Recent Advances in Non-Convex Optimization and its Implications to Learning

Anima Anandkumar

U.C. Irvine

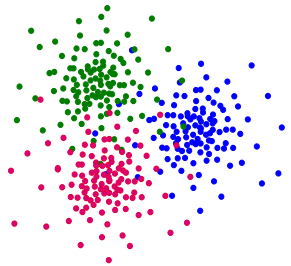
ICML 2016 Tutorial

Optimization at the heart of Machine Learning

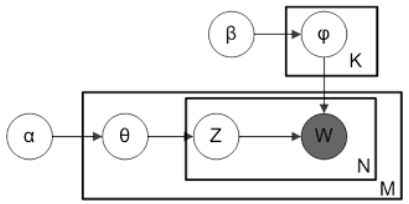
Most learning problems can be cast as optimization.

Unsupervised Learning

- Clustering
  k-means, hierarchical ...

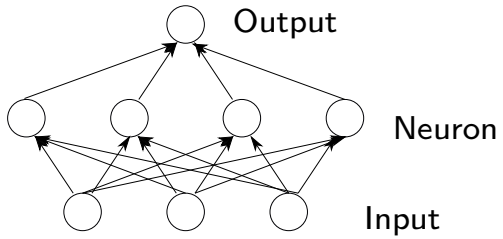


- Maximum Likelihood Estimator
  Probabilistic latent variable models



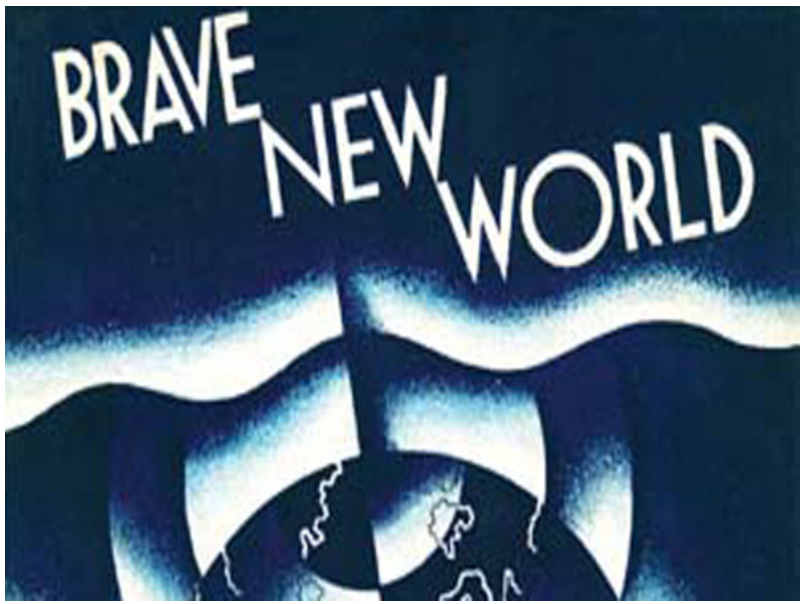
Supervised Learning

- Optimizing a neural network with respect to a loss function

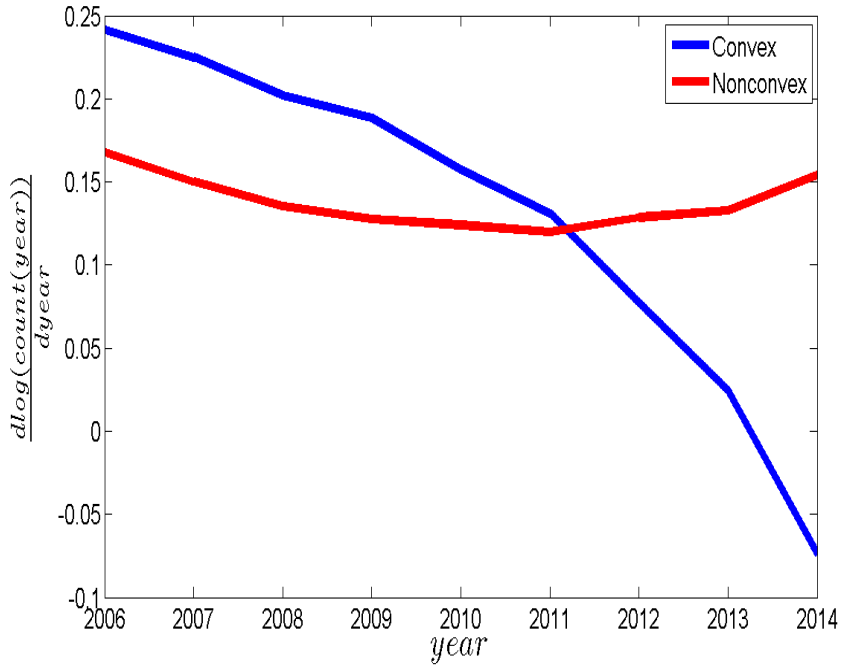


Convex vs. Non-convex Optimization

Guarantees for mostly convex..

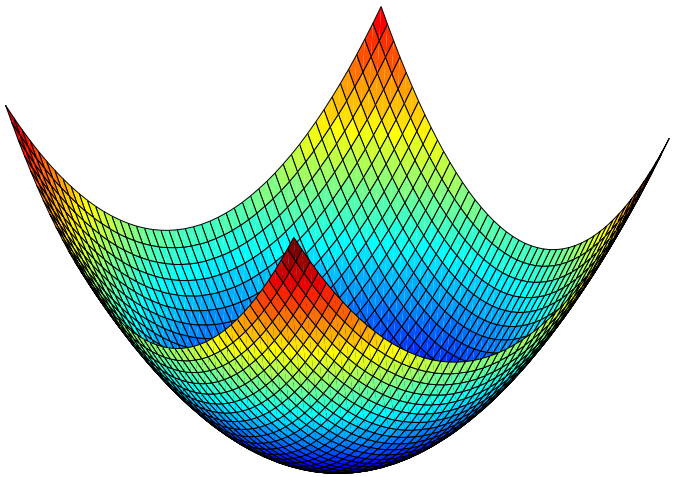


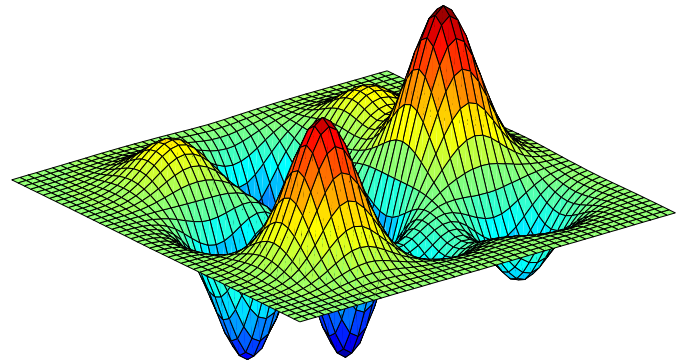
But non-convex is trending!



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

Convex vs. Nonconvex Optimization





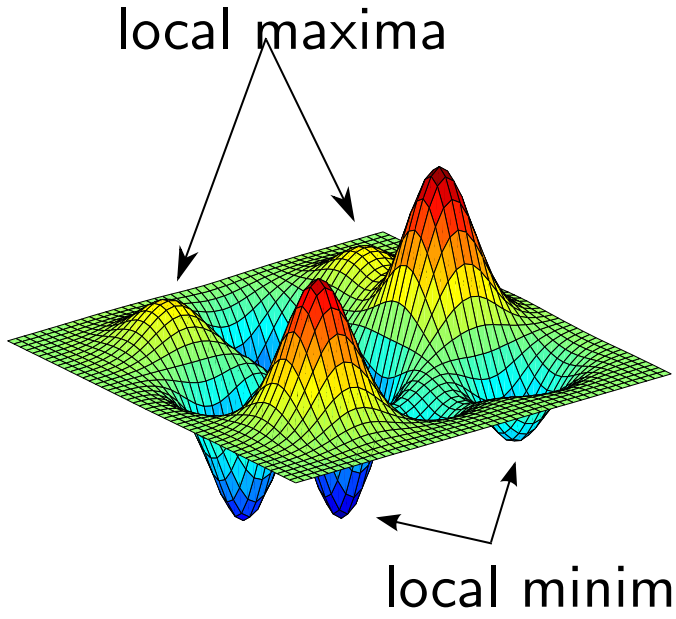
- Unique optimum: global/local.
- Multiple local optima

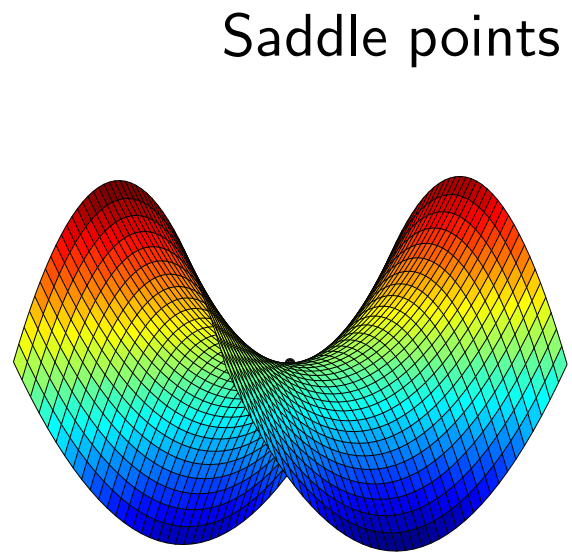
Guaranteed approaches for reaching global optima?

Non-convex Optimization in High Dimensions

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

- Curse of dimensionality: exponential number of critical points.





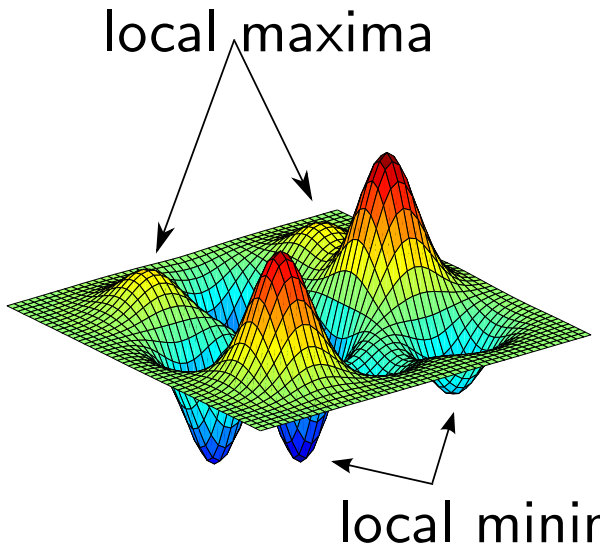
Guaranteed approaches for reaching local optima?

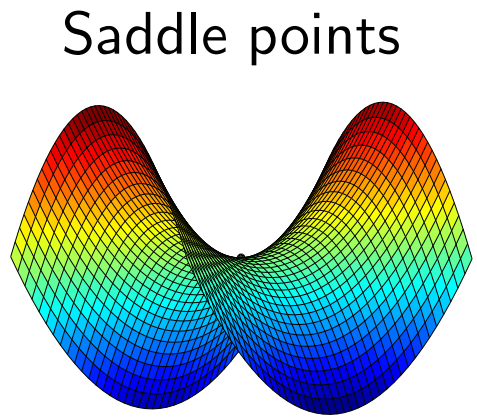
Outline

Non-convex Optimization in High Dimensions

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

- Curse of dimensionality: exponential number of critical points.
- Escaping saddle points in high dimensions?
- Can SGD escape in bounded time?

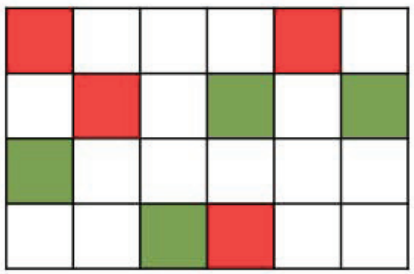




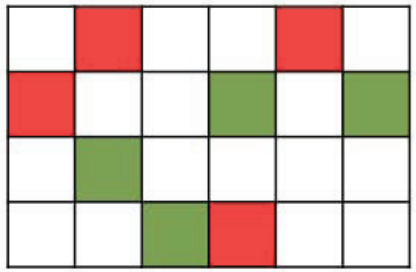
Why are saddle points ubiquitous?

