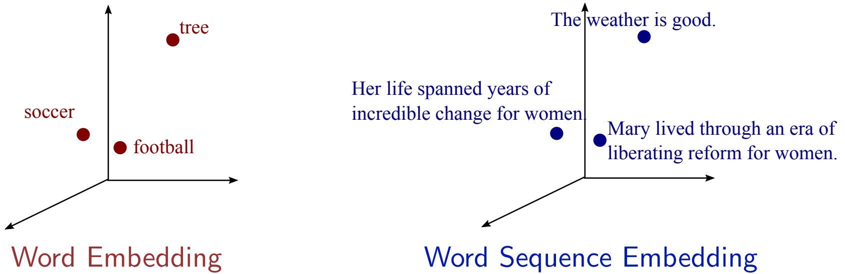
Learning Sentence Embeddings through Tensor Methods

Anima Anandkumar

Joint work with Dr. Furong Huang

ACL Workshop 2016

Representations for Text Understanding



- Word embeddings: Incorporates short range relationships, Easy to train.
- Sentence embeddings: Incorporates long range relationships, hard to train.

Various Frameworks for Sentence Embeddings

Compositional Models (M. Iyyer etal ‘15, T. Kenter ‘16)

- Composition of word embedding vectors: usually simple averaging.
- Compositional operator (averaging weights) based on neural nets.
- Weakly supervised (only averaging weights based on labels) or strongly supervised (joint training).

Paragraph Vector (Q. V. Le & T. Mikolov ‘14)

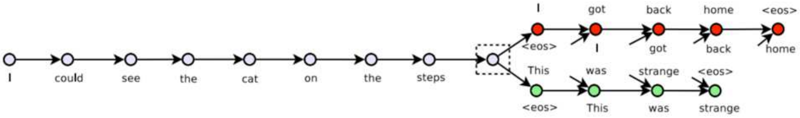
- Augmented representation of paragraph + word embeddings.
- Supervised framework to train paragraph vector.

For both frameworks

- Pros: Simple and cheap to train. Can use existing word embeddings.
- Cons: Word order not incorporated. Supervised. Not universal.

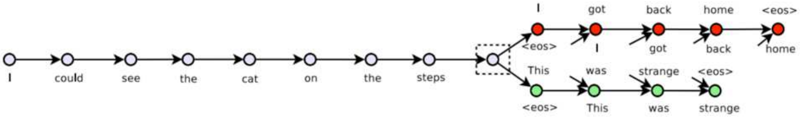
Skip thought Vectors for Sentence Embeddings

- Learn sentence embedding based on joint probability of words, represented using RNN.



Skip thought Vectors for Sentence Embeddings

- Learn sentence embedding based on joint probability of words, represented using RNN.

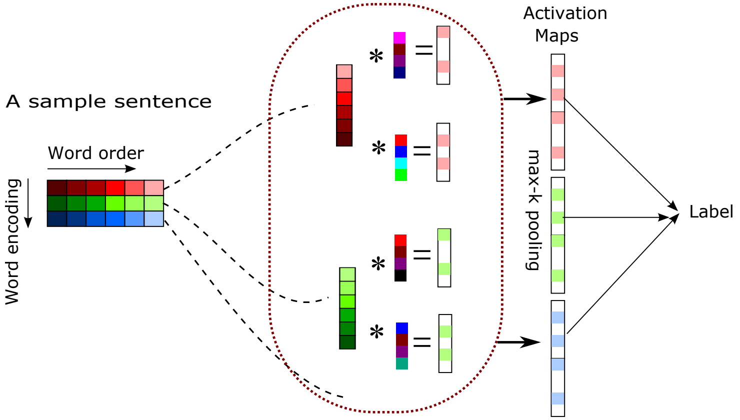


- Pros: Incorporates word order, unsupervised, universal.
- Cons: Requires contiguous long text, lots of data, slow training time. Cannot use domain specific training.

R. Kiros, Y. Zhu, R. Salakhutdinov, R. Zemel, A. Torralba, R. Urtasun, S. Fidler, " Skip-Thought Vectors, " NIPS 2015

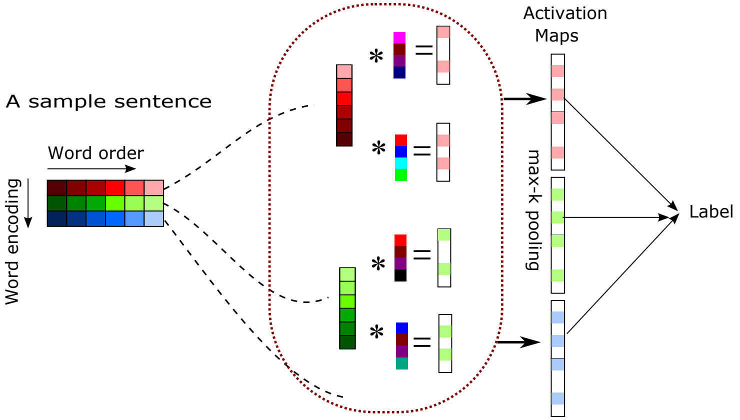
Convolutional Models for Sentence Embeddings

(N. Kalchbrenner, E. Grefenstette, P. Blunsom '14)



Convolutional Models for Sentence Embeddings

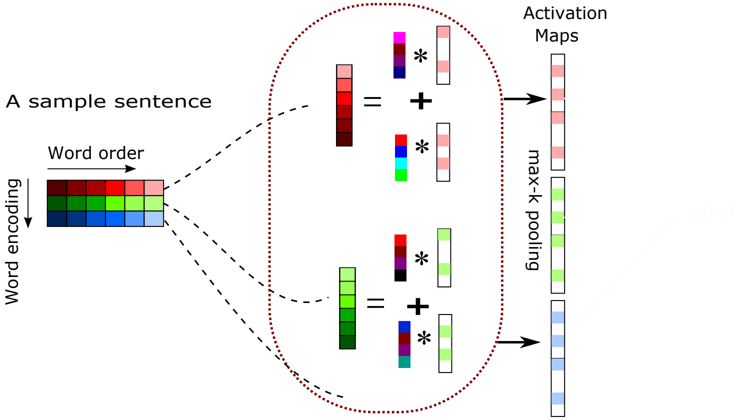
(N. Kalchbrenner, E. Grefenstette, P. Blunsom '14)



- Pros: Incorporates word order. Detect polysemy.
- Cons: Supervised training. Not universal.

