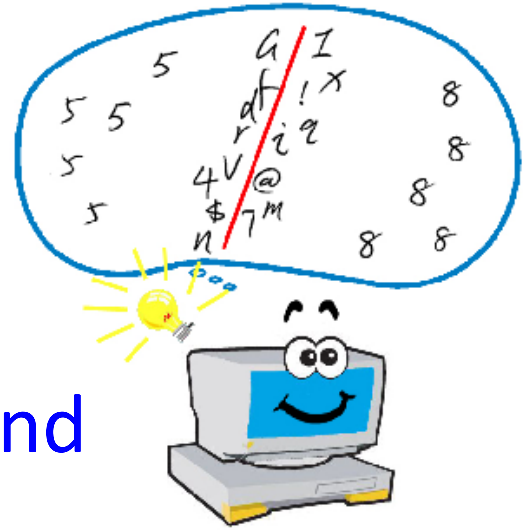
Dagstuhl 2014 Workshop



Adventures in Linear Algebra++ and unsupervised learning.

Sanjeev Arora

Princeton University
Computer Science + Center for Computational Intractability

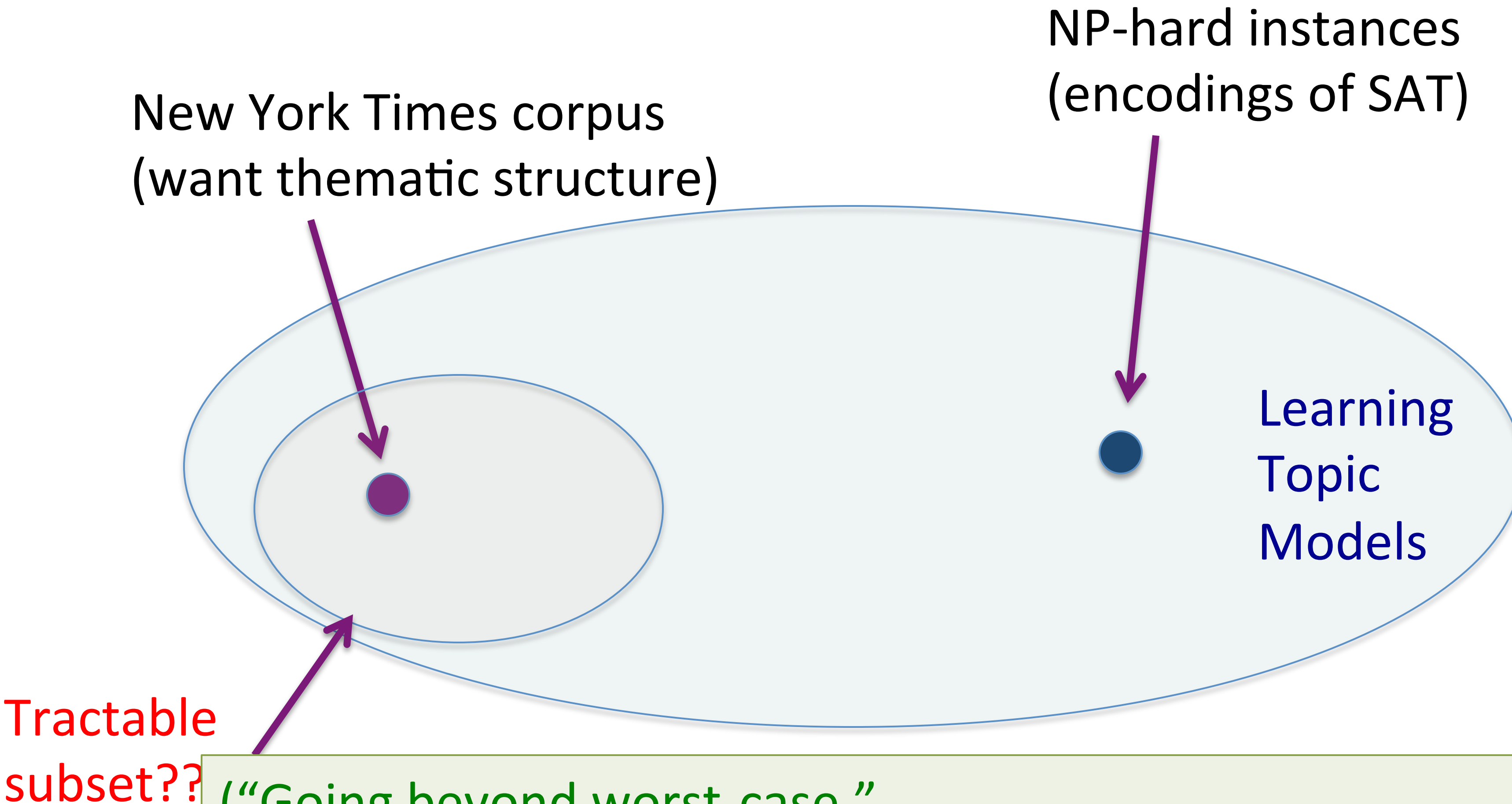
Linear Algebra ++

Set of problems and techniques that extend classical linear algebra.

Often are (or seem) NP-hard; currently solved via nonlinear programming heuristics.

For provable bounds need to make assumptions about the input.

Is NP-hardness an obstacle for theory?



("Going beyond worst-case."
"Replacing heuristics with algorithms with provable bounds")

Classical linear algebra

- Solving linear systems: Ax =b

- Matrix factorization/rank M =AB;
  (A has much fewer columns than M)