Symmetries in optimization landscapes

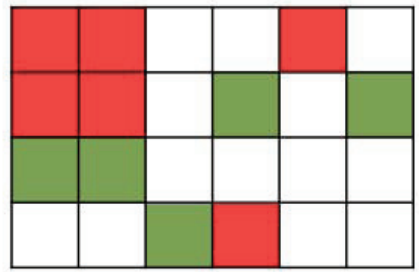
- $f(\cdot)$ invariant to permutations: $f(x_1, x_2, \ldots) = f(x_2, x_1, \ldots) = \ldots$
- E.g. deep learning, mixture models ...
- Peril of non-convexity: Avg. of optimal solutions is a critical point but NOT optimal!



Optimal Solution



Equivalent Solution

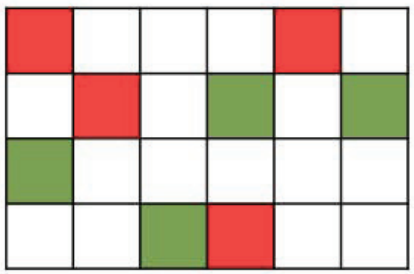


Not optimal

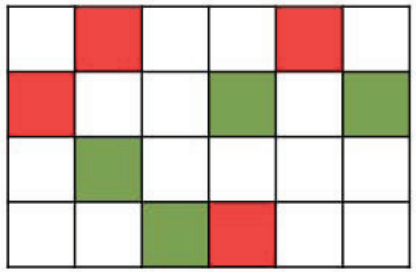
Why are saddle points ubiquitous?

Symmetries in optimization landscapes

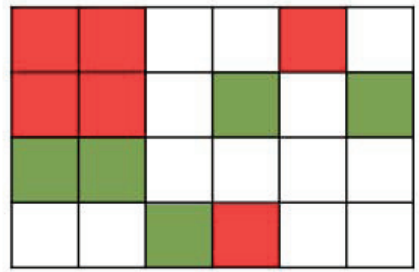
- $f(\cdot)$ invariant to permutations: $f(x_1, x_2, \ldots) = f(x_2, x_1, \ldots) = \ldots$
- E.g. deep learning, mixture models ...
- Peril of non-convexity: Avg. of optimal solutions is NOT optimal!



Optimal Solution



Equivalent Solution



Not optimal

Optimization in deep learning

- Exponentially more equivalent solutions with no. of neurons, layers ...

No free lunch: rich expressivity of large deep networks comes at a cost.

Figures obtained from Rong Ge.

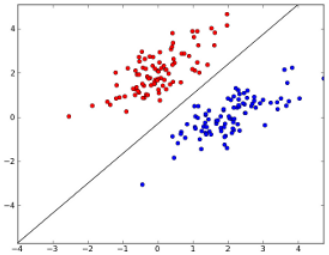
More Critical Points in Overspecified Models

- Overspecification: More capacity (neurons) than required.

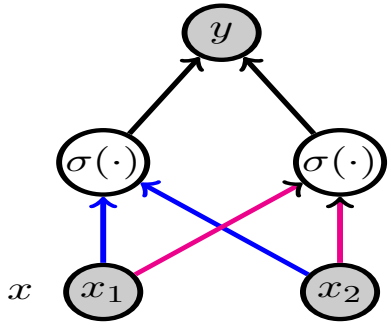
Result

- Each critical point in smaller network (over which function is realized) generates a set of critical points in larger network.
- Even global optima in smaller network can generate local optima and saddle points in larger network.

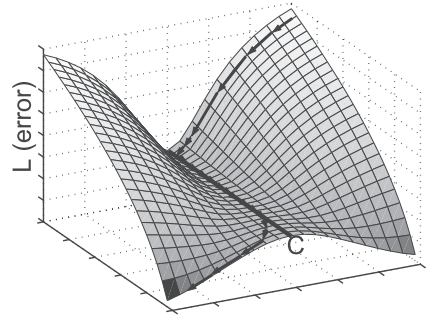
Training Data



Neural Network



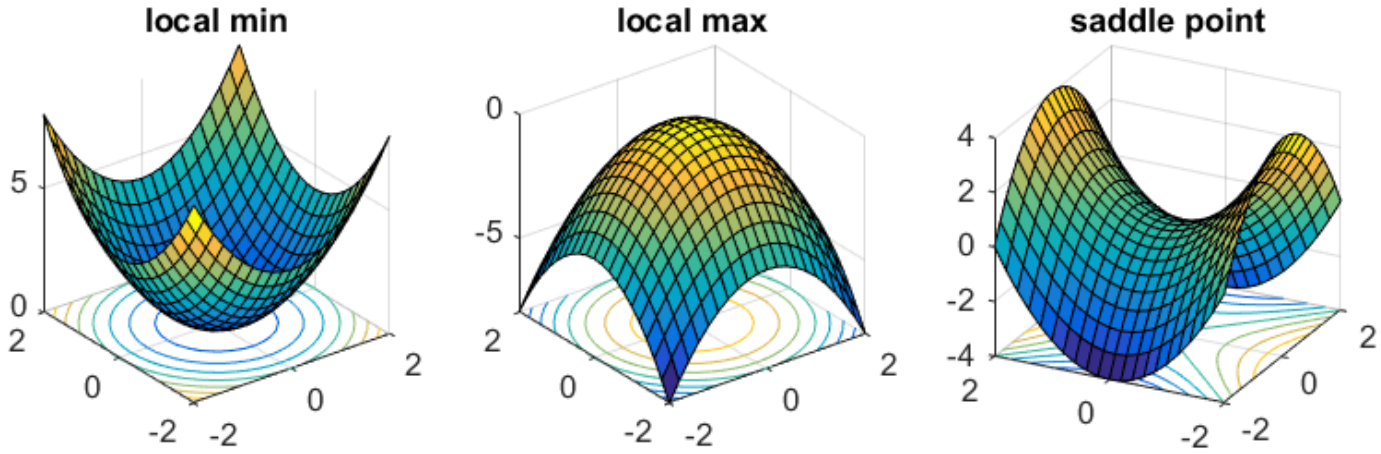
Learning Trajectory



- A long time spent in the manifold of singularities.

K Fukumizu, S Amari. Local minima and plateaus in hierarchical structures of multilayer perceptrons. 2000.

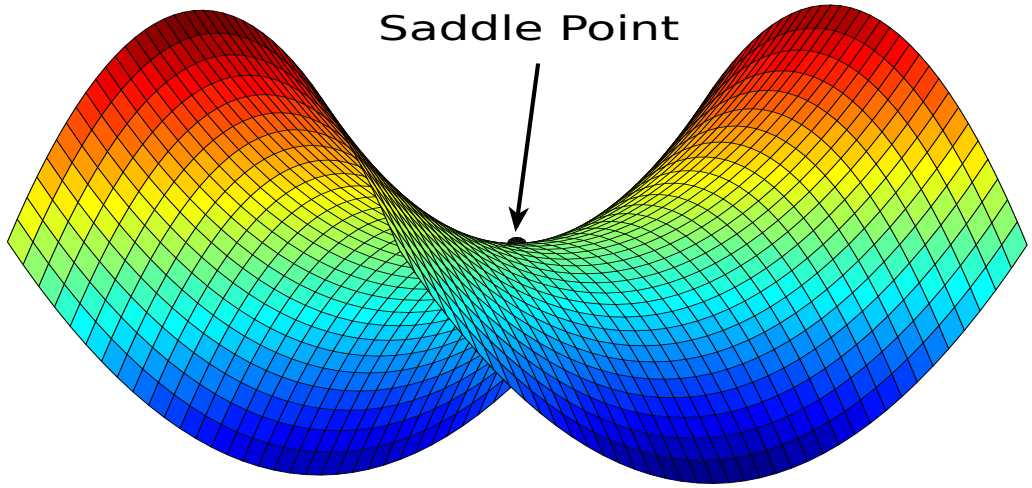
Local Optima vs. Saddle Points



- Optimization function: $f(x)$. Critical points: $\nabla_x f(x) = 0$.
- Local minima: a critical point where $\nabla^2 f(x) \succ 0$
- Local maxima: a critical point where $\nabla^2 f(x) \prec 0$
- Non-degenerate saddle point: a critical point where $\nabla^2 f(x)$ has strictly positive and negative eigenvalues.
- Indeterminate critical points: degenerate Hessian $\nabla^2 f(x) \succeq 0$ or $\nabla^2 f(x) \preceq 0$.

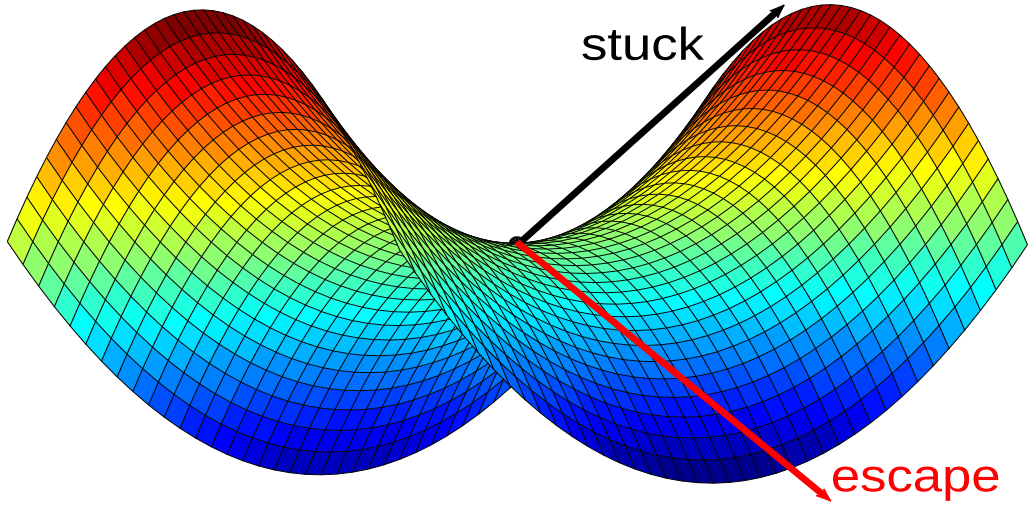
Guaranteed approaches for reaching local optima?

How to escape saddle points?



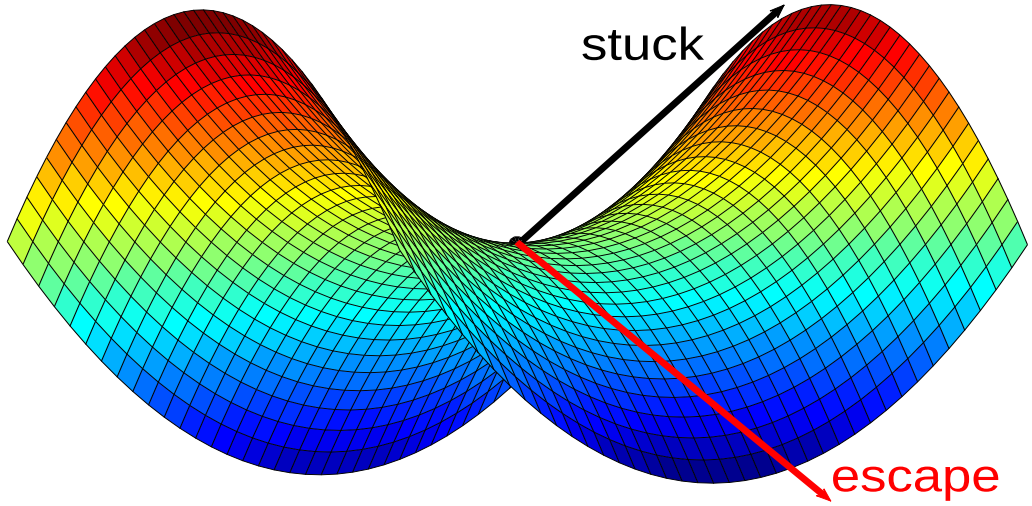
- Saddle point has 0 gradient.

How to escape saddle points?