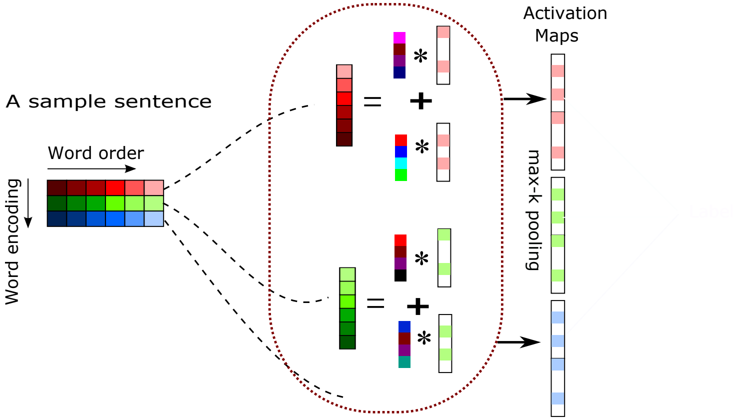
Convolutional Models for Sentence Embeddings

(F. Huang & A. '15)



Convolutional Models for Sentence Embeddings

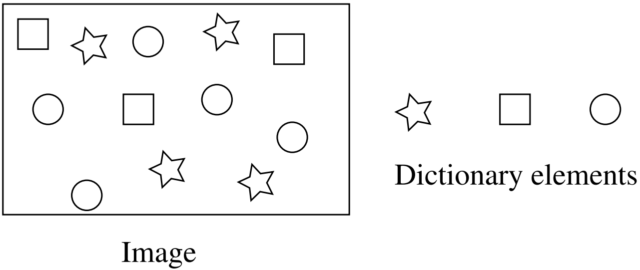
(F. Huang & A. '15)



- Pros: Word order, polysemy, unsupervised, universal.
- Cons: Difficulty in training.

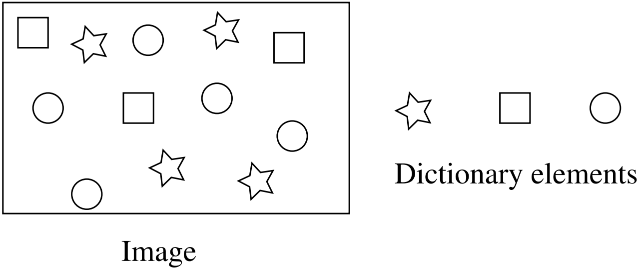
Intuition behind Convolutional Model

- Shift invariance natural in images: image templates in different locations.



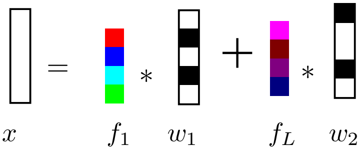
Intuition behind Convolutional Model

- Shift invariance natural in images: image templates in different locations.



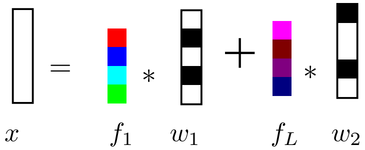
- Shift invariance in language: phrase templates in different parts of the sentence

Learning Convolutional Dictionary Models



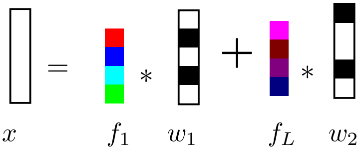
- Input x, phrase templates (filters) f_1, f_2, activations w_1, w_2

Learning Convolutional Dictionary Models



- Input x, phrase templates (filters) f_1, f_2, activations w_1, w_2
- Training objective: $\min_{f_i, w_i} \|x - \sum_i f_i * w_i\|_2^2$

Learning Convolutional Dictionary Models



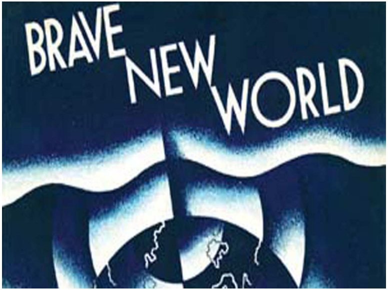
- Input x, phrase templates (filters) f_1, f_2, activations w_1, w_2
- Training objective: $\min\limits_{f_i, w_i} \|x - \sum\limits_i f_i * w_i\|_2^2$

Challenges

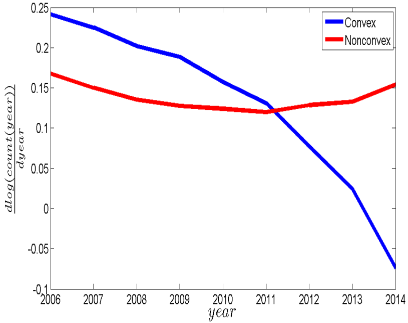
- Nonconvex optimization: no guaranteed solution in general.
- Alternating minimization: Fix w_i's to update f_i's and viceversa.
- Not guaranteed to reach global optimum (or even a stationary point!)
- Expensive in large sample regime: needs updating of w_i's.

Convex vs. Non-convex Optimization

Guarantees for mostly convex..

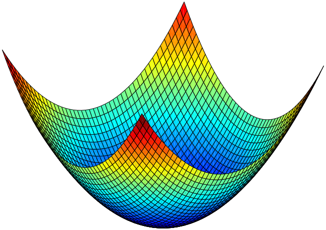


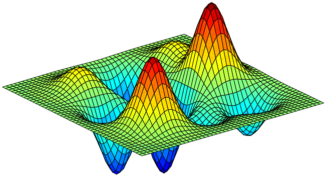
But non-convex is trending!



Images taken from `https://www.facebook.com/nonconvex`

Convex vs. Nonconvex Optimization





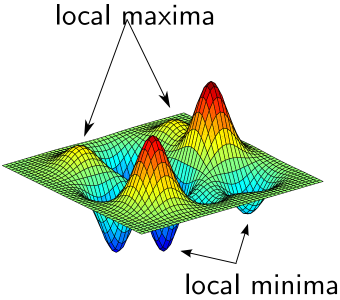
- Unique optimum: global/local.
- Multiple local optima

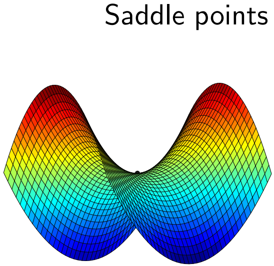
Guaranteed approaches for reaching global optima?