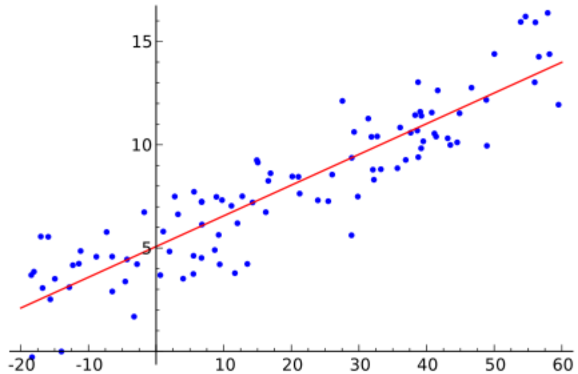
- Eigenvalues/eigenvectors. ("Nice basis")

$$M = \sum_i \lambda_i u_i u_i^T = \sum_i \lambda_i u_i \otimes u_i$$

Classical Lin. Algebra: least square variants

- Solving linear systems: Ax =b

$$\min_x \|Ax - b\|^2 \quad \text{(Least squares fit)}$$



- Matrix factorization/rank M = AB;
  (A has much fewer columns than M)

$$\min \|M - AB\|^2 \quad \text{A has r columns} \ \rightarrow \text{rank-}r\text{-SVD}$$

(“PCA” [Hotelling, Pearson, 1930s]) (“Finding a better basis”)

Semi-classical linear algebra

$$Ax = b \quad \text{s.t. } x \geq 0. \quad \text{(LP)}$$

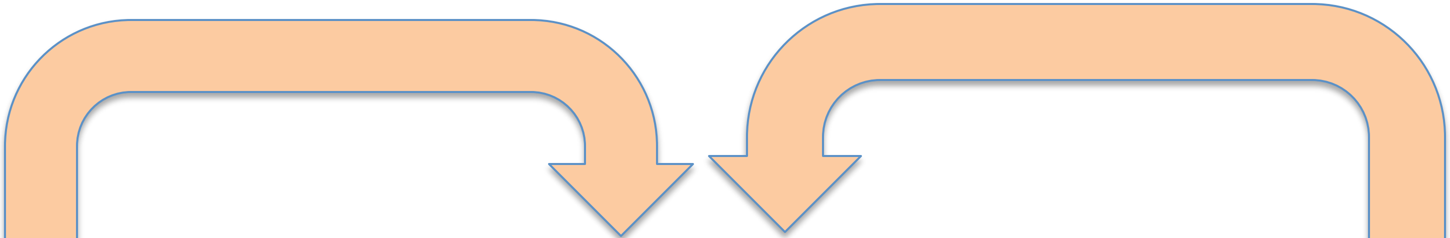
$$Ax = b$$

$$x \text{ is sparse}$$

Can be solved via LP if A is random/incoherent/RIP (Candes,Romberg, Tao;06) ("l_1-trick")