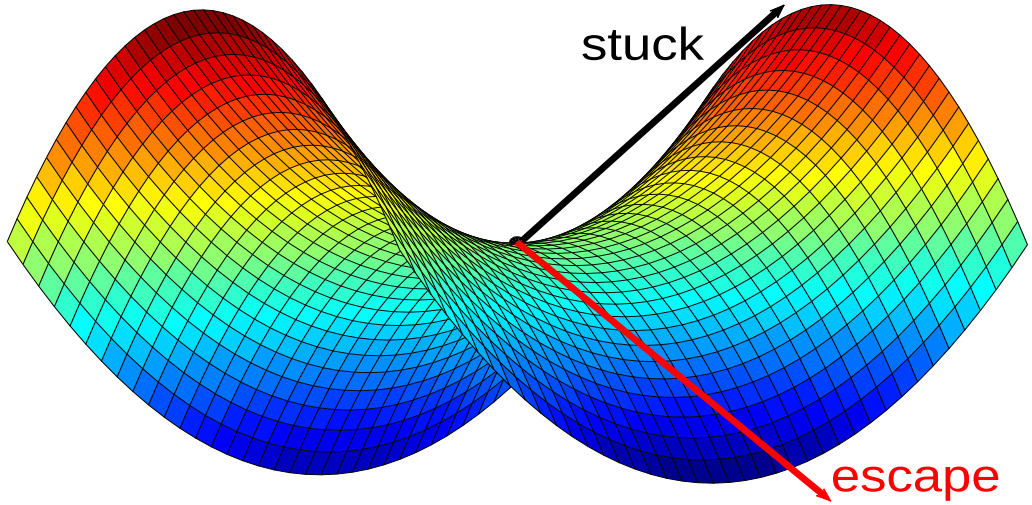
- Saddle point has 0 gradient.

How to escape saddle points?



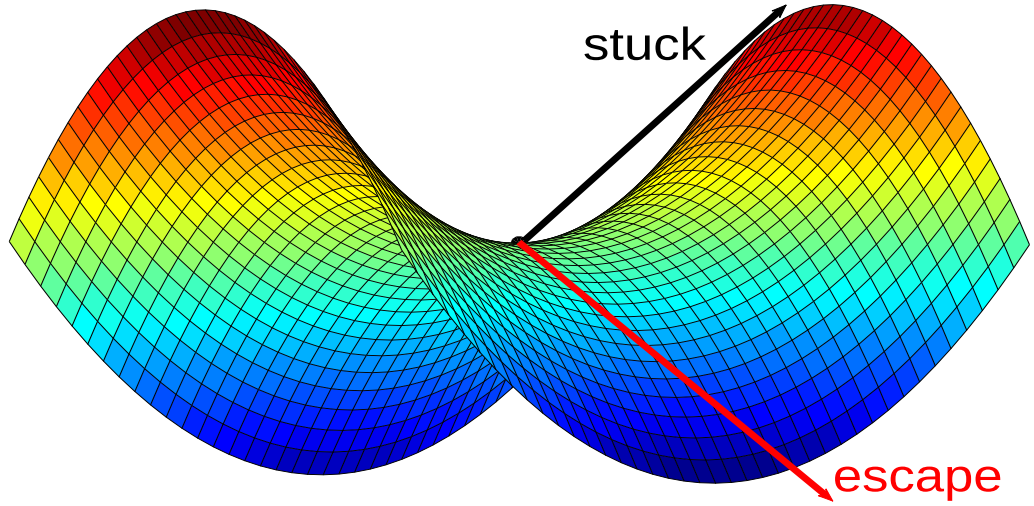
- Saddle point has 0 gradient.
- Non-degenerate saddle: Hessian has both $\pm$ eigenvalues.

How to escape saddle points?



- Saddle point has 0 gradient.
- Non-degenerate saddle: Hessian has both $\pm$ eigenvalues.
- Negative eigenvalue: direction of escape.

How to escape saddle points?



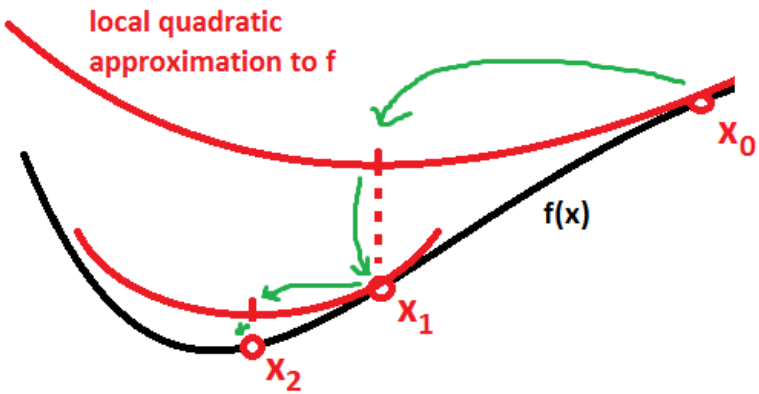
- Saddle point has 0 gradient.
- Non-degenerate saddle: Hessian has both $\pm$ eigenvalues.
- Negative eigenvalue: direction of escape.

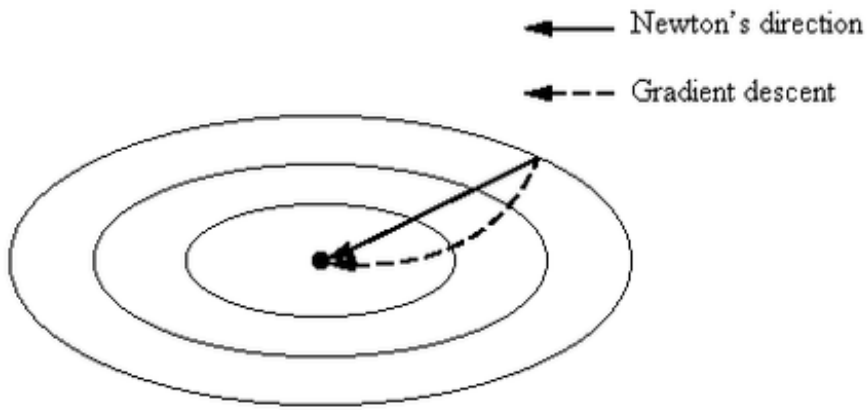
- Second order method: use Hessian information to escape.
  - Cubic regularization of Newton method, Nestorov & Polyak
- First order method: noisy stochastic gradient descent works!
  - Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition, R. Ge, F. Huang, C. Jin, Y. Yuan

Second-order Methods for Escaping Saddle Points

Recall Newton's Method

- $\Delta x = H(x)^{-1} \nabla f(x)$, where $H(x) := \nabla^2 f(x)$
- Better convergence rate for convex functions than gradient descent.

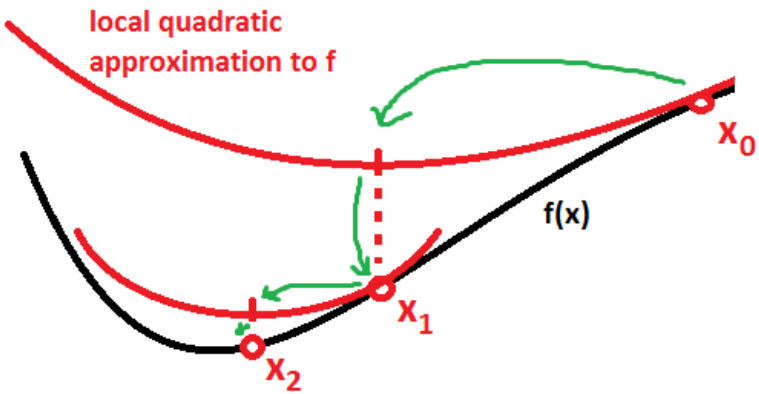


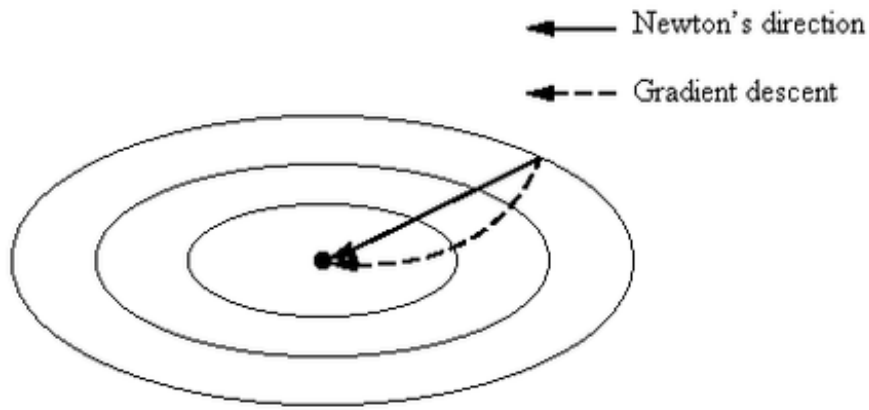


Second-order Methods for Escaping Saddle Points

Recall Newton's Method

- $\Delta x = H(x)^{-1} \nabla f(x)$, where $H(x) := \nabla^2 f(x)$
- Better convergence rate for convex functions than gradient descent.





How does Newton's method perform on non-convex functions?

Analysis of Newton's Method

Local Quadratic Approximation

Assume x^* is a non-degenerate saddle, i.e. Hessian has $-$ve eigenvalue. Let Δv_i be change along each eigenvector v_i.

$$f(x^* + \Delta x) \approx f(x^*) + 0.5 \sum_i \lambda_i \Delta v_i^2.$$

Analysis of Gradient Descent

- Preserves the sign of λ_i: movement in correct direction.
- But if initialized at x^*, stays there and movement in nbd very slow.

Analysis of Newton's Method

Local Quadratic Approximation

Assume x^* is a non-degenerate saddle, i.e. Hessian has $-$ve eigenvalue. Let Δv_i be change along each eigenvector v_i.

$$f(x^* + \Delta x) \approx f(x^*) + 0.5 \sum_i \lambda_i \Delta v_i^2.$$

Analysis of Gradient Descent

- Preserves the sign of λ_i: movement in correct direction.
- But if initialized at x^*, stays there and movement in nbd very slow.

Analysis of Newton's Method

- Rescales each eigen direction as λ_i^{-1}: better convergence.
- If $\lambda_i < 0$: wrong direction since rescaling cancels the sign.

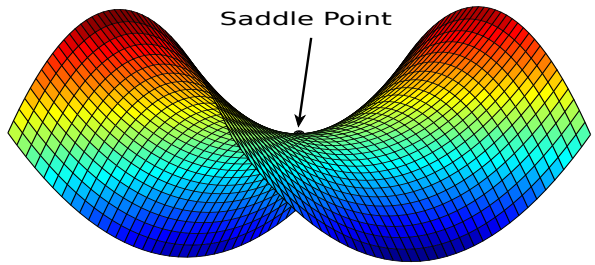
Newton's Method converges to a Saddle Point!

"Identifying and attacking the saddle point problem in high-dimensional non-convex optimization" by Y. Dauphin et al 2014.

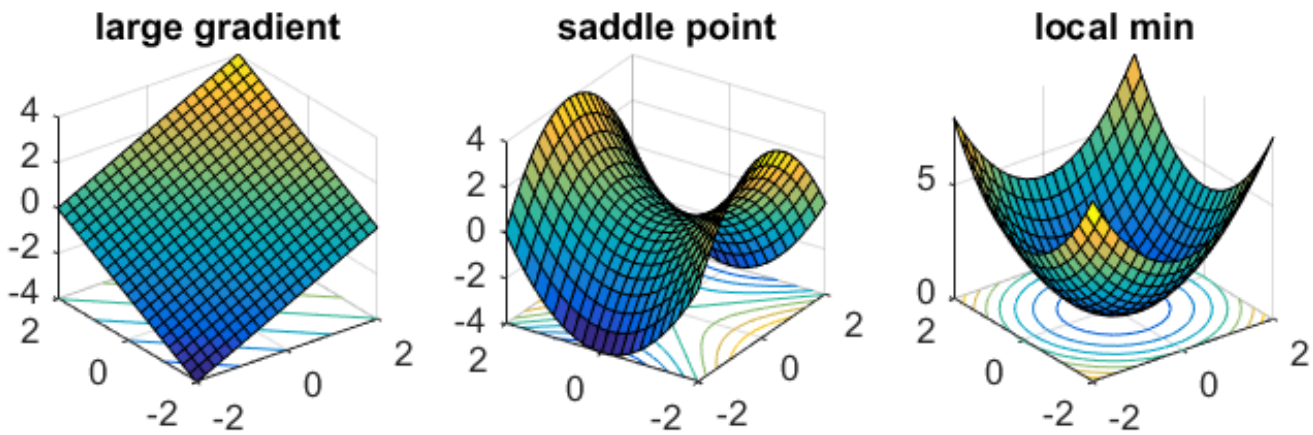
Saddle free Trust Region Methods

Escaping non-degenerate saddle point

- Moving along eigenvector with $\lambda_i < 0$ improves $f(x)$.



