points, find new coordinates that highlight some structure of the data

- Principal component analysis (PCA)
  Find a basis of a subspace so that points are close to its span.
- Column subset selection (CSS)
  Find *a few data points* that are a basis of a subspace so that all points are close to its span.
- Independent component analysis (ICA)
  Find a basis so that coordinates of points in this basis appear *statistically independent*.

Part I: Column subset selection

Example: Eigenfaces

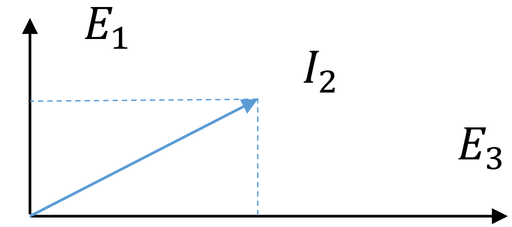
- First successful face recognition algorithm.
- Preprocessing: Principal Component Analysis (dimensionality reduction)
  Given set of 100x100 training images (faces $I_1, I_2, I_3, \ldots$), interpret as 10000-dimensional vector, find subspace V of low dimension (say 100) that is "close" to given vectors (=faces). Store projections of vectors onto V.



Example: Eigenfaces

- Dimensionality reduction via PCA:
  - Decreases computational cost
  - Highlights relevant features (de-noising).

$$I_2 = \frac{1}{2}E_1 + \frac{1}{5}E_3$$



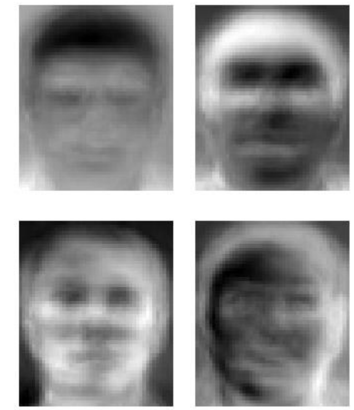


Image credit: AT&T Laboratories Cambridge.

- Numbers are weights in weighted combination of 100 “representative” images in V: singular vectors of data matrix or “Eigenfaces”.
- But Eigenfaces are not faces.
  Can we find *actual representative faces* as the basis of V and write all faces as linear combinations of a few actual faces?

Formalization:
Column subset selection

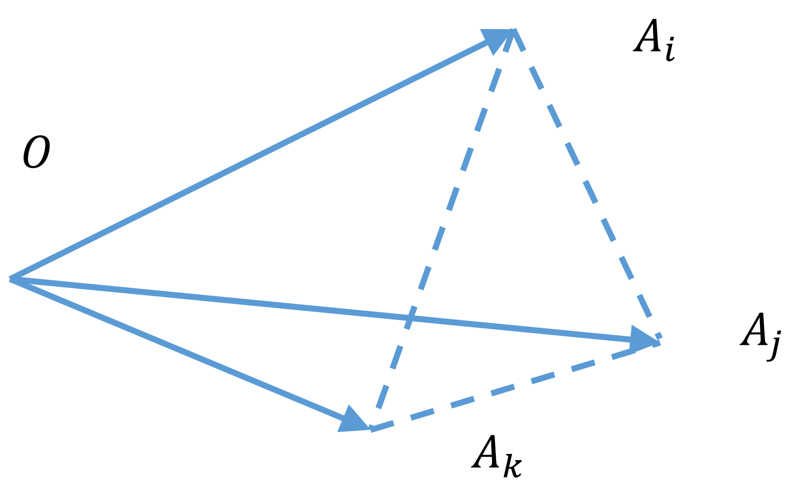
[Golub Businger] [Gu Eisentat] [Boutsidis Mahoney Drineas] [Deshpande R]…

- A refinement of principal component analysis: Given a matrix A of data points as columns,
  - PCA: find k-dim subspace V that minimizes
  $$||A - \pi_V(A)||_F^2$$
  - Subset selection: find V *spanned by k columns of A*.
    - Seemingly harder, combinatorial flavor.