Goal in several machine learning settings: Matrix factorization analogs of above: Find M =AB with such constraints on A, B (NP-hard in worst case)

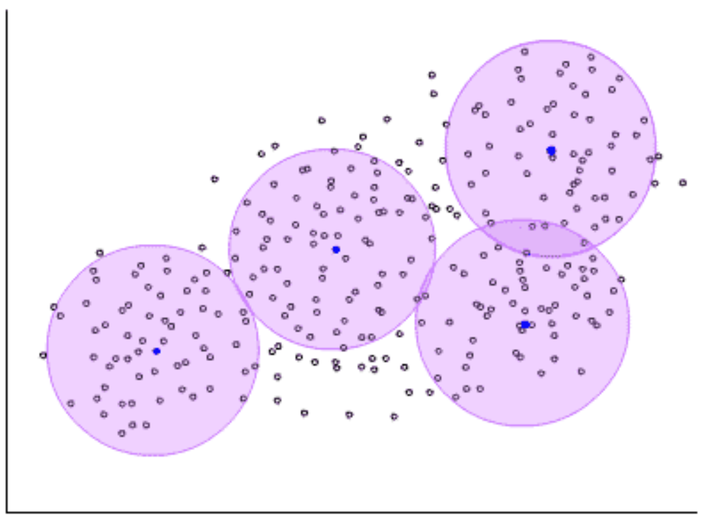
(Buzzwords: Sparse PCA, Nonnegative matrix factorization, Sparse coding, Learning PCFGs,…)



Linear algebra

Linear Algebra++

Combinatorial + Convex Optimization



Example: k-means =

Least-square rank-k matrix factorization

- each column of B has one nonzero entry and it is 1
  (sparsity + nonneg + integrality)

Matrix factorization: Nonlinear variants

NETFLIX

Given M produced as follows: Generate low-rank A, B, apply nonlinear operator f on each entry of AB.

Goal: Recover A, B   "Nonlinear PCA" [Collins, Dasgupta, Schapire'03]

Deep Learning	f(x) = sgn(x) or sigmoid(x)
Topic Modeling	f(x) = output 1 with Prob. x . (Also, columns of B are iid.)
Matrix completion	f(x) = output x with prob. p, else 0

Possible general approach? Convex relaxation via nuclear norm minimization [Candes,Recht'09] [Davenport,Plan,van den Berg, Wooters'12]

Tensor variants of spectral methods

Spectral decomposition:

$$M = \sum_i \lambda_i u_i u_i^T = \sum_i \lambda_i u_i \otimes u_i$$

Analogue decomposition for n x n x n tensors may not exist.

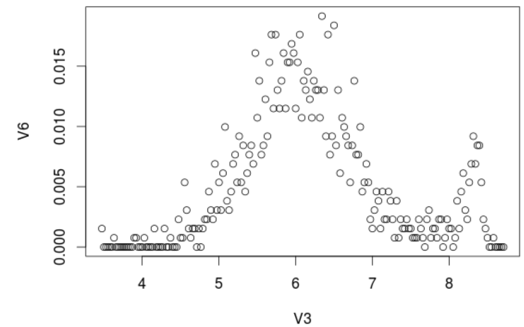
But if it does, and it is "nondegenerate", can be found in poly time.
Many ML applications via inverse moment problems.
See [Anandkumar,Ge, Hsu, Kakade, Telgarsky'13]
and talks of Rong and Anima later.

Applications to unsupervised learning…

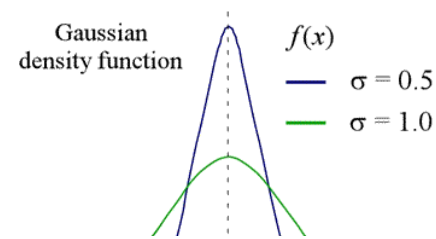
Main paradigm for unsupervised Learning

Given: Data

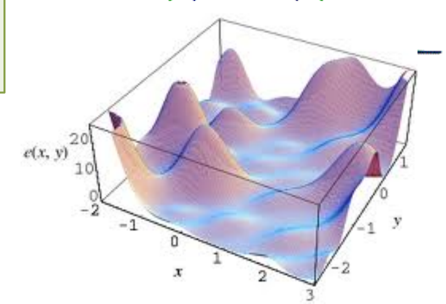
Assumption: Is generated from a prob. distribution that's described by small # of parameters. ("Model").