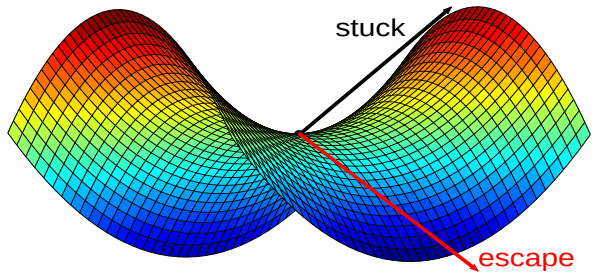
Simple method: Switch between gradient descent & Hessian methods



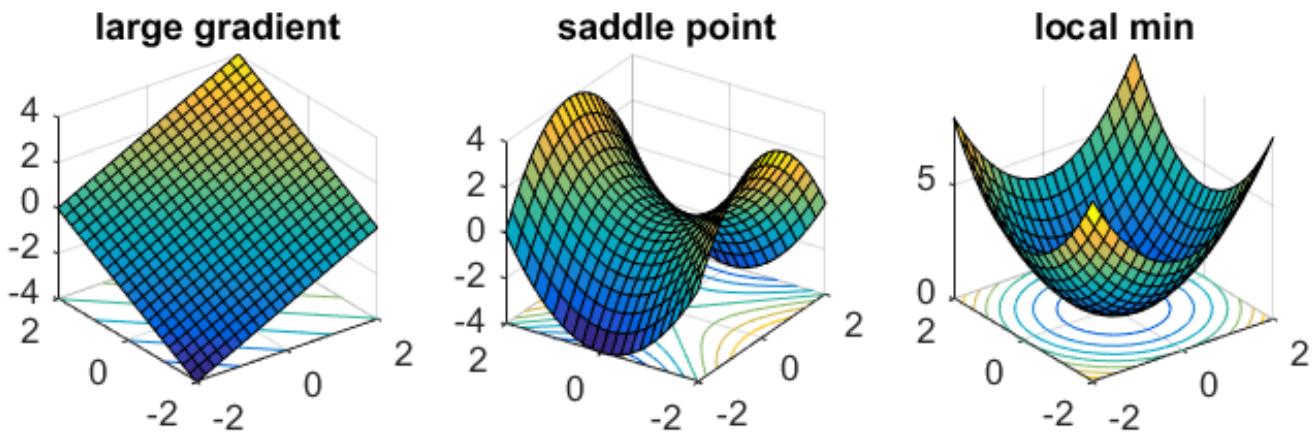
Saddle free Trust Region Methods

Escaping non-degenerate saddle point

- Moving along eigenvector with $\lambda_i < 0$ improves $f(x)$.



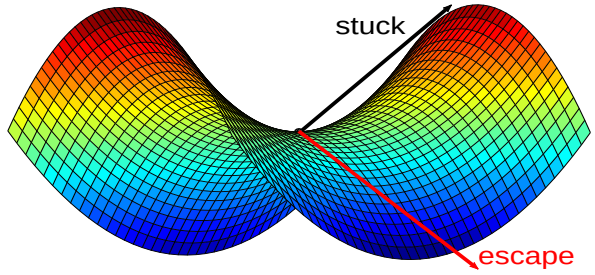
Simple method: Switch between gradient descent & Hessian methods



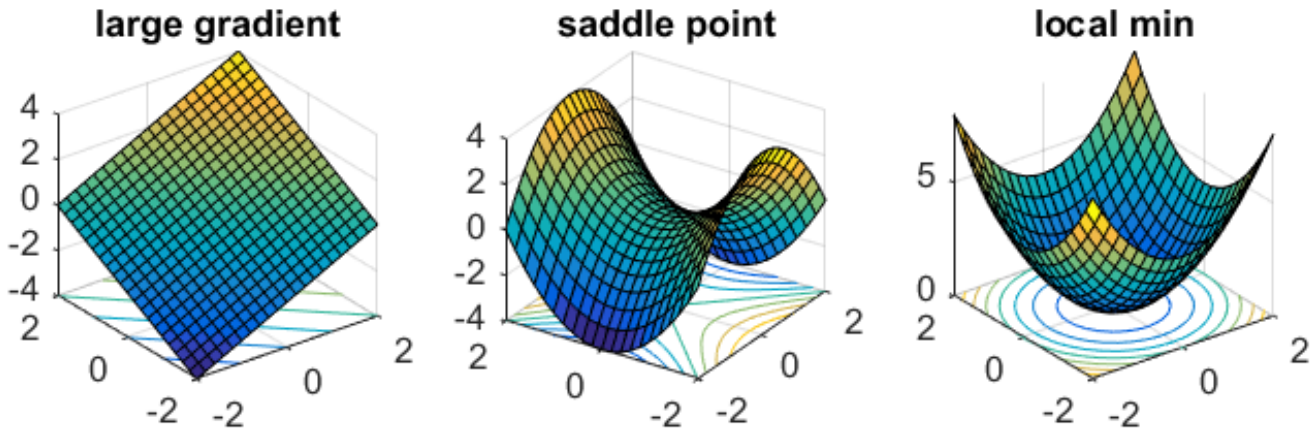
Saddle free Trust Region Methods

Escaping non-degenerate saddle point

- Moving along eigenvector with $\lambda_i < 0$ improves $f(x)$.



Simple method: Switch between gradient descent & Hessian methods



- More sophisticated: cubic regularization of Newton method (Nestorov & Polyak)

First Order Method for Escaping Saddle Points?

- Second order methods expensive due to Hessian computation.
- Approximate Hessian: not easy to analyze for non-convex methods.

Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition, R. Ge, F. Huang, C. Jin, Y. Yuan

First Order Method for Escaping Saddle Points?

- Second order methods expensive due to Hessian computation.
- Approximate Hessian: not easy to analyze for non-convex methods.

Shortcoming of gradient descent

- Can get stuck at saddle point: for certain initializations.

Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition, R. Ge, F. Huang, C. Jin, Y. Yuan

First Order Method for Escaping Saddle Points?

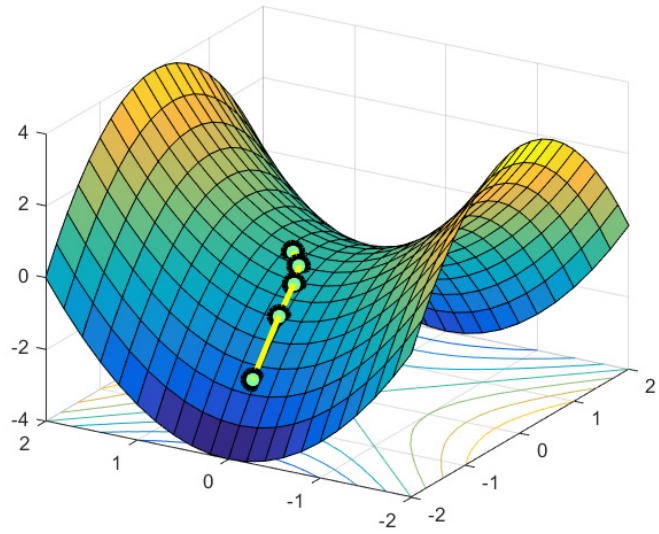
- Second order methods expensive due to Hessian computation.
- Approximate Hessian: not easy to analyze for non-convex methods.

Shortcoming of gradient descent

- Can get stuck at saddle point: for certain initializations.

Stochastic gradient descent with noise

- Noisy gradient cheaper to compute (SGD vs. GD).
- Exact gradient at saddle useless, but with sufficient noise escapes.



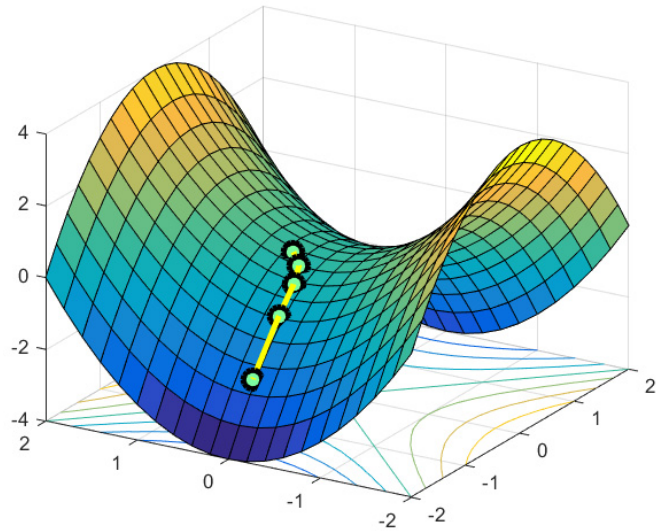
Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition, R. Ge, F. Huang, C. Jin, Y. Yuan

First Order Method for Escaping Saddle Points?

- Second order methods expensive due to Hessian computation.
- Approximate Hessian: not easy to analyze for non-convex methods.

Shortcoming of gradient descent

- Can get stuck at saddle point: for certain initializations.



Stochastic gradient descent with noise

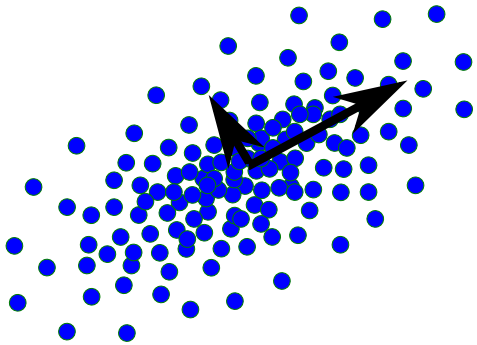
- Noisy gradient cheaper to compute (SGD vs. GD).
- Exact gradient at saddle useless, but with sufficient noise escapes.

Theorem: For smooth, twice-diff fn. with non-degenerate saddle points, noisy SGD converges to a local optimum in polynomial steps.

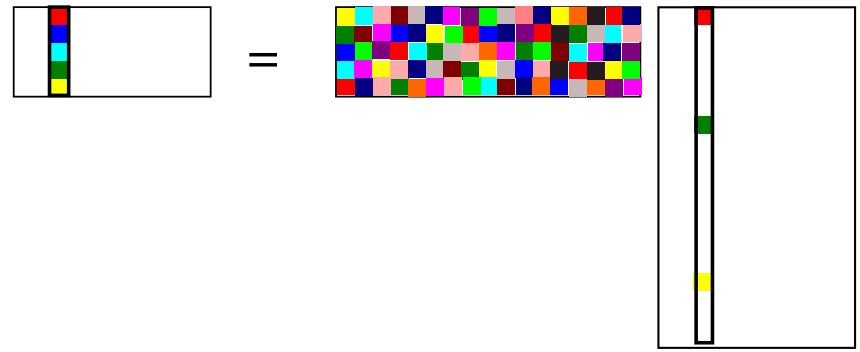
Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition, R. Ge, F. Huang, C. Jin, Y. Yuan

Problems satisfying non-degenerate saddle property

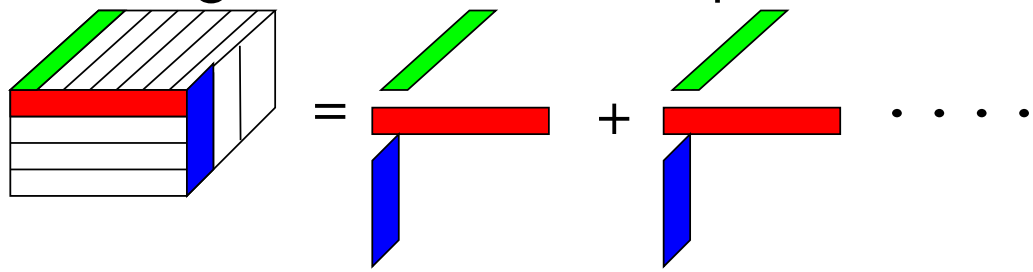
Matrix eigenvector



Dictionary Learning



Orthogonal Tensor Decomposition



- Challenging to establish this property.

What about other kinds of saddle points?

Higher order local optima

Beyond second order saddle points

- Non-degenerate saddle point: a critical point where $\nabla^2 f(x)$ has strictly positive and negative eigenvalues.
- Indeterminate critical points: degenerate Hessian $\nabla^2 f(x) \succeq 0$.