  (π_V projects columns onto V)
  $\|A\|_F^2 = \sum_{ij} A_{ij}^2$ (Frobenius norm, corresponds to sum of squared distances in geometric view)

Volume sampling

- Given n-by-m matrix, pick set of k columns at random with probability proportional to squared volume of k-simplex spanned by them and origin. [Deshpande R. Vempala Wang]



CSS via volume sampling

- **Theorem: Relative error** column subset selection [Deshpande R. Vempala Wang]:
    - S: k-subset of columns according to volume sampling
    - A_k: best rank-k approximation to A in Frobenius norm, given by principal components (or Singular Value Decomposition)
    - π_S: projection of columns onto $columnspan(A_S)$
      $\Rightarrow E_S(\|A - \pi_S(A)\|_F^2) \leq (k+1)\|A - A_k\|_F^2$
- Factor "$k + 1$" is best possible [DRVW]

Volume sampling: probabilistic method in linear algebra

- Choose a suitable distribution over a set of objects $\omega \in \Omega$.
- Show that in expectation a quantity of interest $X(\omega)$ is small
- Conclude that $X(\omega)$ is small for some ω.

- Volume sampling over k-subsets of columns of A.
- $E_S(\|A - \pi_S(A)\|_F^2) \leq (k+1)\|A - A_k\|_F^2$
- $\Rightarrow$ there exist k columns of A such that
  $\|A - \pi_S(A)\|_F^2$
  $\leq (k+1)\|A - A_k\|_F^2$

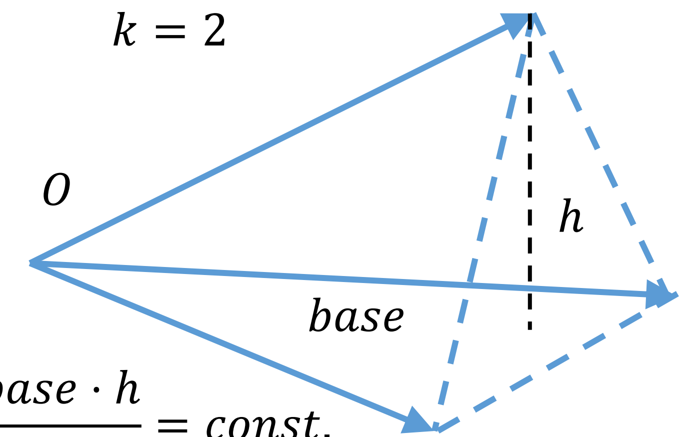
Where does volume sampling come from?

- **No self-respecting architect leaves the scaffolding in place after completing the building.**

  Gauss?

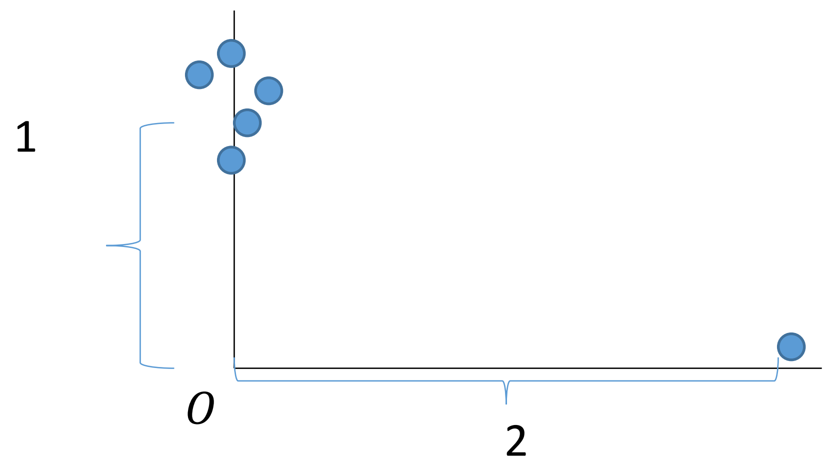
Where does volume sampling come from?

- Illustrative simple cases:
  - For picking k out of $k+1$ points, k with *maximum* volume is optimal.
  - For picking 1 out of m, random according to squared length is better than max. length.
  - For k out of m, this suggest volume sampling.



$$Vol = \frac{base \cdot h}{3} = const.$$

$$h^2 = error$$

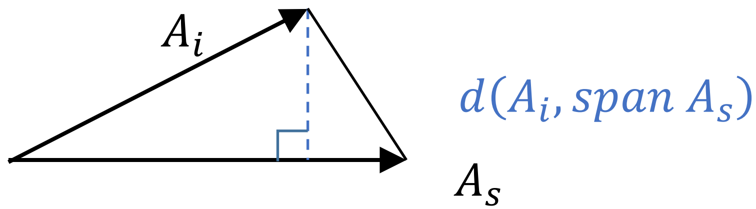


Where does volume sampling come from?

