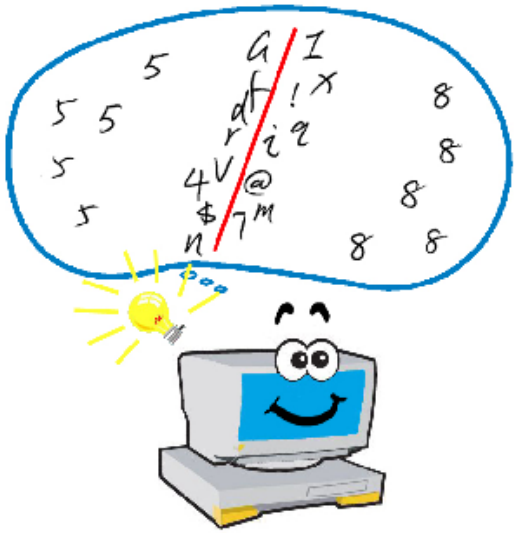
STACS 2015



Overcoming intractability for Unsupervised Learning

Sanjeev Arora

Princeton University
Computer Science + Center for Computational Intractability

(Funding: NSF and Simons Foundation)

Supervised vs Unsupervised learning

Supervised: Given many photos labeled with whether or not they contain a face, generate labels for new photos.

(STOC/FOCS: PAC learning.
In ML: Support vector machines,
online prediction, logistic regression, Neural nets etc…)

Unsupervised: Use google news corpus to answer analogy queries
King: Queen :: Waiter : ??

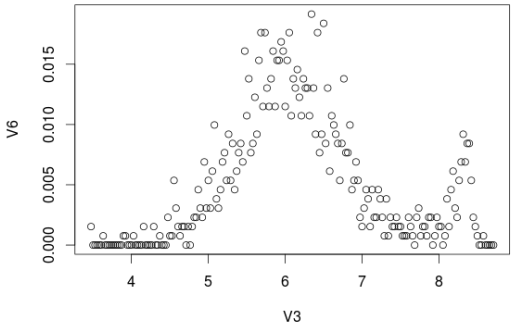
Unlabeled data >> labeled data.
("Big data" world)

CMU

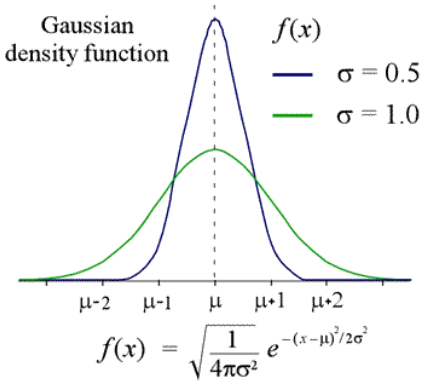
Main paradigm for unsupervised Learning

Given: Data

Assumption: Generated from prob. distribution described by small # of parameters. ("Model").



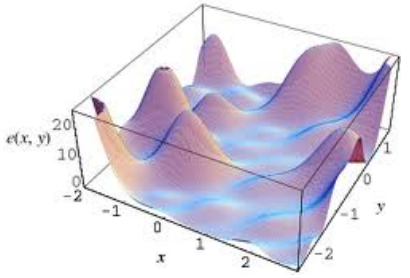
HMMs, Topic Models, Bayes nets, Sparse Coding, …



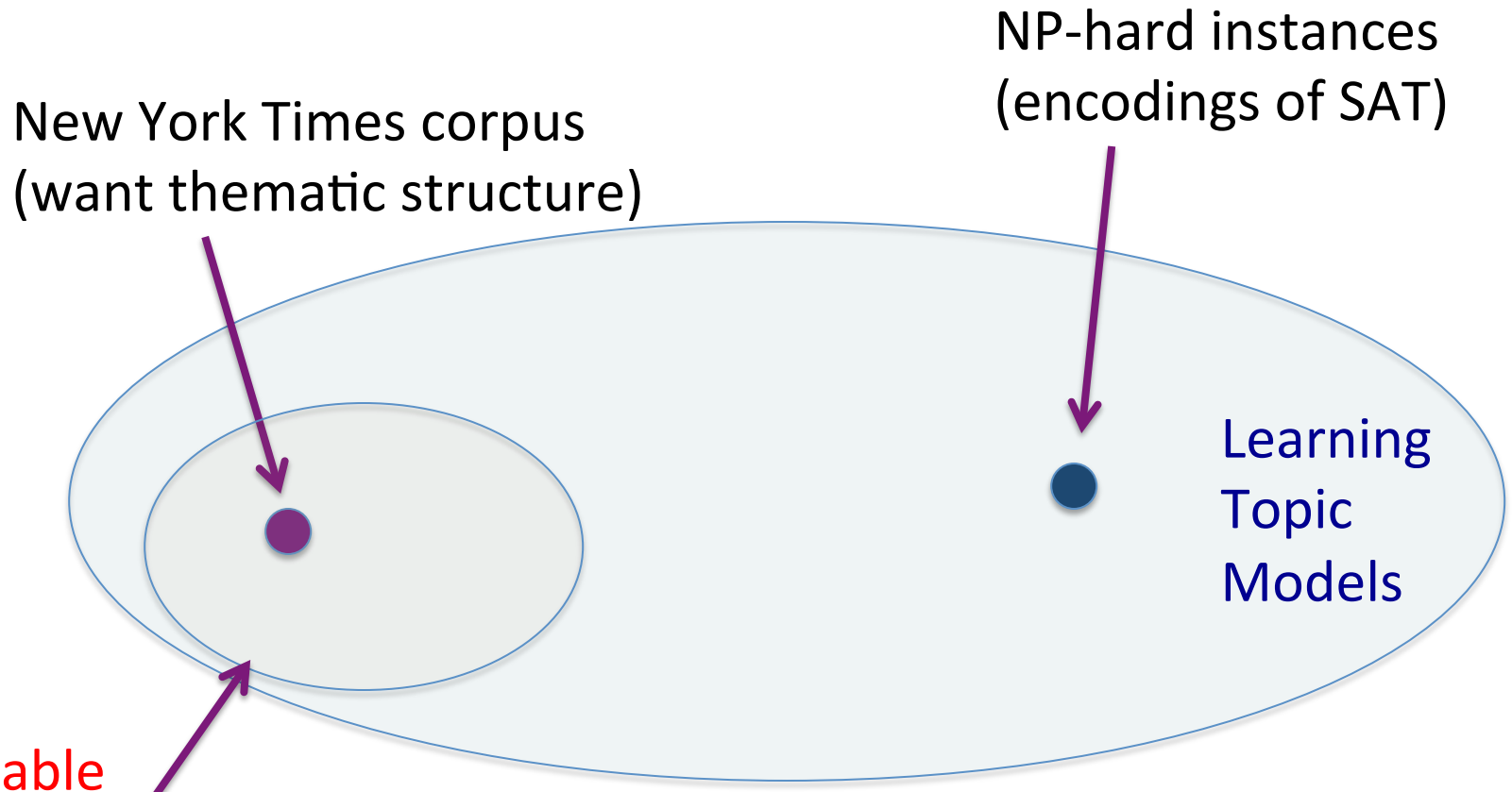
Learning ≅ Find good fit to parameter values (often, "Max-Likelihood")