Higher order local optima

Beyond second order saddle points

- Non-degenerate saddle point: a critical point where $\nabla^2 f(x)$ has strictly positive and negative eigenvalues.
- Indeterminate critical points: degenerate Hessian $\nabla^2 f(x) \succeq 0$.

Weaker notions of local optimality under degeneracy

Higher order local optima

Beyond second order saddle points

- Non-degenerate saddle point: a critical point where $\nabla^2 f(x)$ has strictly positive and negative eigenvalues.
- Indeterminate critical points: degenerate Hessian $\nabla^2 f(x) \succeq 0$.

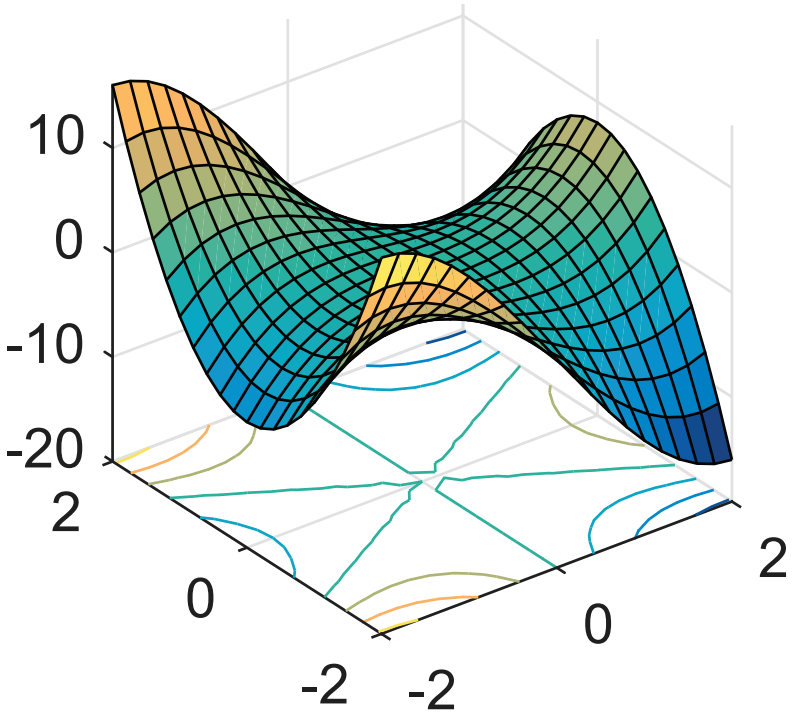
Weaker notions of local optimality under degeneracy

- A critical point x is p^{th} order local optimum if $\boxed{f(x) - f(y) = o(\|x - y\|^p)}$ for every neighbor y.

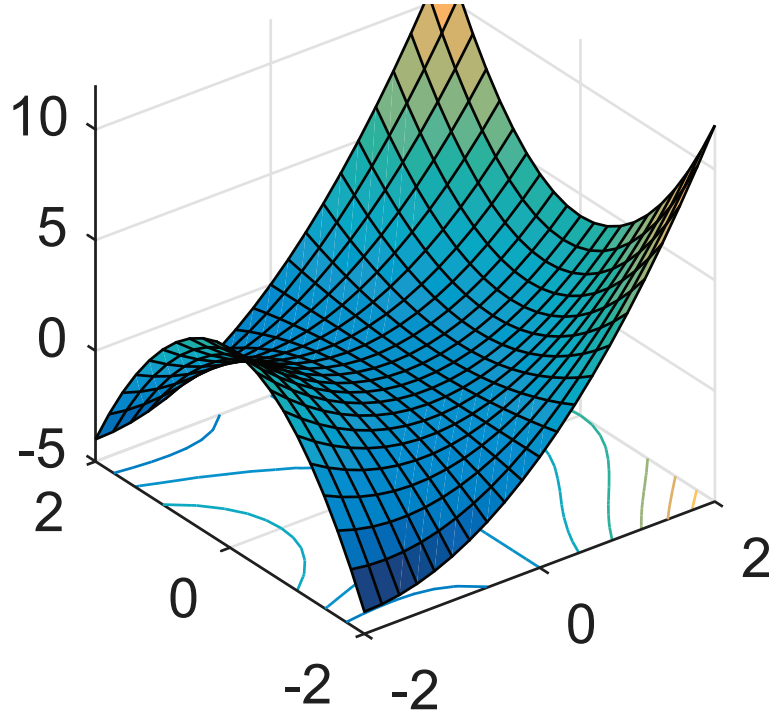
Efficient approaches for escaping higher order saddle points in non-convex optimization by A. , R. Ge, COLT 2016.

Examples of Degenerate Saddle Points

- Second order local min.
- Not third order local min.

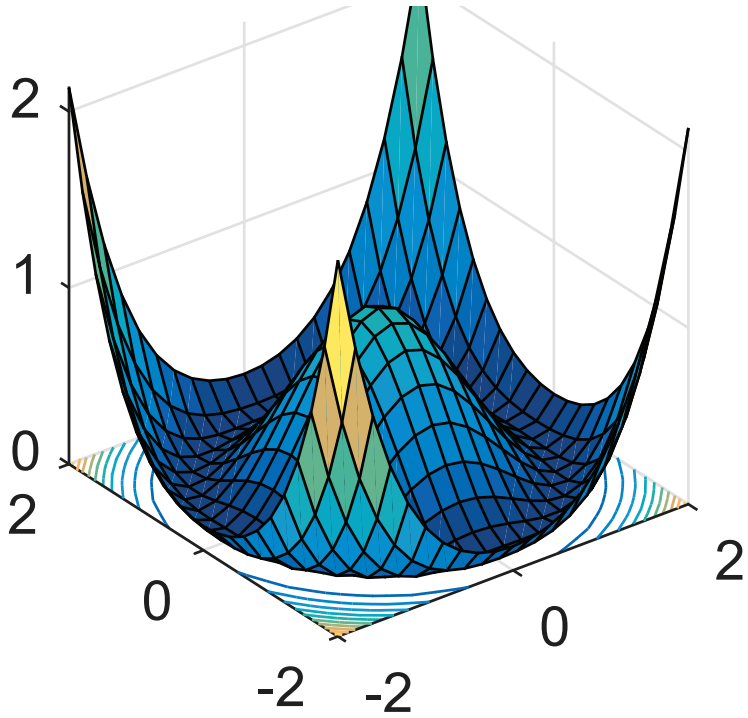


- Third order local min.
- Not fourth order local min.

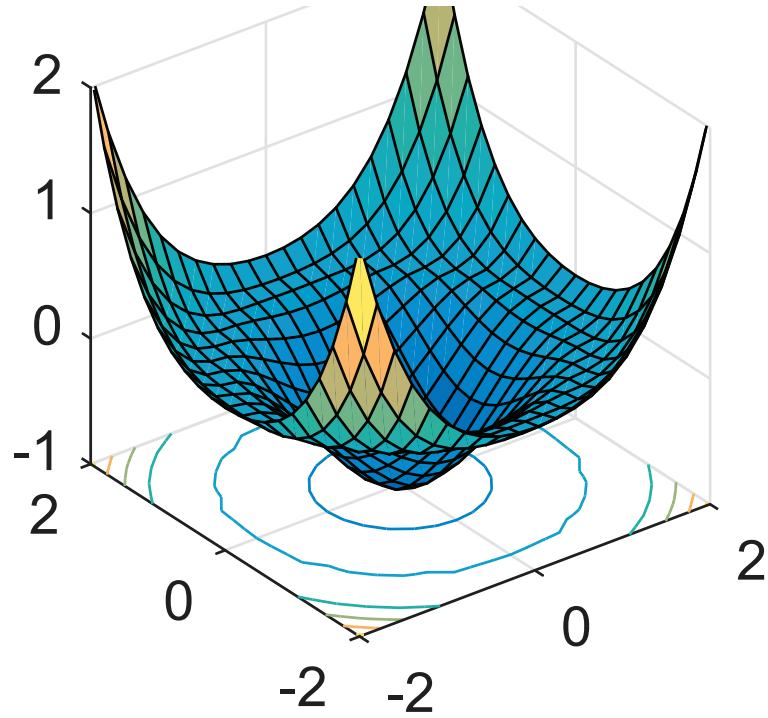


Examples of Degenerate Saddle Points

- Connected set of degenerate Hessian.
- All local optima



- Connected set of degenerate Hessian.
- All saddle points.



Escaping Degenerate Saddle Points

A point is third order local minimum iff

- It is a critical point with $\nabla^2 f(x) \succeq 0$.
- $\forall u$ in null space of $\nabla^2 f(x)$, i.e. $\nabla^2 f(x) u = 0$,

$$\boxed{\sum_{i,j,k} u_i u_j u_k \cdot \frac{\partial^3}{\partial x_i x_j x_k} f(x) = 0.}$$

Efficient approaches for escaping higher order saddle points in non-convex optimization by A. , R. Ge, COLT 2016.

Escaping Degenerate Saddle Points

A point is third order local minimum iff

- It is a critical point with $\nabla^2 f(x) \succeq 0$.
- $\forall u$ in null space of $\nabla^2 f(x)$, i.e. $\nabla^2 f(x) u = 0$,

$$\sum_{i,j,k} u_i u_j u_k \cdot \frac{\partial^3}{\partial x_i x_j x_k} f(x) = 0.$$

First method to escape third order saddle in polynomial time.

- Combination of second order and third order steps.
- Second order: Cubic reg. Newton. Conv. to second-order local opt.
- Third order: Max. tensor norm of $\nabla^3 f(x)$ in null-space of Hessian.
- Run third order only if second order does not make progress.

Efficient approaches for escaping higher order saddle points in non-convex optimization by A. , R. Ge, COLT 2016.

Concluding this Section

Summary

- Saddle points abound in high dimensional non-convex optimization.
- Symmetries and over-specification compound the problem.
- Saddle points slow down convergence.
- Solutions for escaping non-degenerate saddle points: second order trust methods and noisy SGD.
- Degenerate saddle points are harder to escape: polynomial time for third order but NP-hard for higher order.

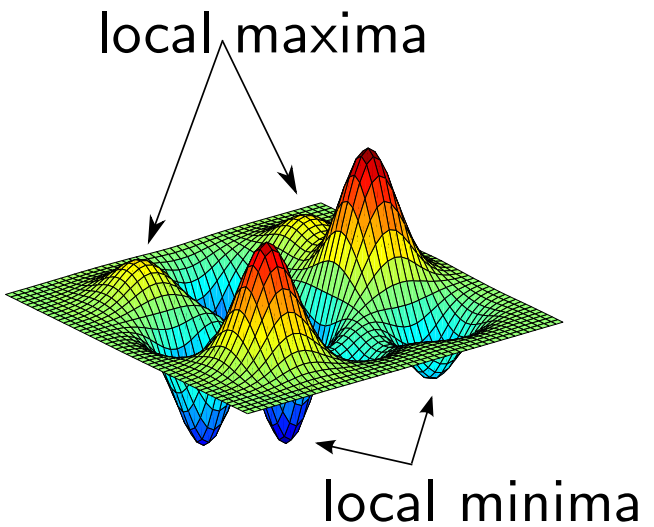
Outlook

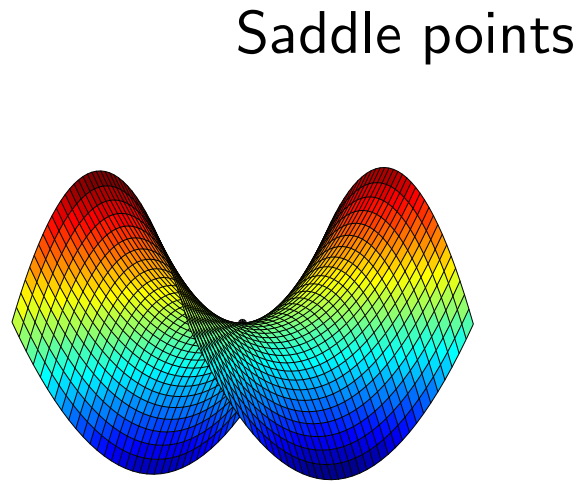
- Noisy SGD has worse scaling with dimension vs. second order methods. Can this be improved?
- What are the saddle structures of popular problems, e.g. deep learning?
- Can noisy SGD escape higher order saddle points in bounded time?