HMMs, Topic Models, Bayes nets, Sparse Coding, …



Learning ≅ Find good fit to parameter values (usually, "Max-Likelihood")



Difficulty: NP-hard in many cases.
Nonconvex; solved via heuristics

Recent success stories…..

Ex 1: Inverse Moment Problem

X ε R^n : Generated by a distribution D with vector of unknown parameters A.

$$M_1 = E[X] = f_1(A)$$
$$M_2 = E[XX^T] = f_2(A)$$
$$M_3 = E[X^{\otimes 3}] = f_3(A)$$

For many distributions, A may in principle be determined by these moments, but finding it may be NP-hard.

Under reasonable "nondegeneracy" assumptions, can be solved via tensor decomposition.

HMMs [Mossel-Roth06, Hsu-Kakade 09];

Topic Models[Anandkumar et al.'12]; many other settings [AGHKT'13]

Ex2: Topic Models

Given corpus of documents uncover their underlying thematic structure.

"Bag of words" Assumption in Text Analysis

=

=

	.
Banana	3
	.
	.
Snow	1
Soccer	0
	.
	.
Walnut	5
	.

Document Corpus = Matrix
(i^{th} column = i^{th} document)

Hidden Variable Explanation

- Document = Mixture of Topics

	.			.		.
Banana	3			3%		0
	.			.		.
	.			.		.
Snow	1	=	0.8	0	+ 0.2	4%
Soccer	0			0		0
	.			.		.
	.			.		.
Walnut	5			5%		0

Nonnegative Matrix Factorization

Given nx m nonnegative matrix M write it as M =AB;
A, B are nonneg. A is n x r; B is r x m

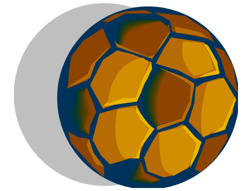
[A,Ge,Kannan, Moitra'12] $n^{f(r)}$ time worst case (also matching complexity lowerbound);

Poly(n) time if M is separable.

[A., Ge, Moitra'12] Use it to do topic modeling with separable topic matrix in poly(n) time
(Very practical; fastest current code uses it ;
[A,Ge, Halpern, Mimno, Moitra, Sontag, Wu, Zhu, ICML'13])

“Separable” Topic Matrices



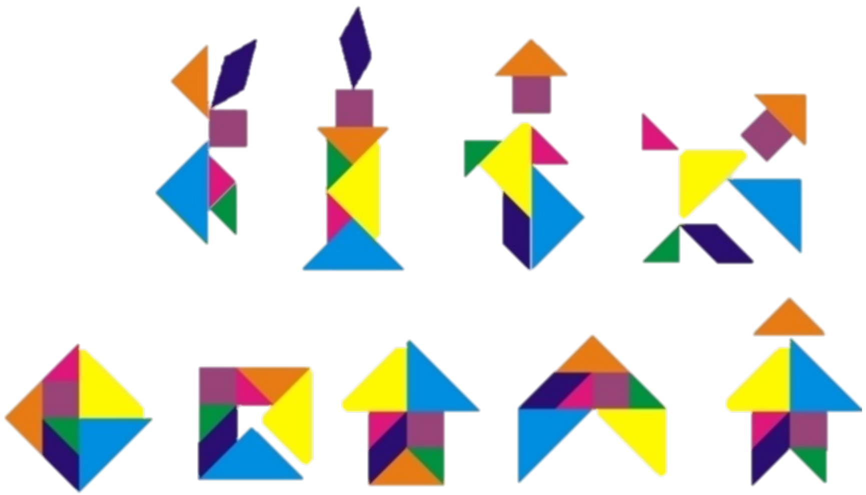


	.		.	.	.
Banana	0		0	.	.
	.		.	.	.
	.		.	.	.
Snow	4%		**0**	**0**	**0**
Soccer	0		8%	.	.
	.		.	.	.
	.		.	.	.
Walnut	0		0	.	.
	.		.	.	.

Notion also useful in vision, linguistics [Cohen, Collins ACL’14]

Ex 3: Dictionary Learning
(aka Sparse Coding)

- Simple "dictionary elements" build complicated objects.