- Why does the algebra work? Idea:
  - When picking 1 out of m random according to squared length, expected error is sum of squares of areas of triangles:

$$\mathrm{E(error)} = \sum_s \frac{\|A_s\|^2}{\sum_t \|A_t\|^2} \sum_i d\big(A_i, span(A_s)\big)^2$$

$$= \frac{1}{\sum_t \|A_t\|^2} \sum_{s,i} \|A_s\|^2 \, d\big(A_i, span(A_s)\big)^2$$



  - This sum corresponds to certain coefficient of the characteristic polynomial of $A^T A$, which can be computed efficiently.

    An example of Valiant's observation? Many algorithms that count efficiently are based on efficient computation of the determinant.

Efficient volume sampling: Key idea
[Deshpande R.][Deshpande Kundu R.]

- For every column, compute probability of including it (given past choices) and include it with that probability. For 1st column and S according to volume sampling:

$$P(1 \notin S) = \frac{\sum_{S' \subseteq \{2 \ldots m\}, |S'|=k} (\mathrm{vol}\, A_{S'})^2}{\sum_{S' \subseteq \{1 \ldots m\}, |S'|=k} (\mathrm{vol}\, A_{S'})^2}$$

$$= \frac{\sum_{S' \subseteq \{2 \ldots m\}, |S'|=k} \det A_{S'}^T A_{S'}}{\sum_{S' \subseteq \{1 \ldots m\}, |S'|=k} \det A_{S'}^T A_{S'}} = \frac{c_{m-k}({A_{-1}}^T A_{-1})}{c_{m-k}(A^T A)}$$

($c_i(A) \ = \ i^{\text{th}}$ coefficient of characteristic polynomial of A)

($A_{-1} = A$ without the first column)

- Similar formula for subsequent rows.

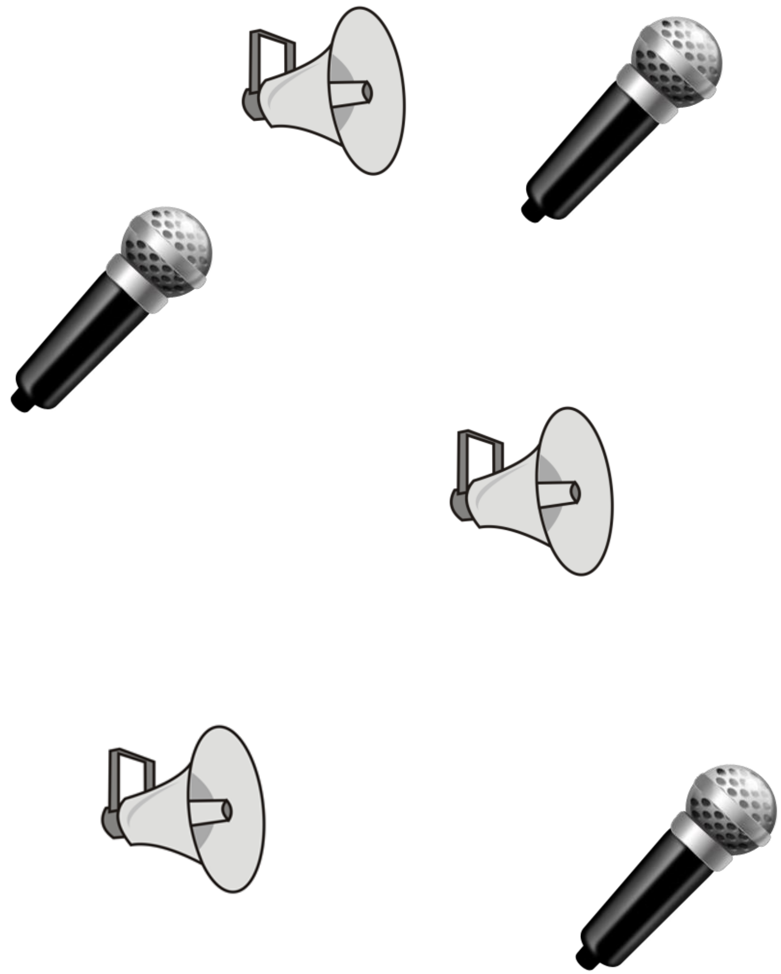
Other applications of volume sampling

- The “Paris Hilton” problem:
  If Google simply returns top ranked results for “paris hilton”, then no results about Hilton hotel in Paris …
  - [Kulesza Taskar (ICML ‘11)]
    Volume sampling to select a set of “diverse” data points (large volume $\approx$ diverse).
- [Guruswami, Sinop (FOCS ‘11)]
  - Improved approximation algorithms for Quadratic Integer Programs using subset selection for Semidefinite Programming rounding.