Outline

Holy Grail: Reaching Global Optimum

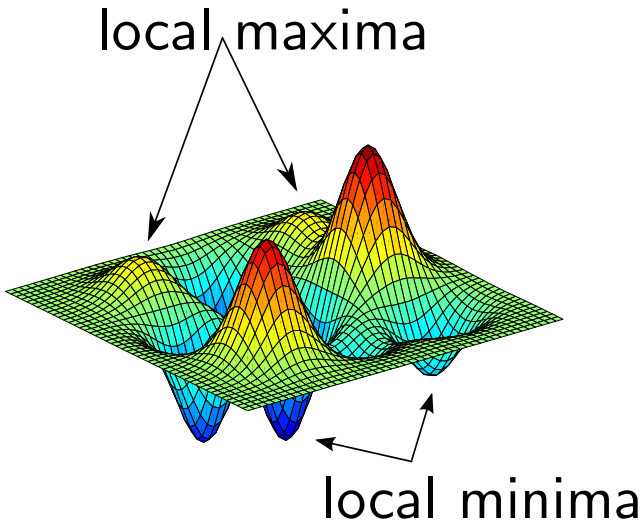
- So far: escaping saddle points using local information, i.e. derivatives.
- Can global optimum be reached using only local information?

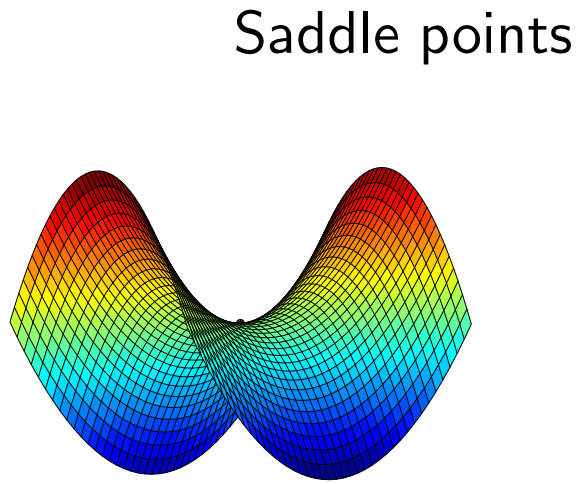




Holy Grail: Reaching Global Optimum

- So far: escaping saddle points using local information, i.e. derivatives.
- Can global optimum be reached using only local information?
- Yes in special cases, e.g. convex optimization, spectral optimization

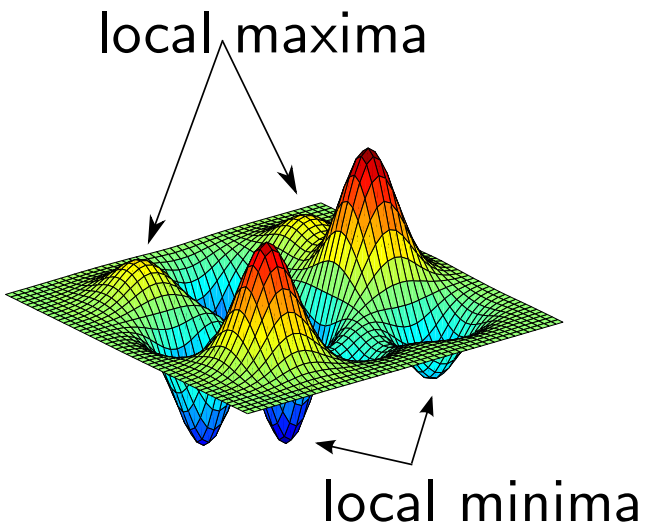


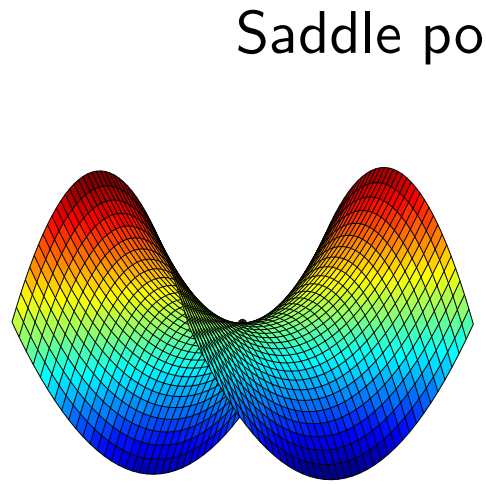


Holy Grail: Reaching Global Optimum

- So far: escaping saddle points using local information, i.e. derivatives.
- Can global optimum be reached using only local information?
- Yes in special cases, e.g. convex optimization, spectral optimization

Plan: attaining global optimality through spectral methods.



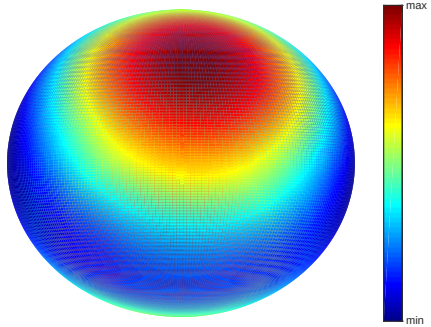


Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



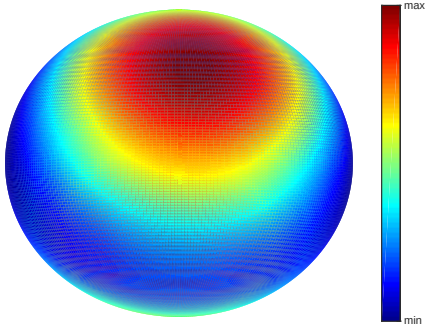
Local optimum $\equiv$ Global optimum!

Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



- Lagrangian: $\langle v, Mv \rangle - \lambda(\|v\|^2 - 1).$

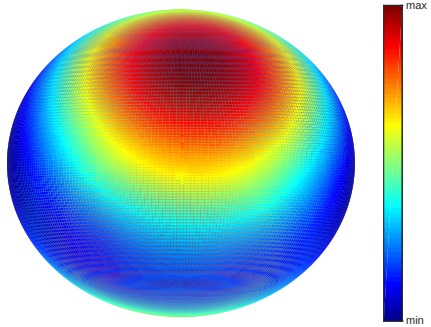
Local optimum $\equiv$ Global optimum!

Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



- Lagrangian: $\langle v, Mv \rangle - \lambda(\|v\|^2 - 1).$
- Critical points: $Mv = \lambda v$ (all eigenvectors).

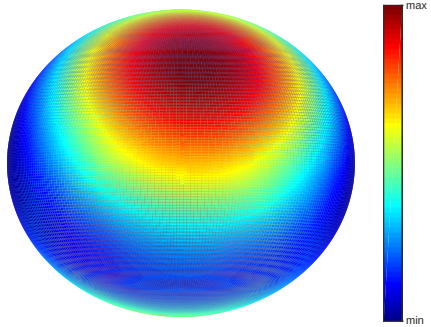
Local optimum $\equiv$ Global optimum!

Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



- Lagrangian: $\langle v, Mv \rangle - \lambda(\|v\|^2 - 1).$
- Critical points: $Mv = \lambda v$ (all eigenvectors).
- All saddle points (at most d) are non-degenerate.

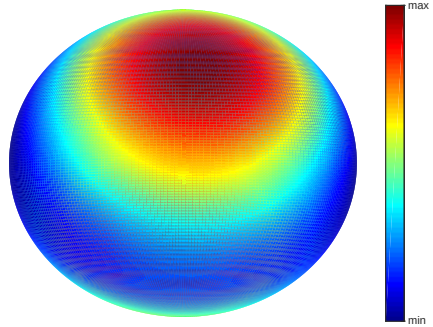
Local optimum $\equiv$ Global optimum!

Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



- Lagrangian: $\langle v, Mv \rangle - \lambda(\|v\|^2 - 1).$
- Critical points: $Mv = \lambda v$ (all eigenvectors).
- All saddle points (at most d) are non-degenerate.
- No. of local optima: 1

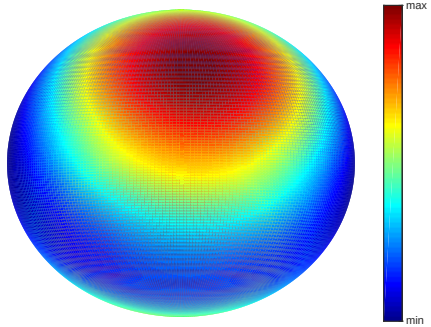
Local optimum $\equiv$ Global optimum!

Matrix Eigen-analysis

Find decomposition of matrix $M = \sum_i \lambda_i v_i v_i^\top$.

- Optimization: find top eigenvector.

$$\max_v \langle v, Mv \rangle \text{ s.t. } \|v\| = 1, v \in \mathbb{R}^d.$$



- Lagrangian: $\langle v, Mv \rangle - \lambda(\|v\|^2 - 1).$
- Critical points: $Mv = \lambda v$ (all eigenvectors).
- All saddle points (at most d) are non-degenerate.
- No. of local optima: 1

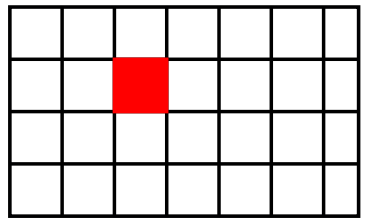
Local optimum $\equiv$ Global optimum!

Algorithmic implication

- Gradient ascent (power method) converges to global optimum!
- Saddle points avoided by random initialization!

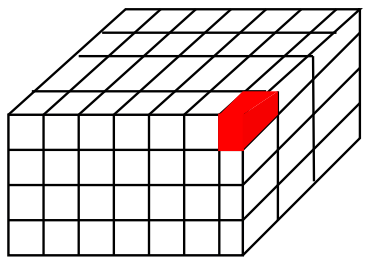
From Matrices to Tensors

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: $\mathbb{E}[x \otimes x] = \mathbb{E}[xx^\top]$.
- $M = uu^\top$ is rank-1 and $M_{i,j} = u_i u_j$.

Tensor: Higher order Moments



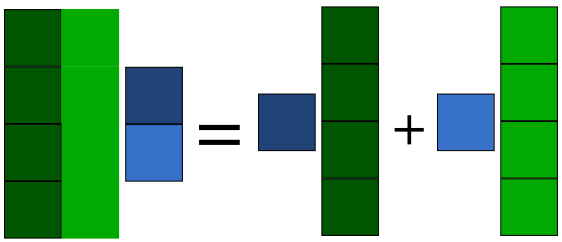
- $\mathbb{E}[x \otimes x \otimes x] \in \mathbb{R}^{d \times d \times d}$ is a third order tensor.
- $\mathbb{E}[x \otimes x \otimes x]_{i_1,i_2,i_3} = \mathbb{E}[x_{i_1} x_{i_2} x_{i_3}]$.
- $T = u \otimes u \otimes u$ is rank-1 and $T_{i,j,k} = u_i u_j u_k$.