Non-convex Optimization in High Dimensions

Critical/statitionary points: $\boxed{x : \nabla_x f(x) = 0.}$

- Curse of dimensionality: exponential number of critical points.
- Saddle points slow down improvement.
- Lack of stopping criteria for local search methods.

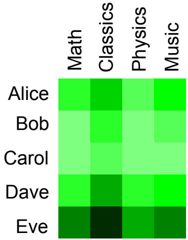




Fast escape from saddle points in high dimensions?

Outline

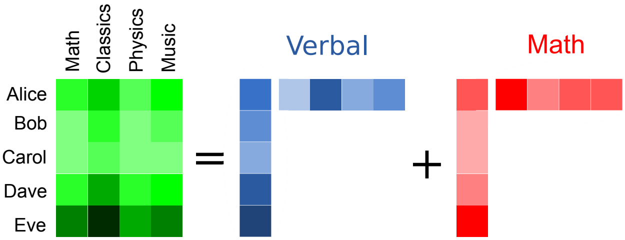
Example: Discovering Latent Factors



- List of scores for students in different tests
- Learn hidden factors for Verbal and Mathematical Intelligence [C. Spearman 1904]

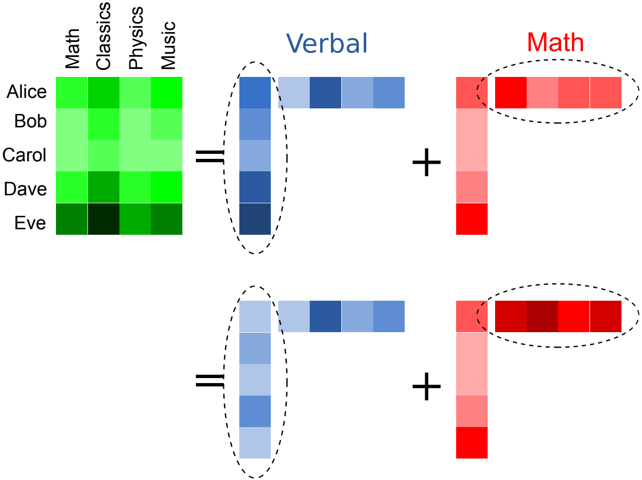
$$\text{Score (student,test)} = \text{student}_{\text{verbal-intlg}} \times \text{test}_{\text{verbal}} + \text{student}_{\text{math-intlg}} \times \text{test}_{\text{math}}$$

Matrix Decomposition: Discovering Latent Factors



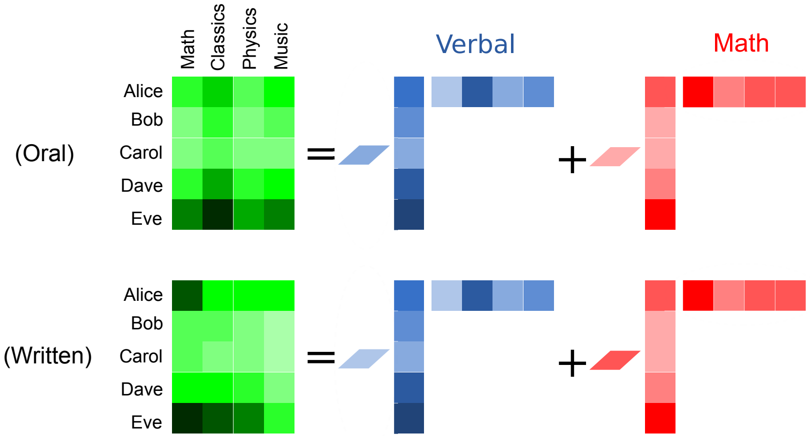
- Identifying hidden factors influencing the observations
- Characterized as matrix decomposition

Matrix Decomposition: Discovering Latent Factors



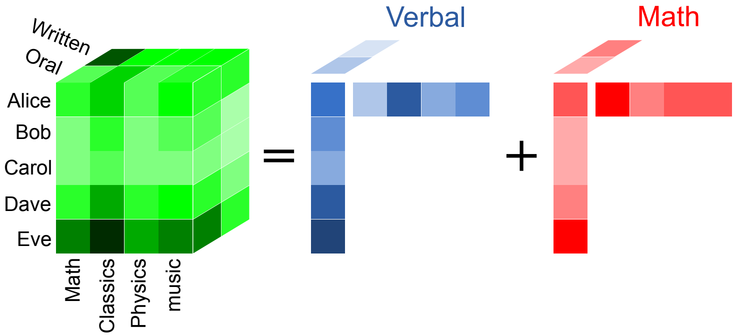
- Decomposition is not necessarily unique.
- Decomposition cannot be overcomplete.

Tensor: Shared Matrix Decomposition



- Shared decomposition with different scaling factors
- Combine matrix slices as a tensor

Tensor Decomposition



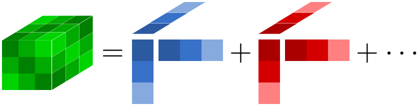
- Outer product notation:

$$T = u \otimes v \otimes w + \tilde{u} \otimes \tilde{v} \otimes \tilde{w}$$

$$\Updownarrow$$

$$T_{i_1,i_2,i_3} = u_{i_1} \cdot v_{i_2} \cdot w_{i_3} + \tilde{u}_{i_1} \cdot \tilde{v}_{i_2} \cdot \tilde{w}_{i_3}$$

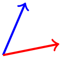
Identifiability under Tensor Decomposition



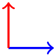
$$T = v_1^{\otimes 3} + v_2^{\otimes 3} + \cdots ,$$

Uniqueness of Tensor Decomposition [J. Kruskal 1977]

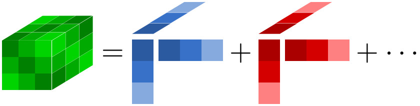
- Above tensor decomposition: unique when rank one pairs are linearly independent



- Matrix case: when rank one pairs are orthogonal



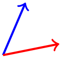
Identifiability under Tensor Decomposition



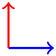
$$T = v_1^{\otimes 3} + v_2^{\otimes 3} + \cdots ,$$