Part II: Independent Component Analysis

Cocktail Party Problem (prototypical)

- Problem: n persons speaking in a room with n microphones.
- Microphones capture a superposition of the speech signals.
- Goal: Recover each persons' speech.



Independent Component Analysis (ICA)

- INPUT: samples $X^{(1)}, X^{(2)}, \ldots$ from random vector $X = AS$, where:
  - S is d-dimensional random vector with independent coordinates. Assume 0-mean for simplicity.
  - A is square invertible matrix.
- GOAL: estimate (directions of columns of) A.
- S, A are not observed. Distribution of S is unknown.

- Wanted: provably efficient and accurate algorithms with wide applicability.

Cocktail party problem as ICA

- Source signals (speech) at time t: $S_1^{(t)}, \ldots, S_n^{(t)}$, assumed to be statistically independent.
- Observed signals: $X_1^{(t)}, \ldots, X_n^{(t)}$, satisfy
$$X^{(t)} = AS^{(t)}$$

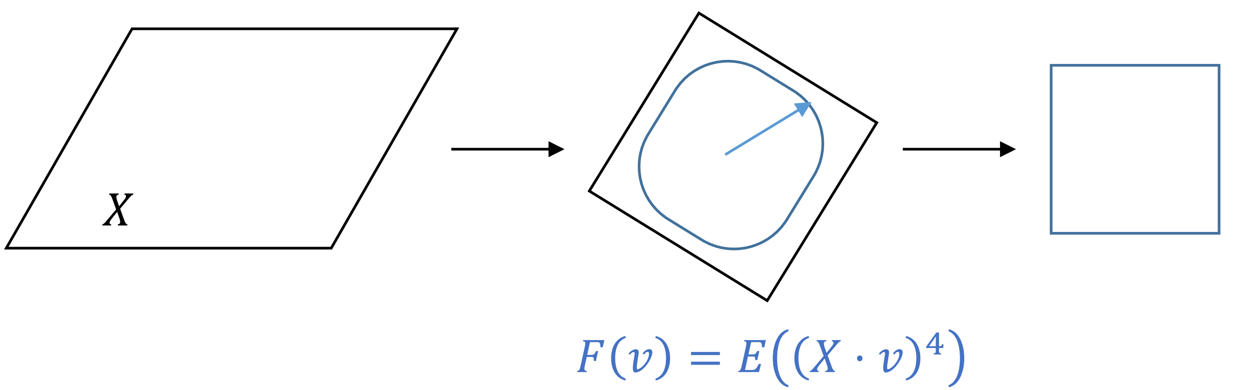
(Unknown mixing matrix A encodes geometry of persons and microphones)

- Estimate A and S from $X_1^{(t)}, \ldots, X_n^{(t)}$.

An ICA algorithm: unexpected usefulness of local optima

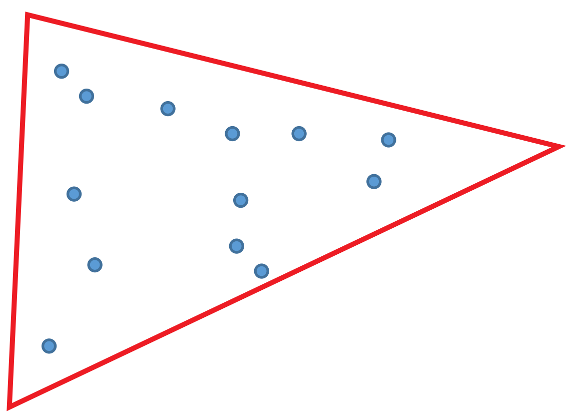
[Delfosse-Loubaton SignalProcessing95] [Frieze-Jerrum-Kannan FOCS96] [Hyvarinen IEEE NeuralNets99]

- Illustrative case: Estimate a parallelepiped from uniformly random samples $X^{(1)}, X^{(2)}$, ... Model:
  S: uniform in axis aligned cube.
  $X = AS$: uniform in a parallelepiped
- By estimating mean and covariance, can assume it is a rotated cube centered at 0.
- To estimate rotation: Enumerate all local minima of directional 4th moment on unit sphere.
  Theorem: Normals to facets are a complete set of local minima.