Difficulty: **NP-hard** in many cases.

Nonconvex; solved via heuristics



Is NP-hardness an obstacle for theory?



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

Example: Inverse Moment Problem

$X \in R^n$: 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.

Recent progress (see later): Can find A in poly time in many settings under mild "nondegeneracy" conditions on A.

"Tensor decomposition" [Anandkumar, Ge, Hsu, Kakade, Telgarsky 2012]

Part 1:

"How to make assumptions and simplify problems."

Example: Topic Modeling.

(Unsupervised Method for uncovering thematic structure in a corpus of documents.)

Goal: Algorithm that runs (under clearly specified conditions on input) in time polynomial in all relevant parameters, and produces solution of specified quality/accuracy.

“Bag of words” Assumption in Text Analysis

=

=

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

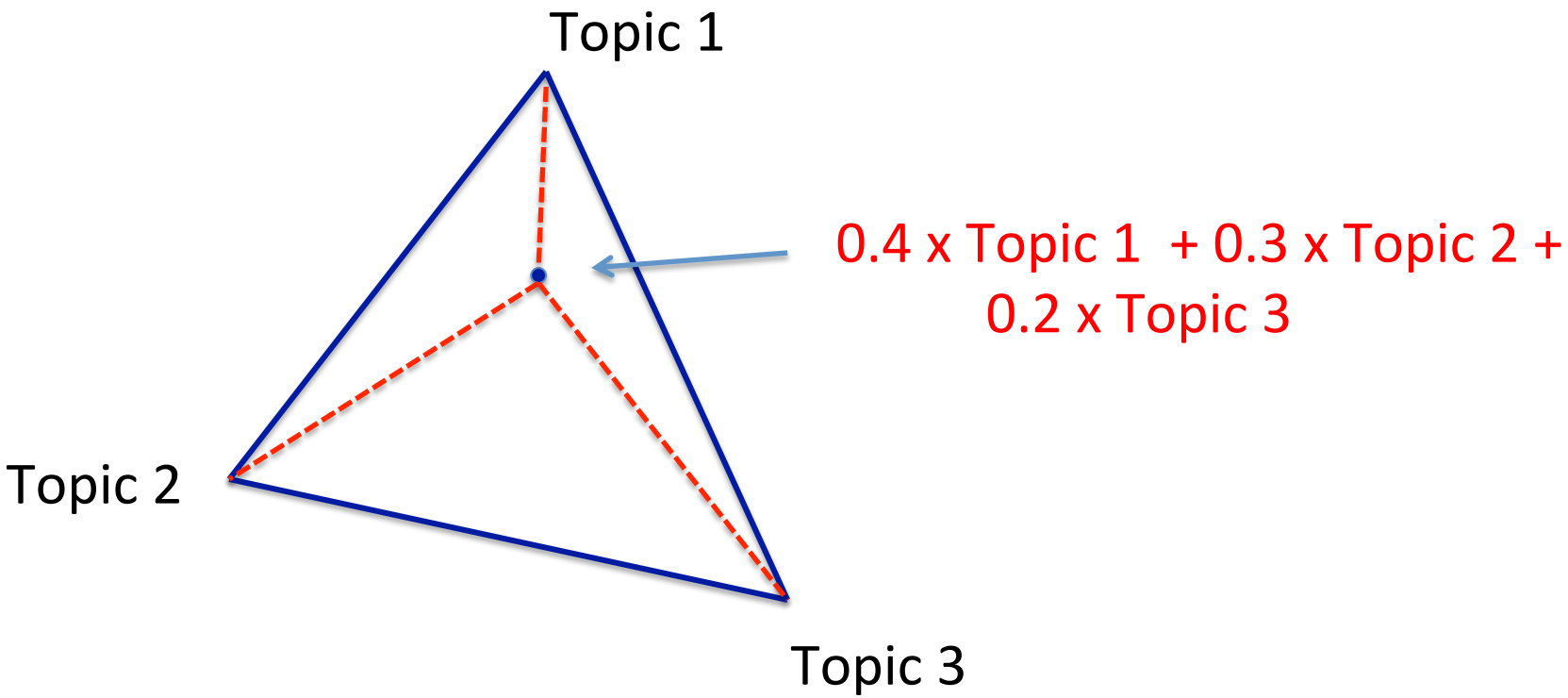
Document Corpus = Matrix
(i^{th} column = i^{th} document)