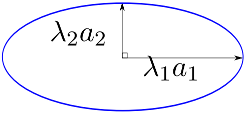
Uniqueness of Tensor Decomposition [J. Kruskal 1977]

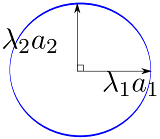
- Above tensor decomposition: unique when rank one pairs are linearly independent

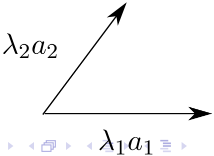


- Matrix case: when rank one pairs are orthogonal

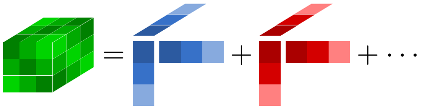








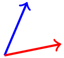
Identifiability under Tensor Decomposition



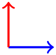
$$T = v_1^{\otimes 3} + v_2^{\otimes 3} + \cdots,$$

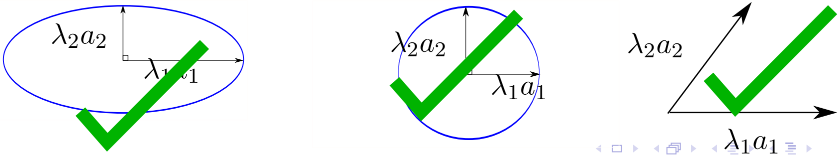
Uniqueness of Tensor Decomposition [J. Kruskal 1977]

- Above tensor decomposition: unique when rank one pairs are linearly independent



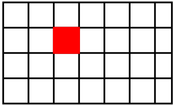
- Matrix case: when rank one pairs are orthogonal





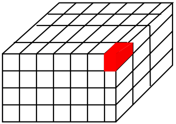
Moment-based Estimation

Matrix: Pairwise Moments



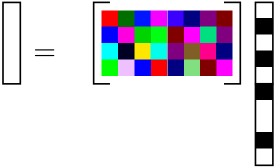
- $\mathbb{E}[x \otimes x] \in \mathbb{R}^{d \times d}$ is a second order tensor.
- $\mathbb{E}[x \otimes x]_{i_1,i_2} = \mathbb{E}[x_{i_1} x_{i_2}]$.
- For matrices: $\mathbb{E}[x \otimes x] = \mathbb{E}[x x^\top]$.
- $M = u u^\top$ is rank-1 and $M_{i,j} = u_i u_j$.

Tensor: Higher order Moments

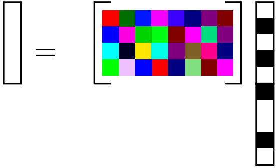


- $\mathbb{E}[x \otimes x \otimes x] \in \mathbb{R}^{d \times d \times d}$ is a third order tensor.
- $\mathbb{E}[x \otimes x \otimes x]_{i_1,i_2,i_3} = \mathbb{E}[x_{i_1} x_{i_2} x_{i_3}]$.
- $T = u \otimes u \otimes u$ is rank-1 and $T_{i,j,k} = u_i u_j u_k$.

Moment forms for Linear Dictionary Models



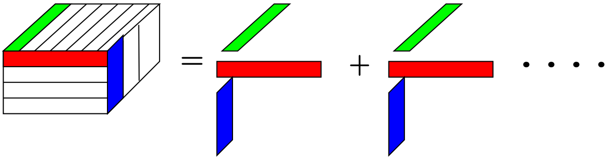
Moment forms for Linear Dictionary Models



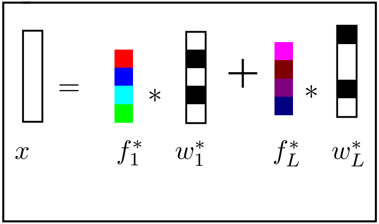
Independent components analysis (ICA)

- Independent coefficients, e.g. Bernoulli Gaussian.
- Can be relaxed to sparse coefficients with limited dependency.

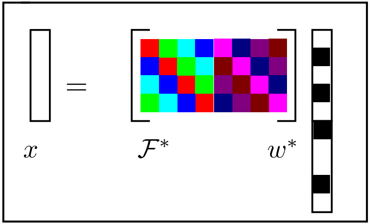
Fourth order cumulant: $M_4 = \sum_{j \in [k]} \kappa_j a_j \otimes a_j \otimes a_j \otimes a_j.$



Convolutional dictionary model



(a) Convolutional model



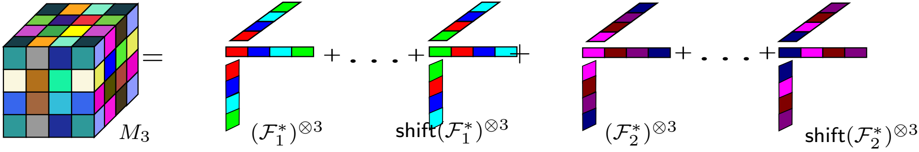
(b) Reformulated model

$$x = \sum_i f_i * w_i = \sum_i \mathsf{Cir}(f_i) w_i = \mathcal{F}^* w^*$$

Moment forms and optimization

$$x = \sum_i f_i * w_i = \sum_i \mathsf{Cir}(f_i) w_i = \mathcal{F}^* w^*$$

- Assume coefficients w_i are independent (convolutional ICA model)
- Cumulant tensor has decomposition with components $\mathcal{F}_i^*$.