Independent component analysis beyond independence

- Simplex learning problem:
  Given uniformly random points $X^{(1)}, X^{(2)}$, …from a simplex in R^n, estimate the simplex.
  - An open problem from [Frieze Jerrum Kannan FOCS '96]
  - Applications to *topic modeling* [Anandkumar Foster Hsu Kakade Liu] [Anandkumar Ge Hsu Kakade Telgarsky]



Simplex learning via ICA

- Idea [Anderson Goyal R. '13]:
  Use the following transformation.
  Theorem: Let X be uniformly random in standard simplex.
  Let $Y = TX$, where $T \sim Exp(n)$ and independent of X.
  Then $Y_i \sim Exp(1)$ and **independent**.
- This works even after linear transformation!
- Y has independent coordinates in some basis.
  ICA algorithm recovers that basis, which recovers the vertices of the simplex.

Heavy-tailed ICA

- All previously known provably efficient ICA methods require at least 4 moments.
- Heavy-tailed distribution $\approx$ no moments or only a few moments exists.
- Heavy-tailed ICA instances appear naturally in speech and financial data.
- [Anderson Goyal Nandi R.]
  - **Preprocessing: Gaussian damping.**
    A provably efficient algorithm that works with no moment assumption when the unknown matrix A is unitary.
  - **Preprocessing: Gaussian damping + centroid body orthogonalization.**
    A provably efficient algorithm that works assuming finite 1st moment, for any matrix.

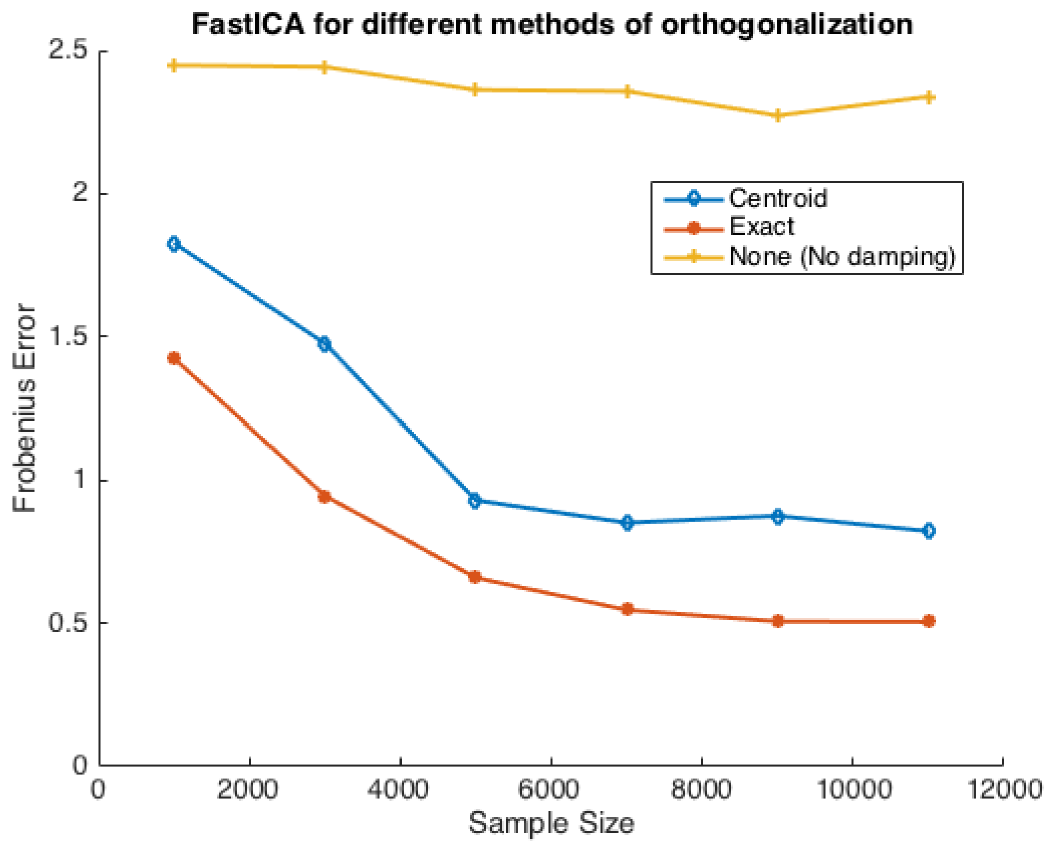
Practical implementation: Experimental results

ICA on 10-dimensional synthetic data with two-heavy tailed components.

"None": Hyvarinen's FastICA, a popular ICA algorithm. No proof of correctness for heavy tailed data.

"Centroid": our preprocessing followed by FastICA.

"Exact": Exact orthogonalization followed by damping and FastICA



Questions?