Notion of Tensor Contraction

Extends the notion of matrix product

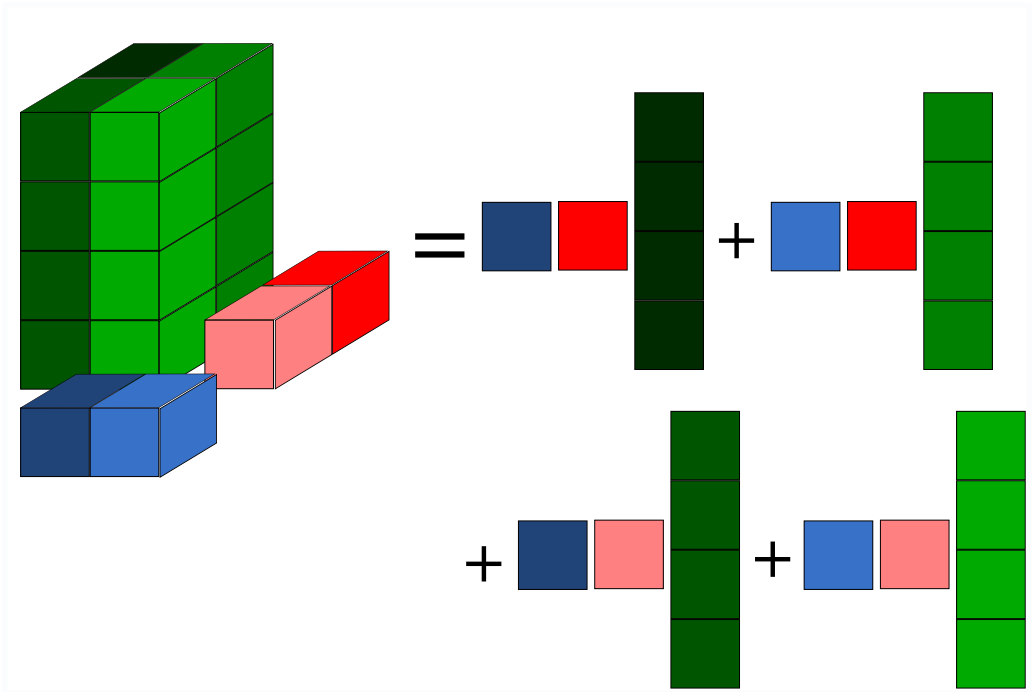
Matrix product

$$Mv = \sum_j v_j M_j$$



Tensor Contraction

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



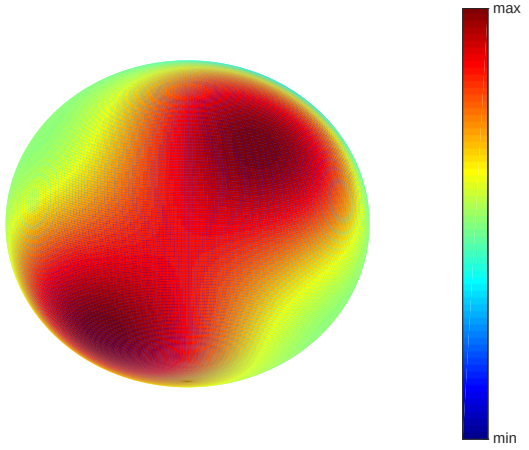
Problem of Tensor Decomposition

- Computationally hard for general tensors.
- Orthogonal tensors $\boxed{T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i}$: $u_i \perp u_j$ for $i \neq j$.

Decomposition through Tensor Norm Max.

$$\boxed{\max_v T(v, v, v),\ \|v\| = 1, v \in \mathbb{R}^d.}$$

- Lagrangian: $\boxed{T(v, v, v) - 1.5\lambda(\|v\|^2 - 1).}$



Multiple local optima, but they correspond to components!

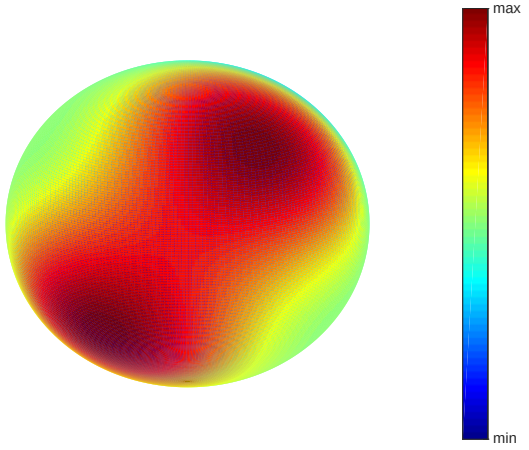
Exponentially many saddle points!

Problem of Tensor Decomposition

- Computationally hard for general tensors.
- Orthogonal tensors $\boxed{T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i}$: $u_i \perp u_j$ for $i \neq j$.

Decomposition through Tensor Norm Max.

$$\boxed{\max_v T(v, v, v),\ \|v\| = 1, v \in \mathbb{R}^d.}$$



- Lagrangian: $\boxed{T(v, v, v) - 1.5\lambda(\|v\|^2 - 1).}$
- Critical points: $T(v, v, \cdot) = \lambda v$.

Multiple local optima, but they correspond to components!

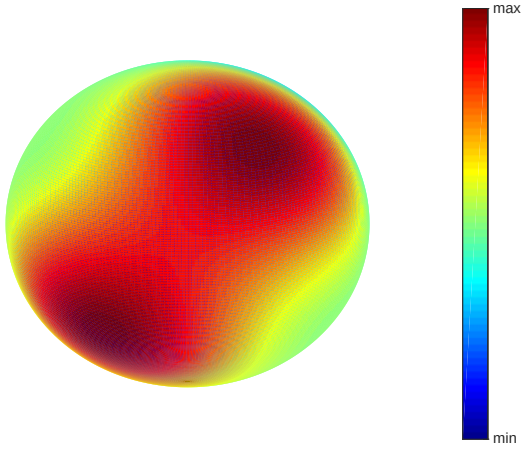
Exponentially many saddle points!

Problem of Tensor Decomposition

- Computationally hard for general tensors.
- Orthogonal tensors $\boxed{T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i}$: $u_i \perp u_j$ for $i \neq j$.

Decomposition through Tensor Norm Max.

$$\boxed{\max_{v} T(v, v, v),\ \|v\| = 1, v \in \mathbb{R}^d.}$$



- Lagrangian: $\boxed{T(v, v, v) - 1.5\lambda(\|v\|^2 - 1).}$
- Critical points: $T(v, v, \cdot) = \lambda v$.
- No. of eigenvectors: $\exp(d)$!

Multiple local optima, but they correspond to components!

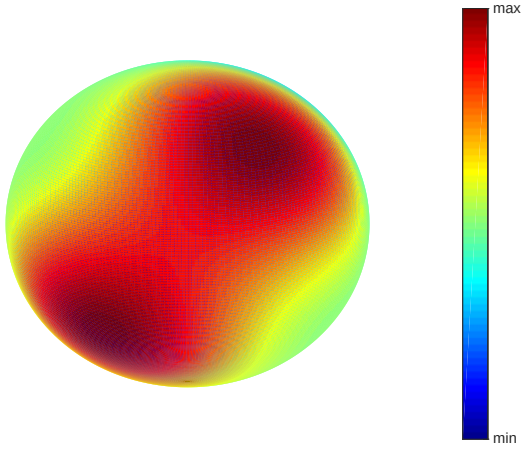
Exponentially many saddle points!

Problem of Tensor Decomposition

- Computationally hard for general tensors.
- Orthogonal tensors $\boxed{T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i}$: $u_i \perp u_j$ for $i \neq j$.

Decomposition through Tensor Norm Max.

$$\boxed{\max_{v} T(v, v, v),\ \|v\| = 1, v \in \mathbb{R}^d.}$$



- Lagrangian: $\boxed{T(v, v, v) - 1.5\lambda(\|v\|^2 - 1).}$
- Critical points: $T(v, v, \cdot) = \lambda v$.
- No. of eigenvectors: $\exp(d)!$
- All saddle points are non-degenerate.

Multiple local optima, but they correspond to components!

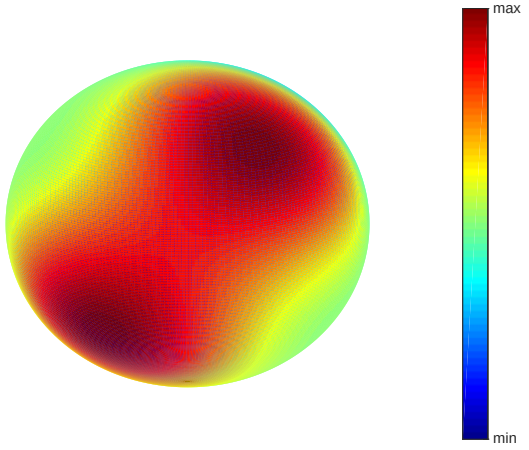
Exponentially many saddle points!

Problem of Tensor Decomposition

- Computationally hard for general tensors.
- Orthogonal tensors $\boxed{T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i}$: $u_i \perp u_j$ for $i \neq j$.

Decomposition through Tensor Norm Max.

$$\boxed{\max_{v} T(v, v, v),\ \|v\| = 1, v \in \mathbb{R}^d.}$$



- Lagrangian: $\boxed{T(v, v, v) - 1.5\lambda(\|v\|^2 - 1).}$
- Critical points: $T(v, v, \cdot) = \lambda v$.
- No. of eigenvectors: $\exp(d)!$
- All saddle points are non-degenerate.
- Local optima: $\{u_i\}$ for $i = 1, \ldots, k$.

Multiple local optima, but they correspond to components!

Exponentially many saddle points!

Implication: Guaranteed Tensor Decomposition

Given Orthogonal Tensor $T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i$.

Recover components one by one

- Run projected SGD on $\boxed{\max_{v:\|v\|=1} T(v,v,v).}$
- Guaranteed to recover a local optimum $\{u_i\}$ (upto scale).
- Find all components $\{u_i\}$ by deflation!

Implication: Guaranteed Tensor Decomposition

Given Orthogonal Tensor $T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i$.

Recover components one by one

- Run projected SGD on $\boxed{\max_{v:\|v\|=1} T(v,v,v).}$
- Guaranteed to recover a local optimum $\{u_i\}$ (upto scale).
- Find all components $\{u_i\}$ by deflation!

Alternative: simultaneous recovery of components (Ge et al '15)

- For fourth order tensor T $\boxed{\min_{\forall i, \|v_i\|=1} \sum_{i \neq j} T(v_i, v_i, v_j, v_j).}$
- All saddle points are non-degenerate.
- All local optima are global.

SGD recovers the orthgonal tensor components optimally

Perturbation Analysis for Tensor Decomposition

- Well understood for matrix decomposition vs. hard for polynomials.
- Subtle analysis for tensor decomposition.

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Perturbation Analysis for Tensor Decomposition

- Well understood for matrix decomposition vs. hard for polynomials.
- Subtle analysis for tensor decomposition.

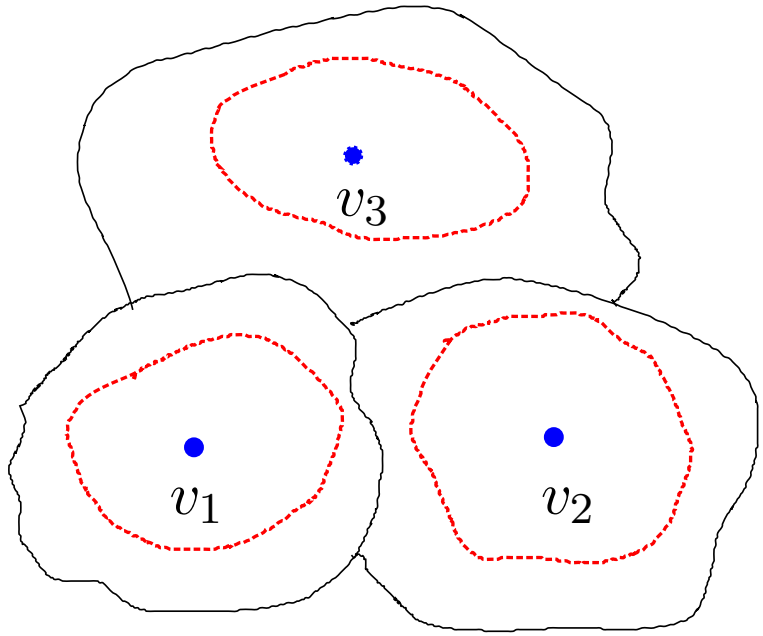
- $T \in \mathbb{R}^{d \times d \times d}$: Orthogonal tensor. E: noise tensor.

$$\hat{T} = T + E, \quad T = \sum_i \lambda_i v_i \otimes v_i \otimes v_i, \quad \|E\| := \max_{x:\|x\|=1} |E(x,x,x)|.$$

Theorem: When $\boxed{\|E\| < \frac{\lambda_{\min}}{\sqrt{d}}}$, power method recovers $\{v_i\}$ up to error $\|E\|$ with linear no. of restarts.