Learning Convolutional model through Tensor Decomposition

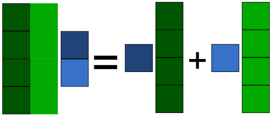
Outline

Notion of Tensor Contraction

Extends the notion of matrix product

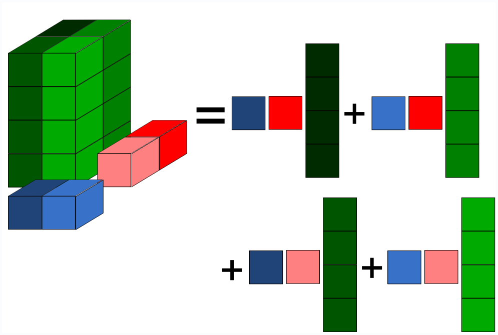
Matrix product

$$Mv = \sum_j v_j M_j$$



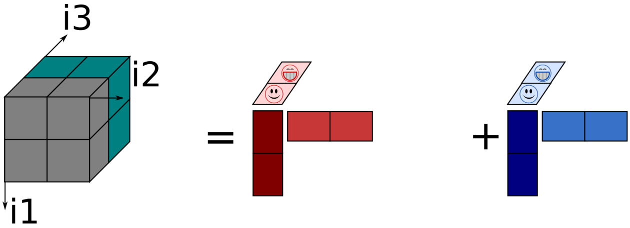
Tensor Contraction

$$T(u, v, \cdot) = \sum_{i,j} u_i v_j T_{i,j,:}$$



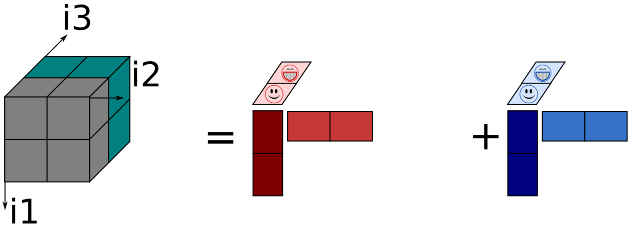
Tensor Decomposition - ALS

- Objective: $\|T - \sum_i a_i \otimes b_i \otimes c_i\|_2^2$



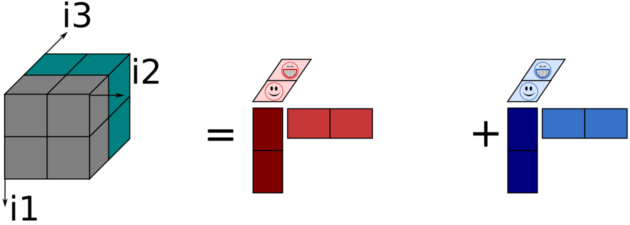
Tensor Decomposition - ALS

- Objective: $\|T - \sum_i a_i \otimes b_i \otimes c_i\|_2^2$
- Key observation: If b_i, c_i's are fixed, objective is linear in a_i's.



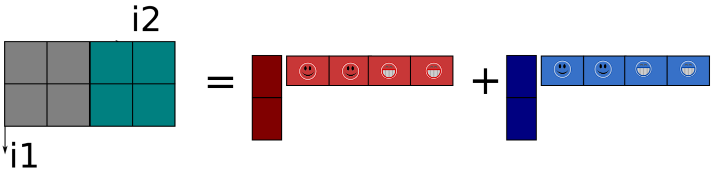
Tensor Decomposition - ALS

- Objective: $\|T - \sum_i a_i \otimes b_i \otimes c_i\|_2^2$
- Key observation: If b_i, c_i's are fixed, objective is linear in a_i's.
- Tensor unfolding



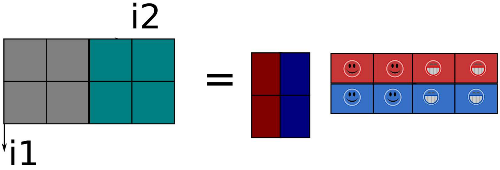
Tensor Decomposition - ALS

- Objective: $\|T - \sum_i a_i \otimes b_i \otimes c_i\|_2^2$
- Key observation: If b_i, c_i's are fixed, objective is linear in a_i's.
- Tensor unfolding



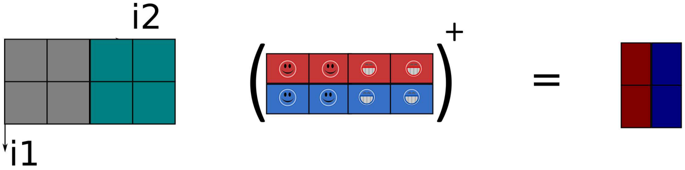
Tensor Decomposition - ALS

- Objective: $\|T - \sum_i a_i \otimes b_i \otimes c_i\|_2^2$
- Key observation: If b_i, c_i's are fixed, objective is linear in a_i's.
- Tensor unfolding



Tensor Decomposition - ALS

- Objective: $\|T - \sum_i a_i \otimes b_i \otimes c_i\|_2^2$
- Key observation: If b_i, c_i's are fixed, objective is linear in a_i's.
- Tensor unfolding



Convolutional Tensor Decomposition

- Objective: $\|T - \sum_i a_i \otimes a_i \otimes a_i\|_2^2$
- Constraint: $A := [a_1, a_2, \ldots]$ is concatenation of circulant matrices.

Convolutional Tensor Decomposition

- Objective: $\|T - \sum_i a_i \otimes a_i \otimes a_i\|_2^2$
- Constraint: $A := [a_1, a_2, \ldots]$ is concatenation of circulant matrices.

Modified Alternating Least Squares Method

- Project onto set of concatenated circulant matrices in each step.

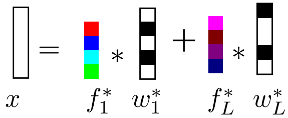
Convolutional Tensor Decomposition

- Objective: $\|T - \sum_i a_i \otimes a_i \otimes a_i\|_2^2$
- Constraint: $A := [a_1, a_2, \ldots]$ is concatenation of circulant matrices.

Modified Alternating Least Squares Method

- Project onto set of concatenated circulant matrices in each step.
- Our contribution: Efficient computation through FFT and blocking.

Comparison with Alternating Minimization



- L is the number of filters.
- n is the dimension of filters.
- N is the number of samples.