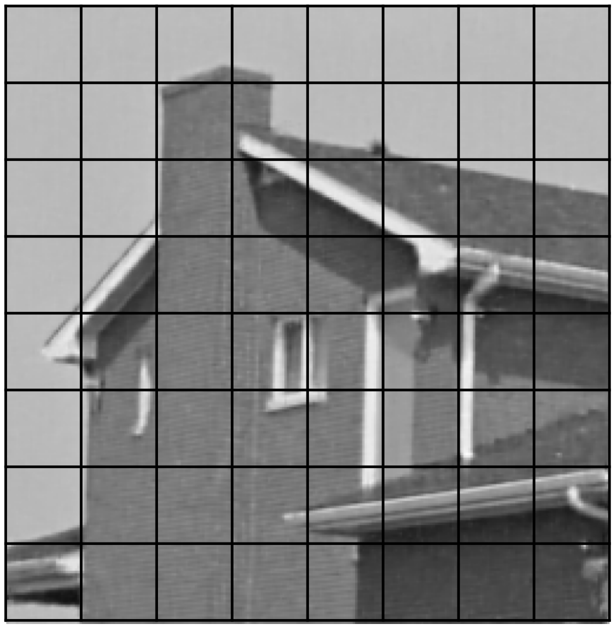


- Each object composed of small # of dictionary elements (sparsity assumption)
- Given the objects, can we learn the dictionary?

Given: Samples y_i generated as $A\, x_i$, where x_i's k-sparse, iid from some distrib.

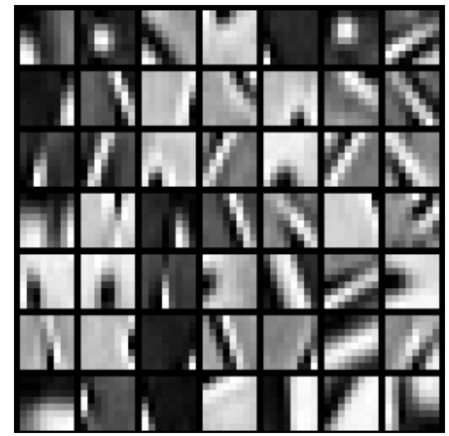
Goal: Learn matrix A, and x_i's

Why dictionary learning? [Olshausen Field '96]



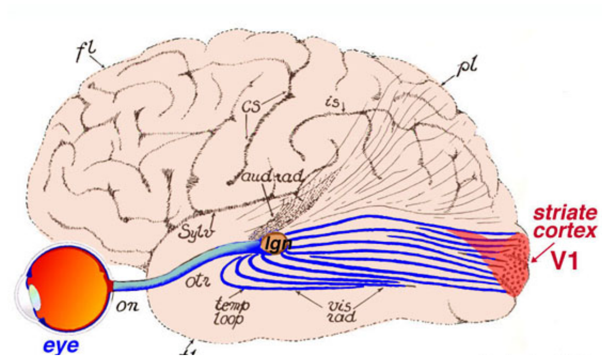
natural image patches

dictionary learning



Gabor-like Filters

Other uses: Image Denoising, Compression, etc.



“Energy minimization” heuristic

$$\min_{B, x_1, x_2, \ldots,} \sum_i \|y_i - Bx_i\|_2^2$$

x_i's are k-sparse

- Alternating Minimization (kSVD):
  Fix one, improve the other; REPEAT

- Approximate gradient descent (“neural”)

[A., Ge,Ma,Moitra’14] Under some plausible assumptions, these heuristics find global optimum.