Hidden Variable Explanation

- Document = Mixture of Topics

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

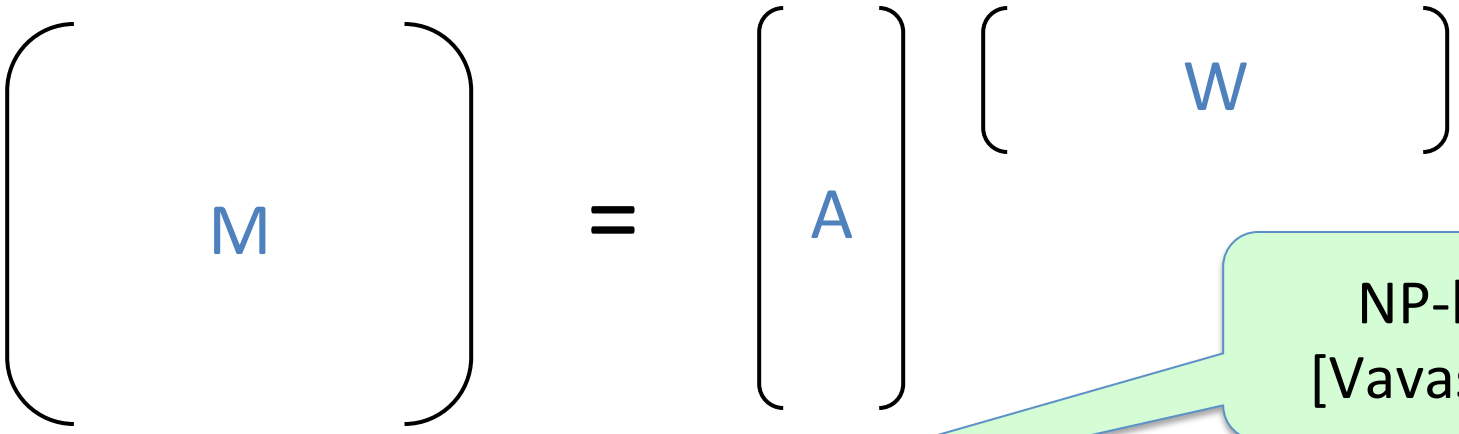
Hidden Variable explanation (geometric view)



Nonnegative Matrix Factorization (NMF)

[Lee Seung'99]

Given: Nonnegative n x m matrix M (all entries ≥ 0)



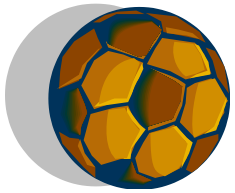
NP-hard
[Vavasis 09]

Want: Nonnegative matrices A (n x r) and W (r x m),
s.t. M = AW. (Aside: Given W, easy to find A via linear programming.)

Applications: Image Segmentation, Info Retrieval, Collaborative filtering, document classification.

"Separable" Topic Matrices



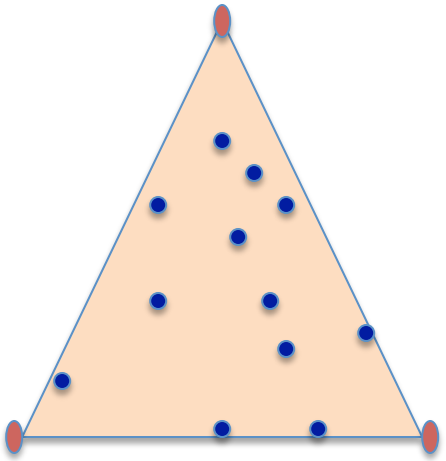


Banana	0	0	
Snow	4%	0	0 0
Soccer	0	8%	
Walnut	0	0	

Geometric restatement of NMF

(after some trivial rescaling)

Given n nonnegative vectors
(namely, rows of M)



Find r-dimensional simplex
with nonnegative vertices
that contains all.

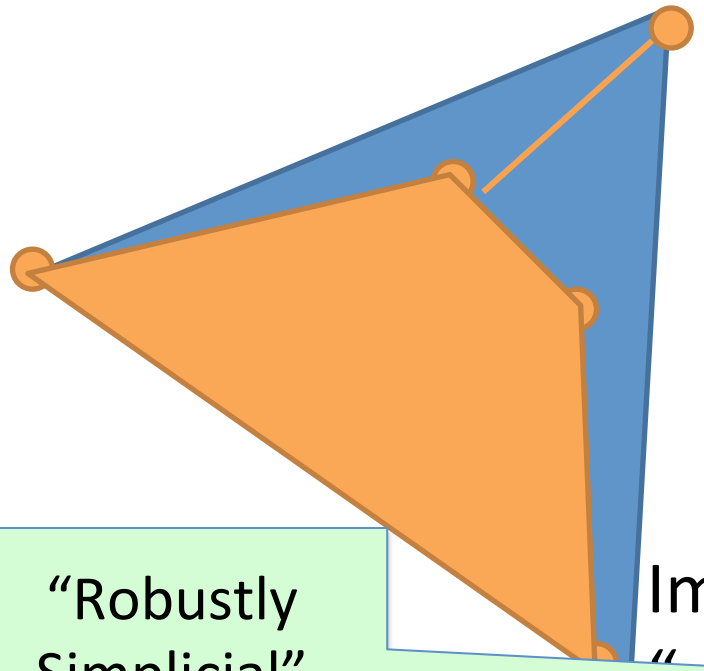
(rows of W = vertices of this simplex;
Rows of A = convex combinations)

Separable → Vertices of simplex appear among rows of M

Finding Separable Factorization

[A,Ge, Kannan, Moitra STOC'12]

- Algorithm: Remove a row, test if it is in the convex hull of other rows



- Case 1: Inside Row
- Can be represented by other rows

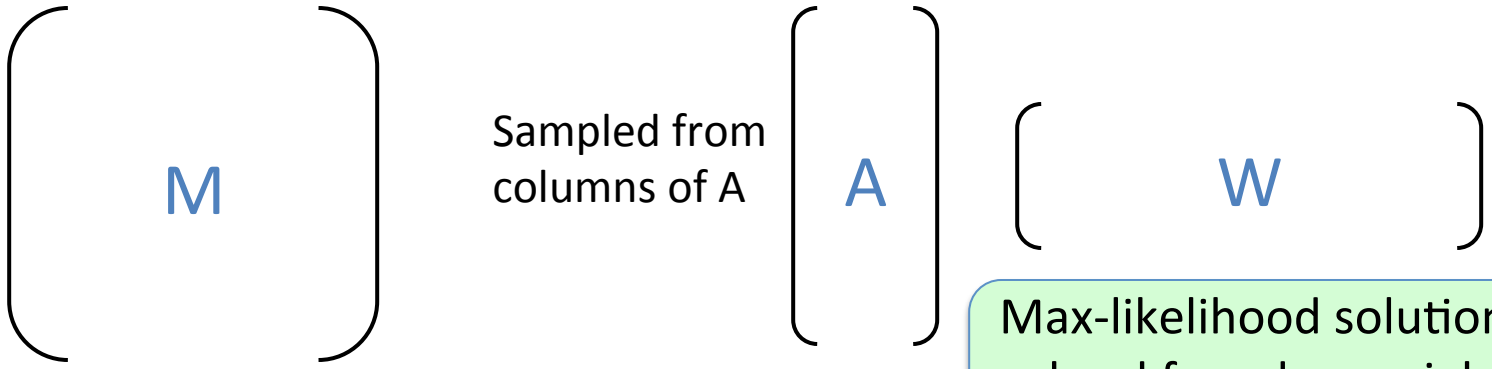
- Case 2: Row at a vertex
- Cannot be represented by other rows

"Robustly Simplicial"

Important: Procedure can tolerate "noisy data" if simplex "not too flat."