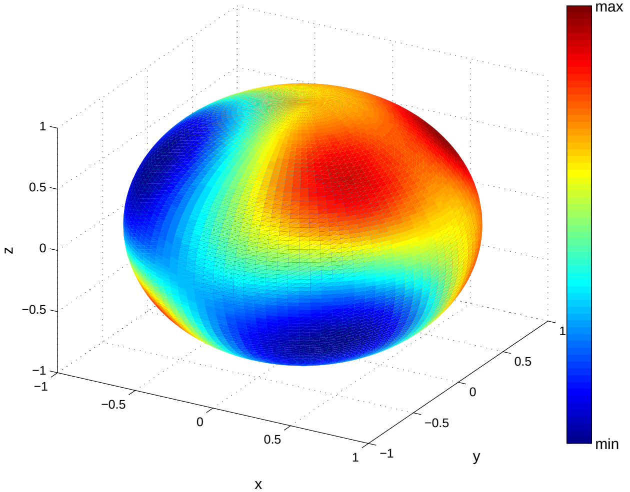
Computation complexity

Methods	Running Time	Processors
Tensor Factorization	$O(\log(n)+\log(L))$	$O(L^2\ n^3)$
Alt. Min	$O(\max(\log(n)\log(L),\ \log(n)\log(N)))$	$O(\max(\ \mathbf{N}nL, \mathbf{N}nL))$

Complexity for tensor method independent of sample size

Analysis

- Non-convex optimization: guaranteed convergence to local optimum
- Local optima are shifted filters



Experiments using Sentence Embeddings

Dataset	**Domain**	N
Review	Movie Reviews	64720
SUBJ	Obj/Subj comments	1000
MSRpara	news sources	5801×2
STS-MSRpar	newswire	1500×2
STS-MSRvid	video caption	1500×2
STS-OnWN	glosses	750×2
STS-SMTeuroparl	machine translation	1193×2
STS-SMTnews	machine translation	399×2

Experiments using Sentence Embeddings

Dataset	Domain	N
Review	Movie Reviews	64720
SUBJ	Obj/Subj comments	1000
MSRpara	news sources	5801×2
STS-MSRpar	newswire	1500×2
STS-MSRvid	video caption	1500×2
STS-OnWN	glosses	750×2
STS-SMTeuroparl	machine translation	1193×2
STS-SMTnews	machine translation	399×2

Sentiment Analysis

Method	MR	SUBJ
Paragraph-vector	74.8	90.5
Skip-thought	75.5	92.1
ConvDic+DeconvDec	**78.9**	**92.4**

- Paragraph vector weakly supervised.
- Skip thought and our method unsupervised

Paraphrase Detection Results

Method	Outside Information	F score
Vector Similarity	word similarity	75.3%
RMLMG	syntacticinfo	80.5%
ConvDic+DeconvDec	none	**80.7%**
Skip-thought	book corpus	81.9%

- Paraphrase detected: (1) Amrozi accused his brother, whom he called the witness, of deliberately distorting his evidence. (2) Referring to him as only the witness, Amrozi accused his brother of deliberately distorting his evidence.
- Non-paraphrase detected: (1) I never organised a youth camp for the diocese of Bendigo. (2) I never attended a youth camp organised by that diocese.

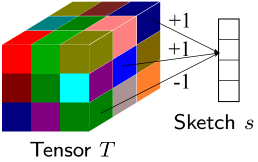
Semantic Textual Similarity Results

	Supervised				Unsupervised	
Dataset	DAN	RNN	LSTM	S-CBOW	Skip-thought	**Ours**
MSRpar	40.3	18.6	9.3	43.8	16.8	**36.0**
MSRvid	70.0	66.5	71.3	45.2	41.7	**61.8**
SMT-eur	43.8	40.9	44.3	45.0	35.2	**37.5**
OnWN	65.9	63.1	56.4	64.4	29.7	**33.1**
SMT-news	60.0	51.3	51.0	39.0	30.8	**72.1**

Outline

Tensor Sketches for Multilinear Representations

- Randomized dimensionality reduction through sketching.
  - Complexity independent of tensor order: exponential gain!

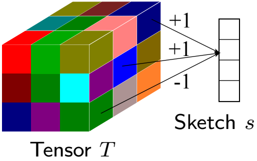


Wang, Tung, Smola, A. " Guaranteed Tensor Decomposition via Sketching", NIPS'15.

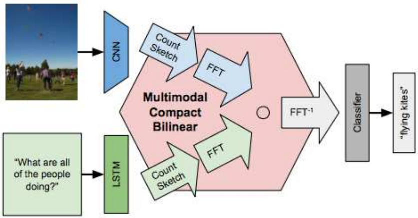
Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding by A. Fukui, D.H. Park, D. Yang, A. Rohrbach, T. Darrell, M. Rohrbach, CVPR 2016.

Tensor Sketches for Multilinear Representations

- Randomized dimensionality reduction through sketching.
  - Complexity independent of tensor order: exponential gain!



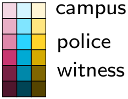
State of art results for visual Q & A



Wang, Tung, Smola, A. " Guaranteed Tensor Decomposition via Sketching", NIPS'15.

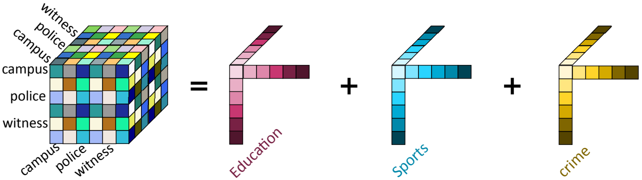
Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding by A. Fukui, D.H. Park, D. Yang, A. Rohrbach, T. Darrell, M. Rohrbach, CVPR 2016.

Tensor Methods for Topic Modeling



- Topic-word matrix $\boxed{\mathbb{P}[\text{word} = i | \text{topic} = j]}$
- Linearly independent columns

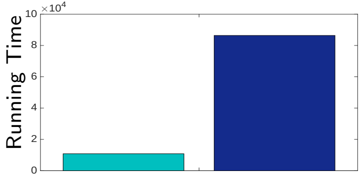
Moment Tensor: Co-occurrence of Word Triplets

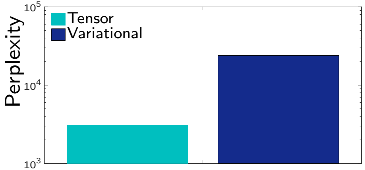


Tensors vs. Variational Inference

Criterion: Perplexity $= \exp[-\text{likelihood}]$.

Learning Topics from PubMed on Spark, 8mil articles



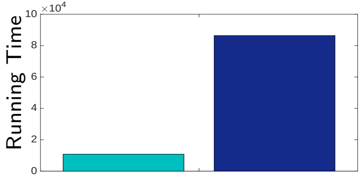


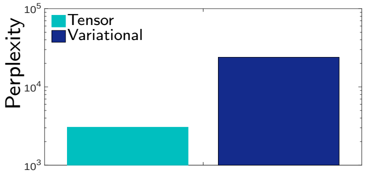
F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

Tensors vs. Variational Inference

Criterion: Perplexity $= \exp[-\text{likelihood}]$.