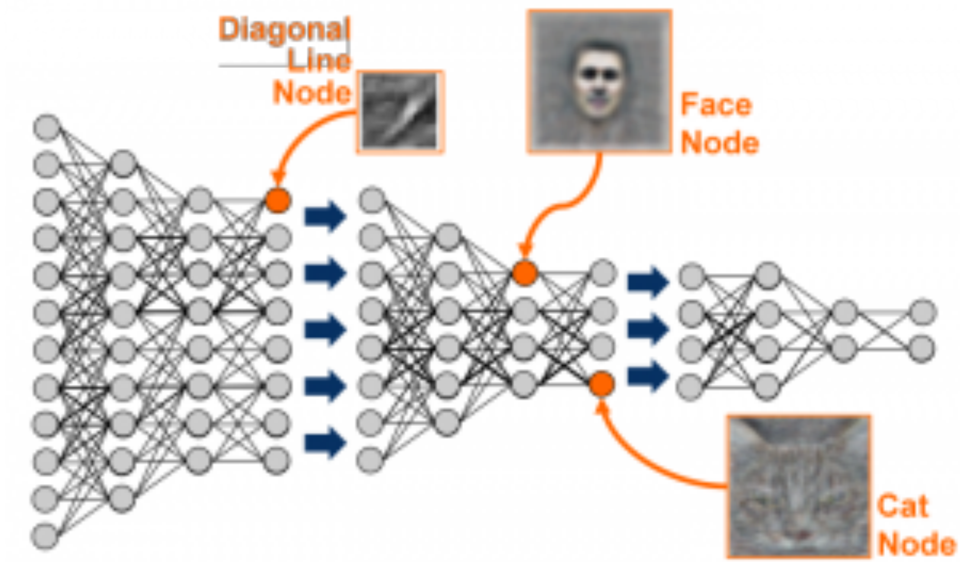
Lots of other work, including an approach using SDP hierarchies.

[Barak, Kelner, Steurer’14]

Ex 4: A Theory for Deep Nets?

Deep learning: learn multilevel representation of data (nonlinear)

(inspired e.g. by 7-8 levels of visual cortex)

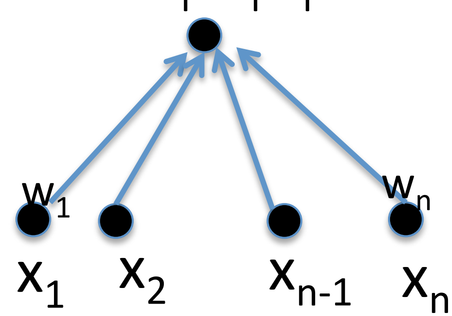


Successes: speech recognition, image recognition, etc.

[Krizhevsky et al NIPS'12.]
600K variables; Millions of training images. 84% success rate on IMAGENET (multiclass prediction).

(Current best: 94% [Szegedy et al'14])

$1 \text{ iff } \Sigma_i w_i x_i > \Theta$



Understanding "randomly-wired" deep nets

Inspirations: Random error correcting codes, expanders, etc…

[A.,Bhaskara, Ge, Ma, ICML'14] Provable learning in Hinton's generative model. Proof of hypothesized "autoencoder" property.

- No nonlinear optimization.
- Combinatorial algorithm that leverages correlations.

"Inspired and guided" Google's leading deep net code [Szegedy et al., Sept 2014]

Example of a useful ingredient

Perturbation bounds for top eigenvector (Davis-Kahan, Wedin)

v_1: top eigenvector for A
v_1': top eigenvector for A +E

If $|Ev_1| << $ difference of top two eigenvals of A,
then $v_1' \approx v_1$

Open Problems (LinAL++)

- NP-hardness of various LinAl++ problems?
- Generic $n^{f(r)}$ time algorithm for rank-r matrix decomposition problems (linear/nonlinear)? (Least square versions seem most difficult.)
- Efficient gradient-like algorithms for LinAL++ problems, especially nonlinear PCA?
  (OK to make more assumptions)
- Application of LinAl++ algorithms to combinatorial optimization?
- Efficient dictionary learning beyond sparsity $\sqrt{n}$?

Open Problems (ML)

- Analyse other local-improvement heuristics.
- More provable bounds for deep learning.
- Rigorous analysis of nonconvex methods (variational inference, variational bayes, belief propagation..)
- Complexity theory of avg case problems (say interreducibilityin Lin Al++)?

Variants of matrix factorization (finding better bases)

Rank: Given n x m matrix M rewrite it (if possible) as
$M = A B$ (A: n x r; B: r x m)

"Least squares" version: min $|M - AB|^2$ (rank-r SVD)

Nonnegative matrix factorization: M, A, B have nonneg entries
Solvable in n^r time; [AGKM'12, M'13]; in poly time for separable M

Sparse PCA: Rows of A are sparse. (Dictionary learning is a special case.) Solvable under some condns.

Least squares versions of above are open (k-means is a subcase…)