Learning Topic Models

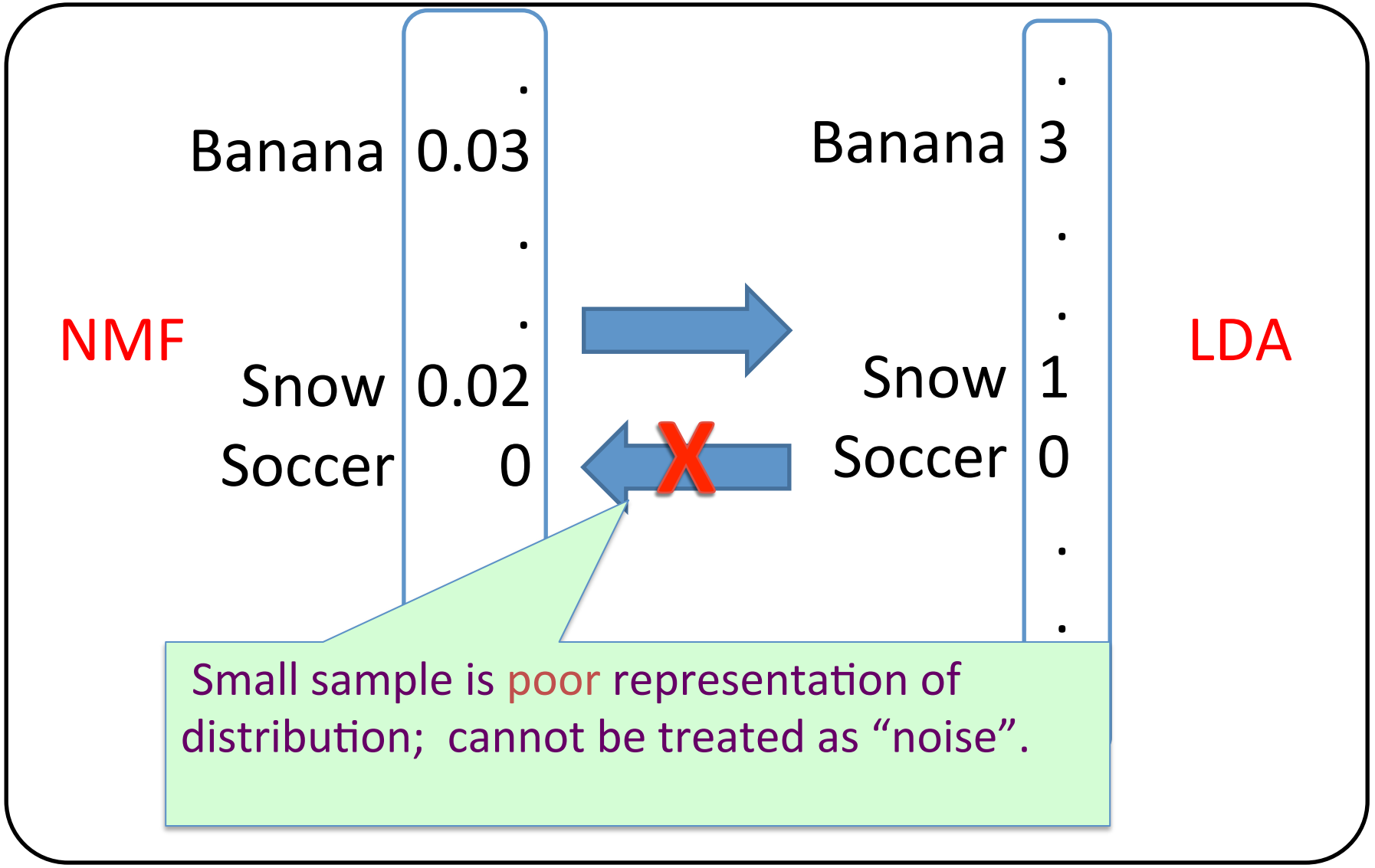
[Papadimitriou et al.'98, Hoffman'99, Blei et al.'03]



Max-likelihood solution is NP-hard for adversarial data, even for r=2 (AGM'12)

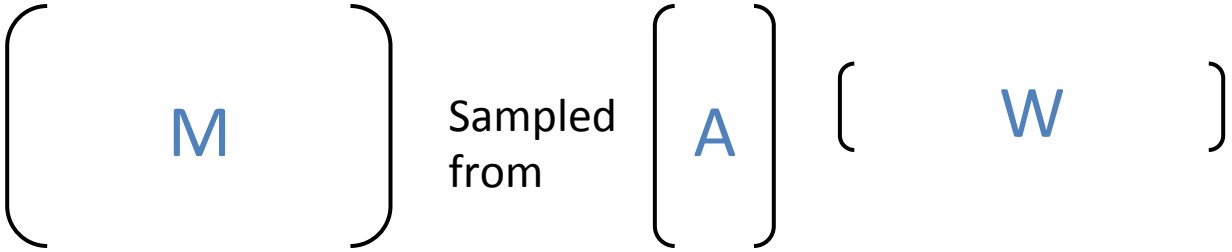
- Topic matrix A (n x r) arbitrary, nonnegati
  Stochastic W (r x m). Columns iid from un vn distrib.
- Given: M (n x m). i^{th} column has 100 sam es
  from distribution given by i^{th} column of W.
- Goal: Find A and parameters of distribution that generated W.
- Popular choice of distribution: Dirichlet. ("LDA" Blei, Jordan, Ng '03.)