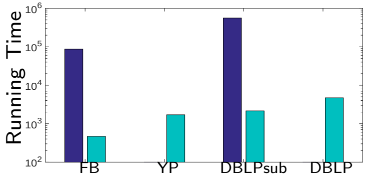
Learning Topics from PubMed on Spark, 8mil articles

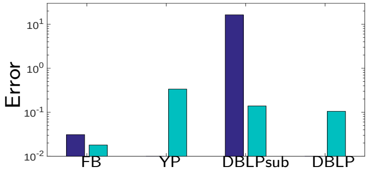




Learning network communities from social network data

Facebook $n \sim 20k$, Yelp $n \sim 40k$, DBLP-sub $n \sim 1e5$, DBLP $n \sim 1e6$.



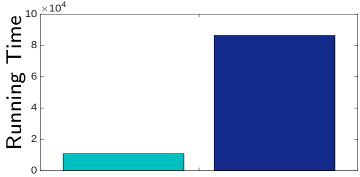


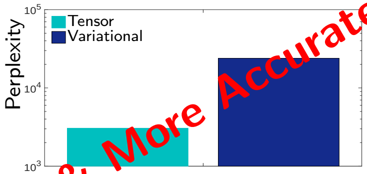
F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

Tensors vs. Variational Inference

Criterion: Perplexity $= \exp[-\text{likelihood}]$.

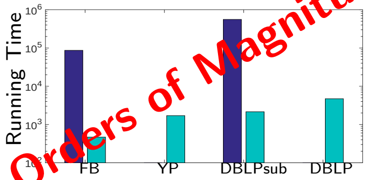
Learning Topics from PubMed on Spark, 8mil articles

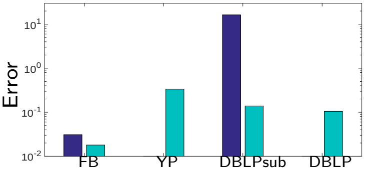




Learning network communities from social network data

Facebook $n \sim 20k$, Yelp $n \sim 40k$, DBLP-sub $n \sim 1e5$, DBLP $n \sim 1e6$.





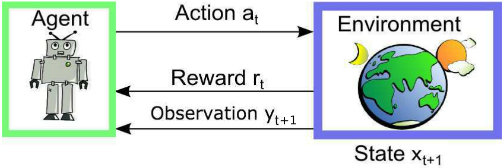
Orders of Magnitude Faster & More Accurate

F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

Reinforcement Learning of POMDPs

Reinforcement Learning

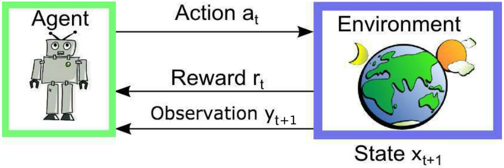
- Rewards from hidden state.
- Actions drive hidden state evolution.



Reinforcement Learning of POMDPs

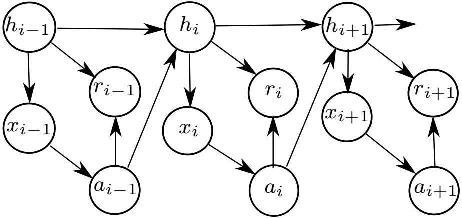
Reinforcement Learning

- Rewards from hidden state.
- Actions drive hidden state evolution.



Partially Observable Markov Decision Process

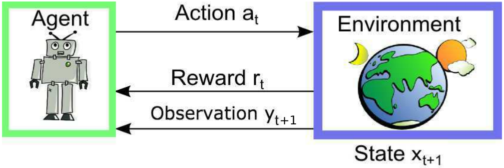
Learning using tensor methods under memoryless policies



Reinforcement Learning of POMDPs

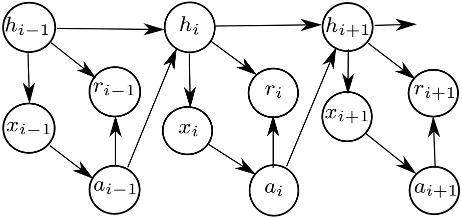
Reinforcement Learning

- Rewards from hidden state.
- Actions drive hidden state evolution.



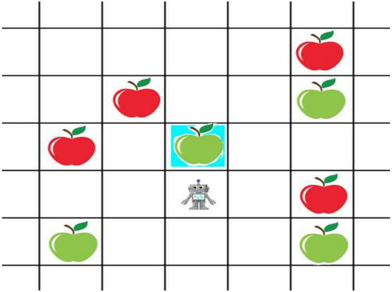
Partially Observable Markov Decision Process

Learning using tensor methods under memoryless policies

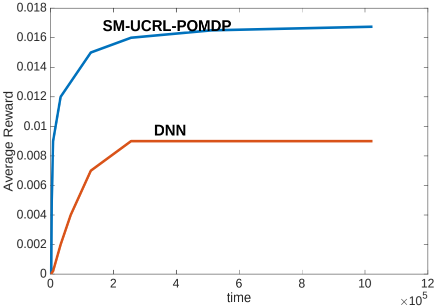


Reinforcement Learning of POMDPs

Gridworld game



Average Reward vs. Time.



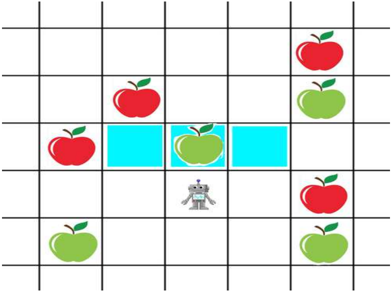
- POMDP model with 3 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 10 neurons each (trained using RmsProp).

K. Azzizade, Lazaric, A, Reinforcement Learning of POMDPs using Spectral Methods, COLT16.

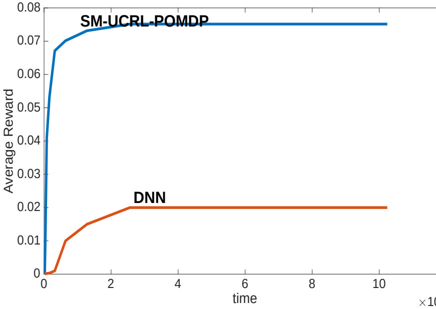
`http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html`

Reinforcement Learning of POMDPs

Observation Window



Average Reward vs. Time.