The main difficulty (why LDA learning ≠ NMF)



Reducing topic modeling to NMF

[A, Ge , Moitra FOCS'12]



Word-word co-occurence matrix = MM^T (2^{nd} Moment)

$\approx AWW^TA^T$ (up to scaling)

$= AW_{new}$ where

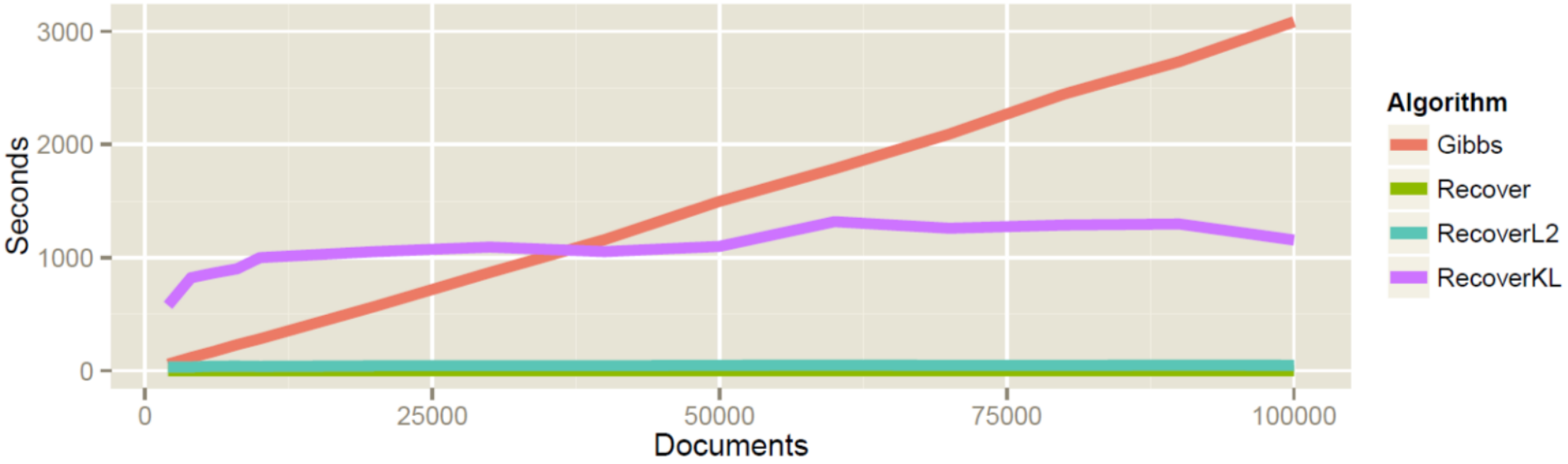
$W_{new} = WW^TA^T$

Can factorize using noise tolerant NMF algorithm!

Important: Need for separability assumption removed by [Anandkumar, Hsu, Kakade'12] (slower algorithm).

Empirical Results

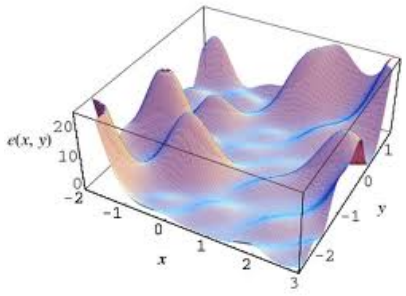
[A, Ge, Halpern, Mimno, Moitra, Sontag, Wu, Zhu ICML'13]



- 50x faster on realistic data sizes.
- Comparable error on synthetic data
- Similar quality scores on real-life data (NYT corpus).
- Works better than theory can explain.

Part 2:

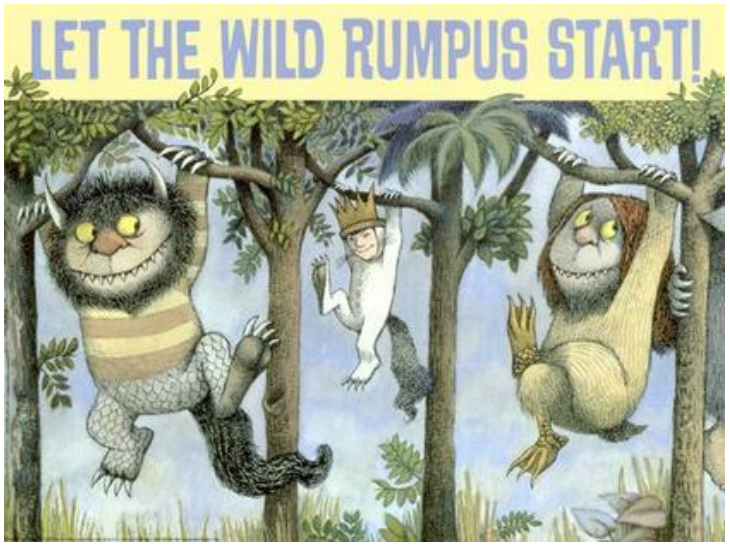
"The unreasonable effectiveness of nonconvex heuristics."



Branch & Bound for integer programming,
DPLL-family for SAT solving/Software verification.
Markov Chain Monte Carlo for counting problems (#P),
Belief propagation for Bayes Nets,..

ML : Great setting to study heuristics

- Clean models of how data was generated
- Heuristics so "natural" that even natural systems use them (e.g., neurons).
- Theorists understand hardness; hence well-equipped to identify assumptions that provably simplify the problem.



Example 1: Dictionary Learning
(aka Sparse Coding)

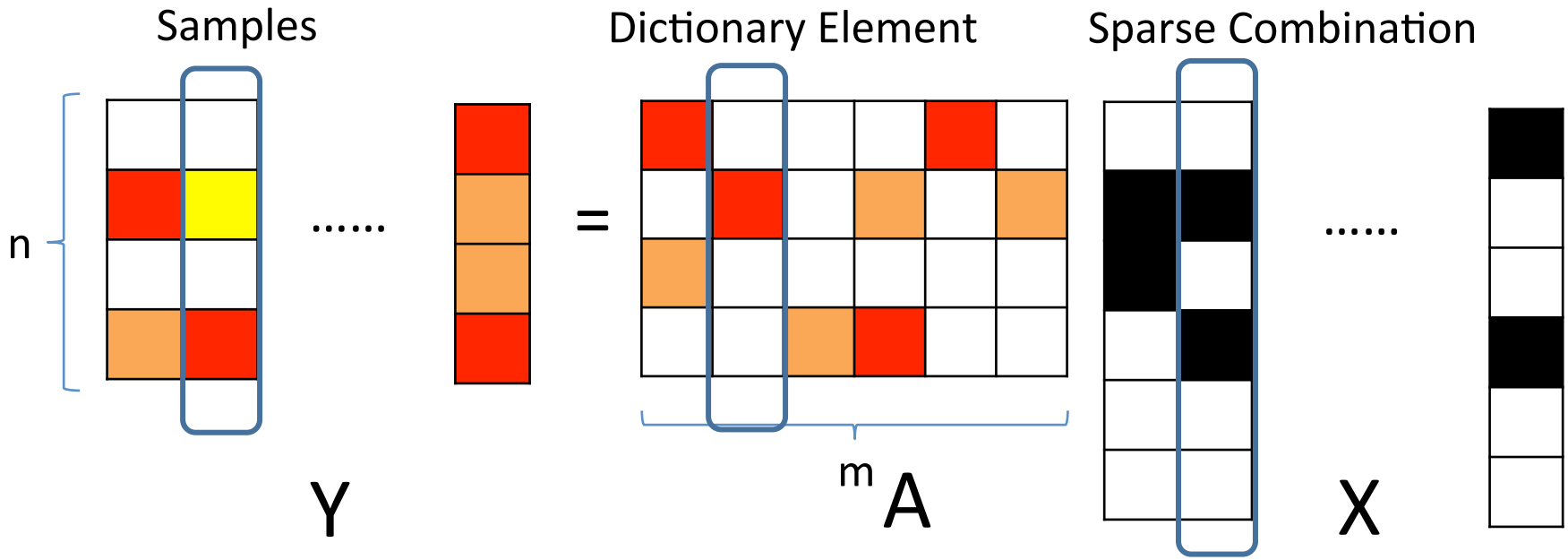
- Simple "dictionary elements" build complicated objects.

- Each object composed of small # of dictionary elements (sparsity assumption)

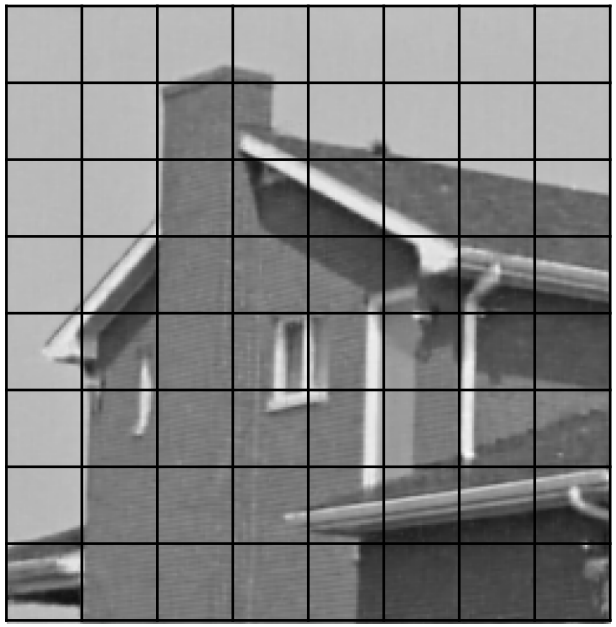
- Given the objects, can we learn the dictionary?

Dictionary Learning: Formalization

- Given samples of the form Y = AX
- X is unknown matrix with sparse columns; m X S
- A (dictionary): n x m, unknown. Has to be learnt
- Interesting case: m > n (overcomplete)
- Assumption: Columns of X iid from suitable distrib.

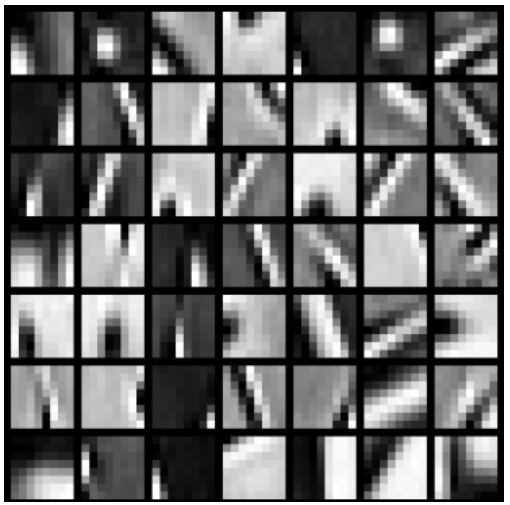


Why dictionary learning? [Olshausen Field '96]

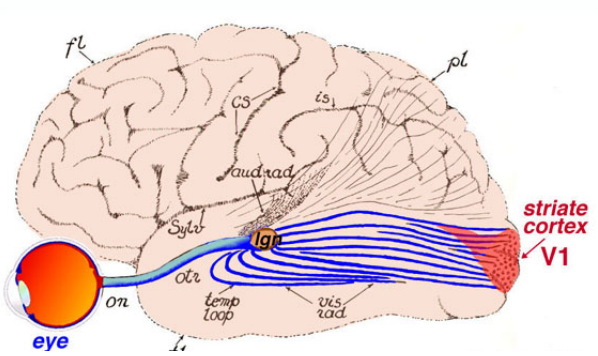


natural image patches

dictionary learning



Gabor-like Filters

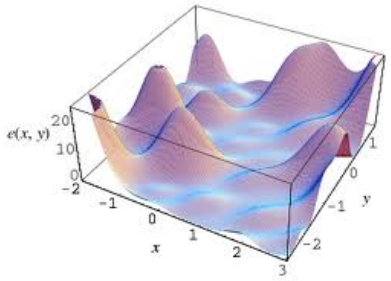


Other uses: Image Denoising,
Compression, etc.
Good example of "neural algorithm"

"Energy minimization" heuristic

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

$$x_i\text{'s are } k\text{-sparse}$$



- Nonconvex; heuristics use approximate gradient descent ("neural" algorithm)

[A., Ge,Ma,Moitra'14] Finds approx. global optimum in poly time. (updates will steadily decrease distance to optimum)

Assumptions:

- unknown A is "incoherent" (columns have low pairwise inner product) and has low matrix norm.
- X has pairwise indep. coordinates; is √n-sparse.