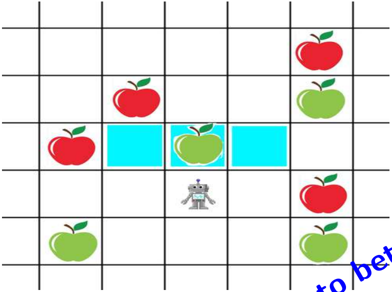
- POMDP model with 8 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 30 neurons each (trained using RmsProp).

K. Azzizade, Lazaric, A, Reinforcement Learning of POMDPs using Spectral Methods, COLT16.

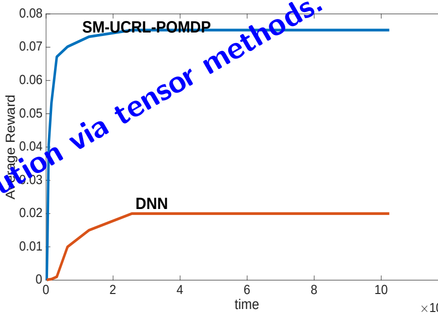
http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html

Reinforcement Learning of POMDPs

Observation Window



Average Reward vs. Time.



Faster convergence to better solution via tensor methods.

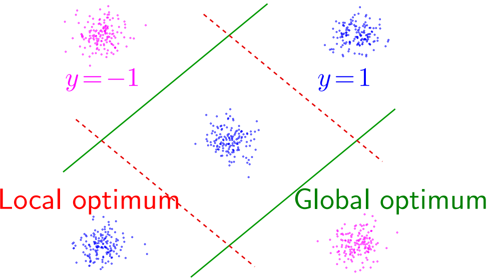
- POMDP model with 8 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 30 neurons each (trained using RmsProp).

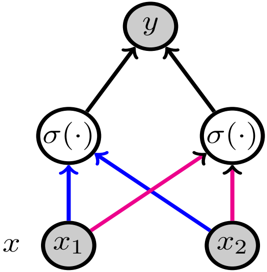
K. Azzizade, Lazaric, A, Reinforcement Learning of POMDPs using Spectral Methods, COLT16.

http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html

Local Optima in Backpropagation

"..few researchers dare to train their models from scratch.. small miscalibration of initial weights leads to vanishing or exploding gradients.. poor convergence..*"

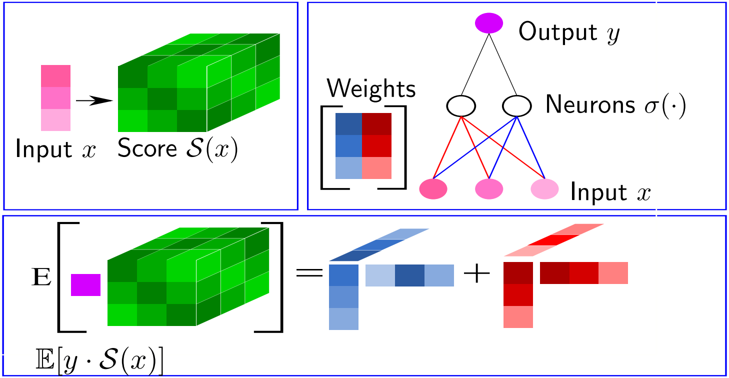




Exponential (in dimensions) no. of local optima for backpropagation

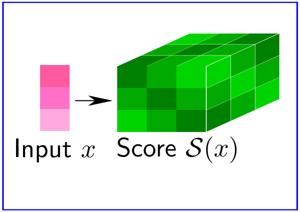
(*) P. Krahenbhl, C. Doersch, J. Donahue, T. Darrell "Data-dependent Initializations of Convolutional Neural Networks", ICLR 2016.

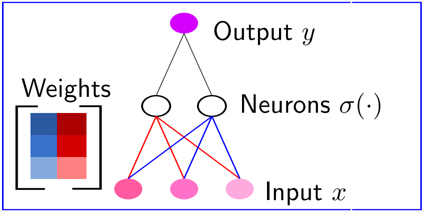
Training Neural Networks with Tensors

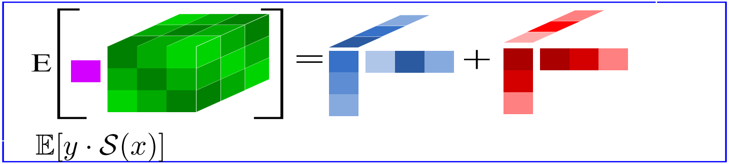


M. Janzamin, H. Sedghi, and A., "Beating the Perils of Non-Convexity: Guaranteed Training of Neural Networks using Tensor Methods," June. 2015.

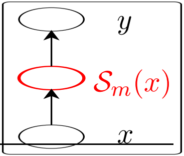
Training Neural Networks with Tensors







Given input pdf $p(\cdot)$, $\boxed{\mathcal{S}_m(x) := (-1)^m \frac{\nabla^{(m)} p(x)}{p(x)}}$.



Gaussian $x \Rightarrow$ Hermite polynomials.

M. Janzamin, H. Sedghi, and A., "Beating the Perils of Non-Convexity: Guaranteed Training of Neural Networks using Tensor Methods," June. 2015.

Outline

Conclusion

Unsupervised Convolutional Models for Sentence Embedding

- Desirable properties: incorporates word order, polysemy, universality.
- Efficient training through tensor methods.
- Faster and better performance in practice.

Conclusion

Unsupervised Convolutional Models for Sentence Embedding

- Desirable properties: incorporates word order, polysemy, universality.
- Efficient training through tensor methods.
- Faster and better performance in practice.

Steps Forward

- Universal embeddings using tensor methods on large corpus.
- More challenging setups: multilingual, multimodal (e.g. image and caption embeddings) etc.
- Bias-free embeddings? Can gender/race and other undesirable biases be avoided?

Research Connections and Resources

Collaborators

Rong Ge (Duke), Daniel Hsu (Columbia), Sham Kakade (UW), Jennifer Chayes, Christian Borgs, Alex Smola (CMU), Prateek Jain, Alekh Agarwal & Praneeth Netrapalli (MSR), Srinivas Turaga (Janelia), Allesandro Lazaric (Inria), Hossein Mobahi (Google).

- Podcast/lectures/papers/software available at `http://newport.eecs.uci.edu/anandkumar/`