Builds upon recent progress in Dictionary Learning

- Poly-time algorithm when dictionary is full-rank (m =n); sparsity of X < √n. (Uses LP; not noise-tolerant)
  [Spielman, Wang, Wright, COLT'12]

- Polytime algorithm for overcomplete case (m > n) .
  A has to be "incoherent;" sparsity << √n
  [A., Ge, Moitra'13], [Agarwal, Anankumar, Netrapalli'13]

- New algorithms that allow almost-dense X
  [A., Bhaskara, Ge, Ma'14], [Barak, Kelner, Steurer'14]

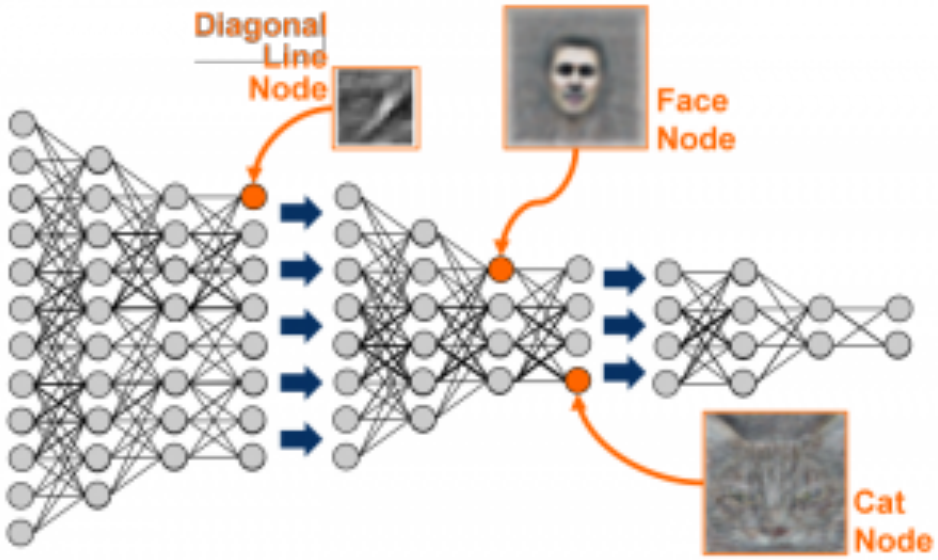
- Alternating minimization works in poly time.
  [A., Ge, Ma, Moitra '14]

Crucial idea in all: Stability of SVD/PCA; allows digging for "signal"

Example 2: Deep Nets

Deep learning: learn multilevel representation of data (nonlinear)

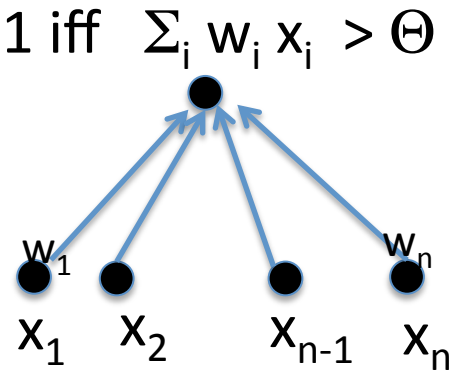
(inspired e.g. by 7-8 levels of visual cortex)



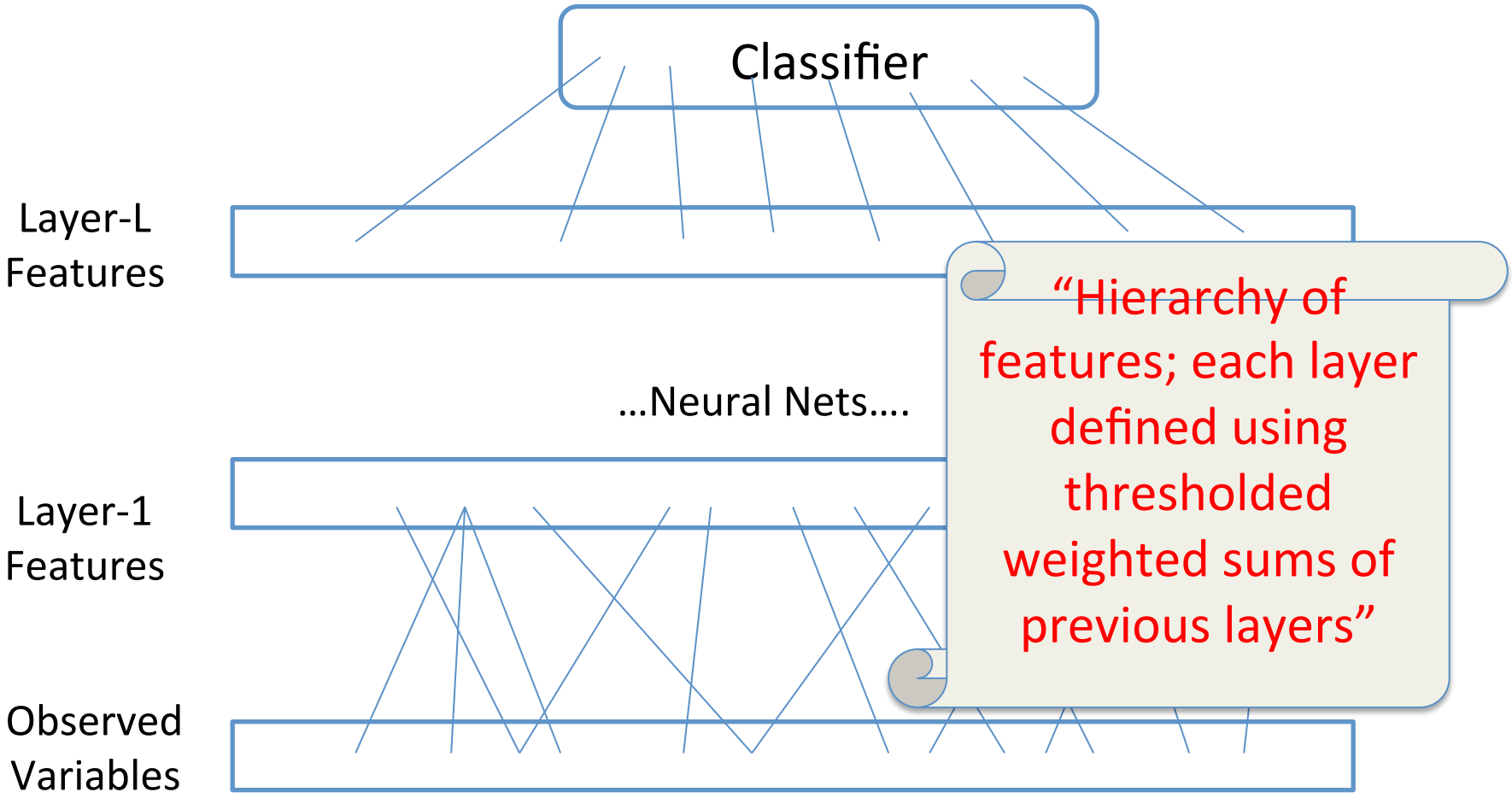
Successes: speech recognition, image recognition, etc.

[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])



Deep Nets at a glance



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]

Part 3:

"Linear Algebra++"

Mathematical heart of these ML problems (extends classical Linear Algebra, problems usually NP-hard)

Classical linear algebra

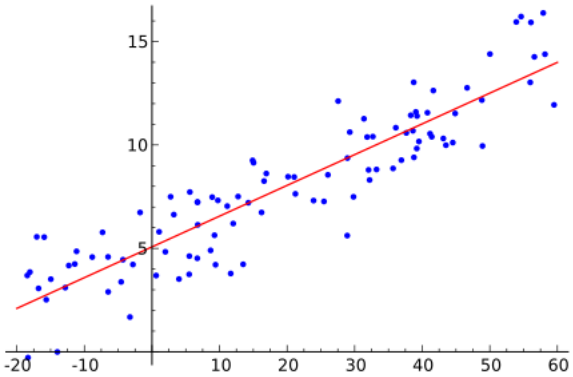
- Solving linear systems: Ax =b

- 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/ noisy 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")

Linear Algebra ++

Classical linear algebra together with any subset of following type of constraints:

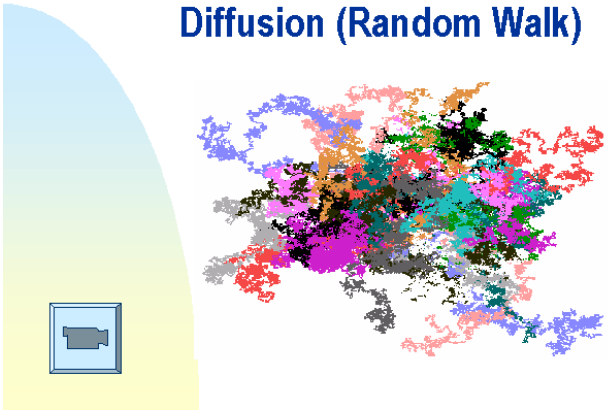
- Nonnegativity of variables $x \geq 0$
- Sparsity (at most k variables are nonzero)
- Correct for observation "noise" (least squares, deletion,)

Most are NP-hard; making progress requires making assumptions

Part 4:

Constructing theories that explain observations

Example:



Random walks on semantic spaces:
Towards an explanation of mysteries of semantic word embeddings.
(A., Li, Liang, Ma, Risteski arxiv'15)

(explains unsupervised methods for solving analogies like man: woman :: king : ??)

Concluding Thoughts

- Can circumvent intractability by novel assumptions between avg case and worst case): e.g., separability; randomly wired neural nets, etc.

- Thinking of provable bounds often leads to new kinds of algorithms. (Sometimes can analyse existing heuristics ..)

- Algorithms with provable bounds can be practical, or give new insights.

- Time to rethink ugrad/grad algorithms courses?
  An attempt: http://www.cs.princeton.edu/courses/archive/fall14/cos521/

THANK YOU

Matrix factorization: Nonlinear variants

Given M produced as follows: Generate low-rank A, B, apply nonlinear function 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]

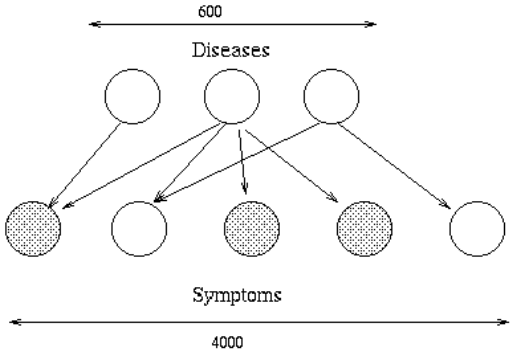
Part 4:

"Some favorite open problems/research directions"

Inference via Bayes Nets [Pearl'88]

Your symptoms: fever + red spots.

Probability that you have measles??



Bayes net succinctly describes
Pr[symptom| diseases d_1, d_2,..]

Desired: Posterior
Pr[disease| symptom s_1, s_2,..]

#P-complete, currently estimated via heuristics (MCMC, Variational Inf., Message Propagation..)

Realistic assumptions that simplify?

Provable versions of Variational Inference?

(reference: [Jaakola, Jordan] survey)

Very general setting: Prob. Distribution p(x, z)
(explicit formula)

z is observed. Estimate Bayesian Posterior p(x|z) (#P-hard!)

Method: Hypothesize simple functional form q(x, v) where
v is a small set of "variational parameters."
(akin to Mean Field Assumption from statistical physics)

Minimize KL(q(x, v) || p(x|z)) using a series of "simplifications"

Suggested 1st step: Analyse V.I. in settings where we already have provable algorithms: Topic Models, Dictionary Learning, HMMs etc.