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

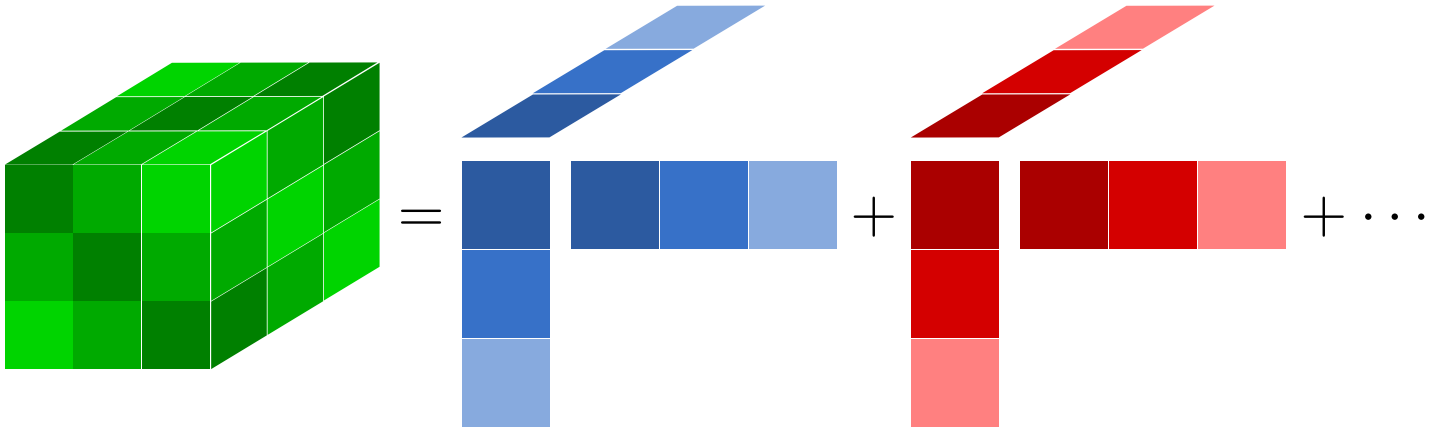
Perturbation Analysis for Tensor Decomposition



Theorem: When $\boxed{\|E\| < \frac{\lambda_{\min}}{\sqrt{d}}}$, power method recovers $\{v_i\}$ up to error $\|E\|$ with linear no. of restarts.

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

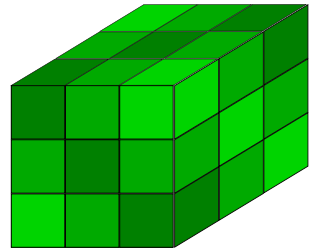


$$T = v_1{}^{\otimes 3} + v_2{}^{\otimes 3} + \cdots,$$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

Orthogonalization

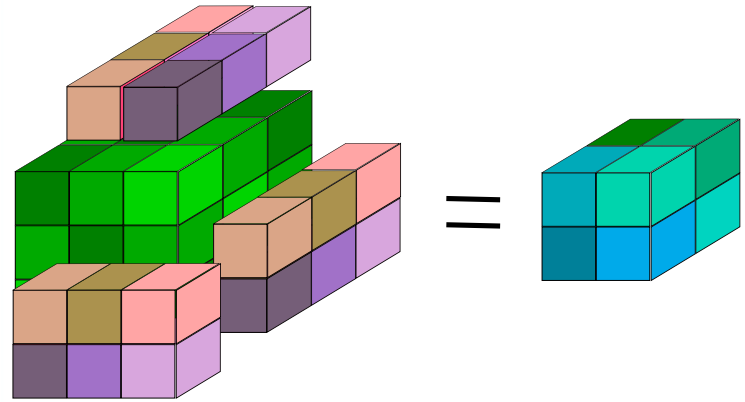


Input tensor T

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

Orthogonalization

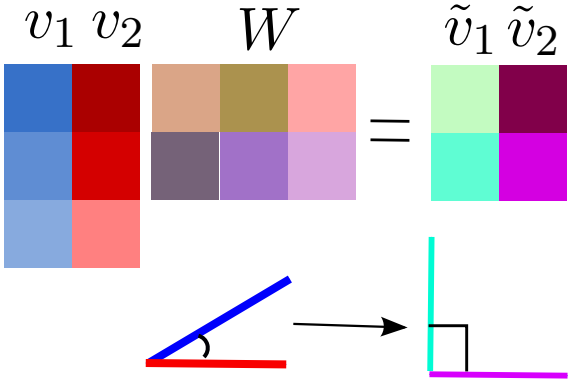


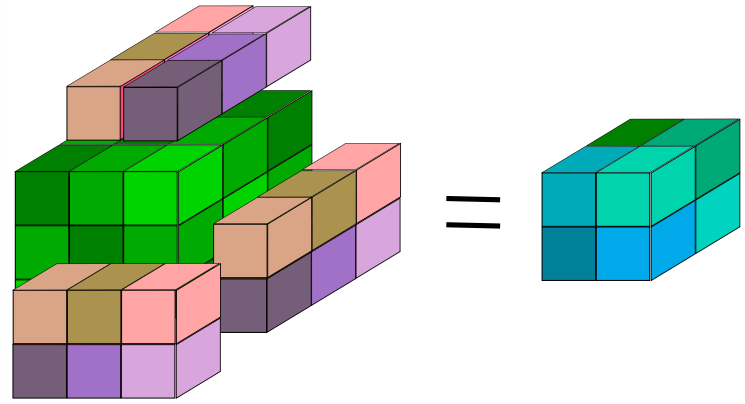
$$T(W, W, W) = \tilde{T}$$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

Orthogonalization



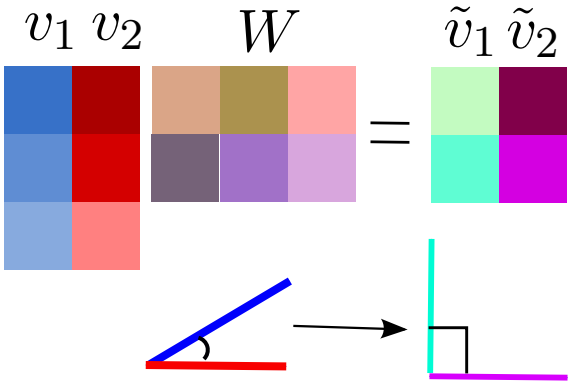


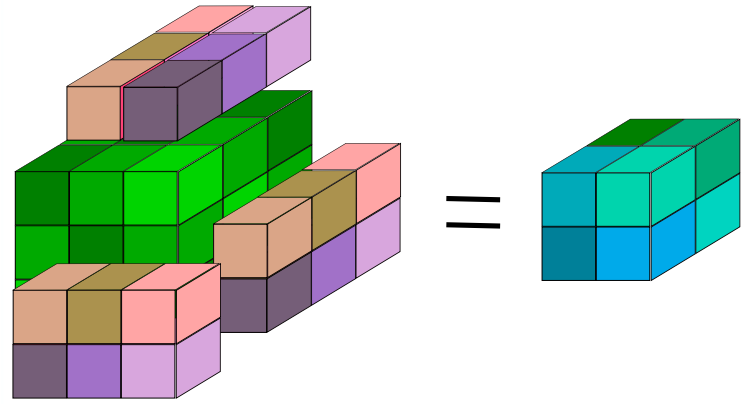
$$T(W, W, W) = \tilde{T}$$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

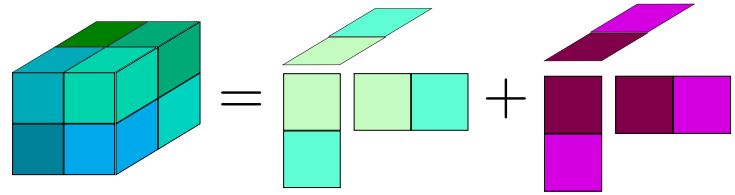
Orthogonalization





$$T(W, W, W) = \tilde{T}$$

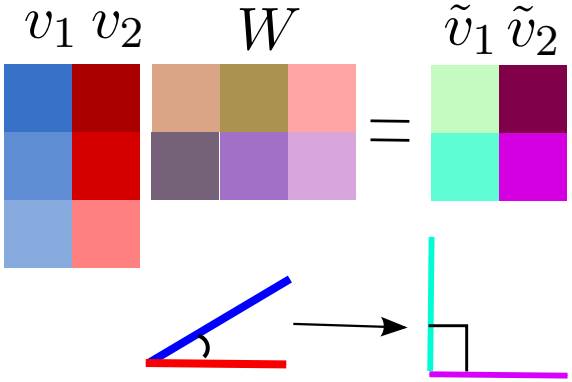
$$\tilde{T} = T(W, W, W) = \tilde{v_1}^{\otimes 3} + \tilde{v_2}^{\otimes 3} + \cdots,$$

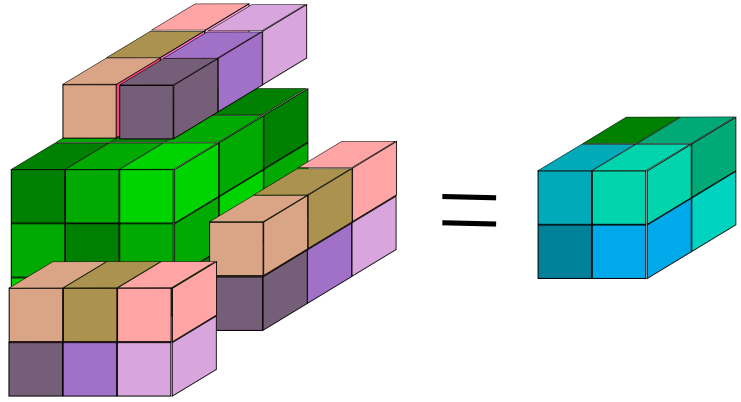


A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

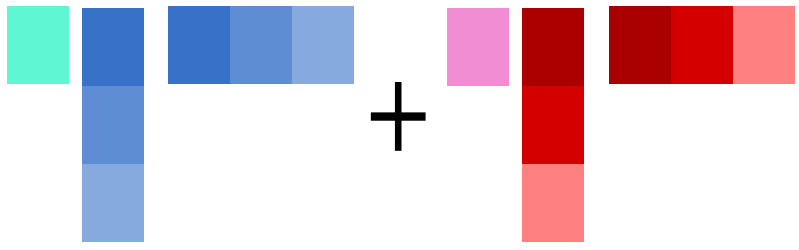
Orthogonalization



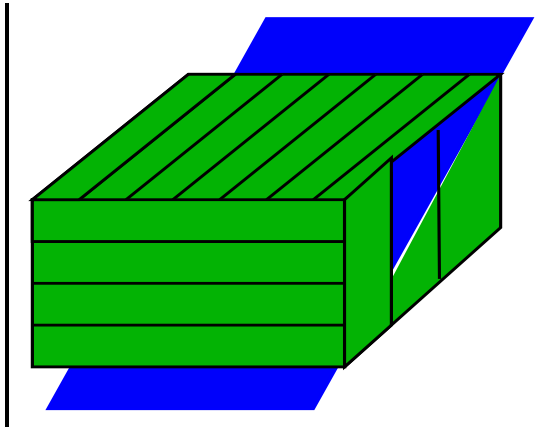


$$T(W, W, W) = \tilde{T}$$

Find W using SVD of Matrix Slice



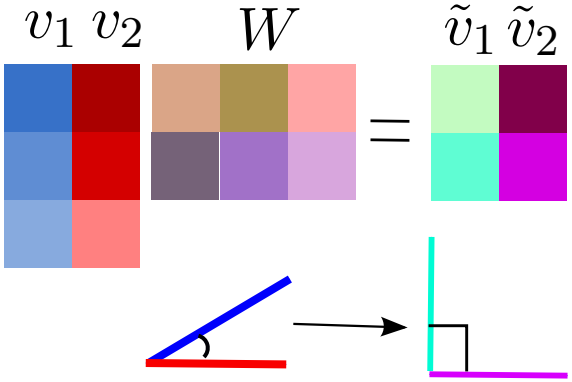
$$M = T(\cdot, \cdot, \theta) =$$

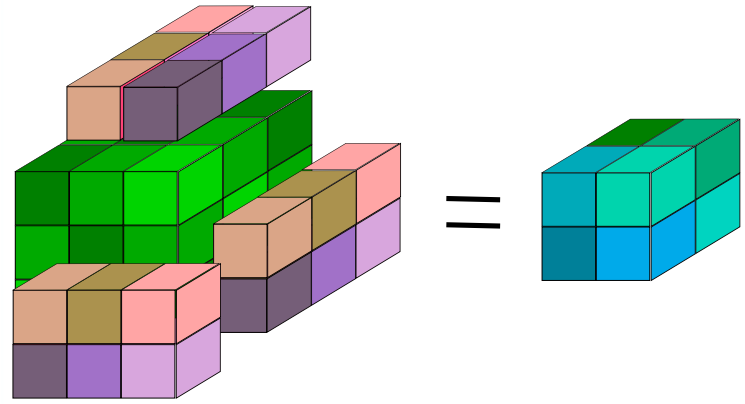


A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Non-orthogonal Tensor Decomposition

Orthogonalization





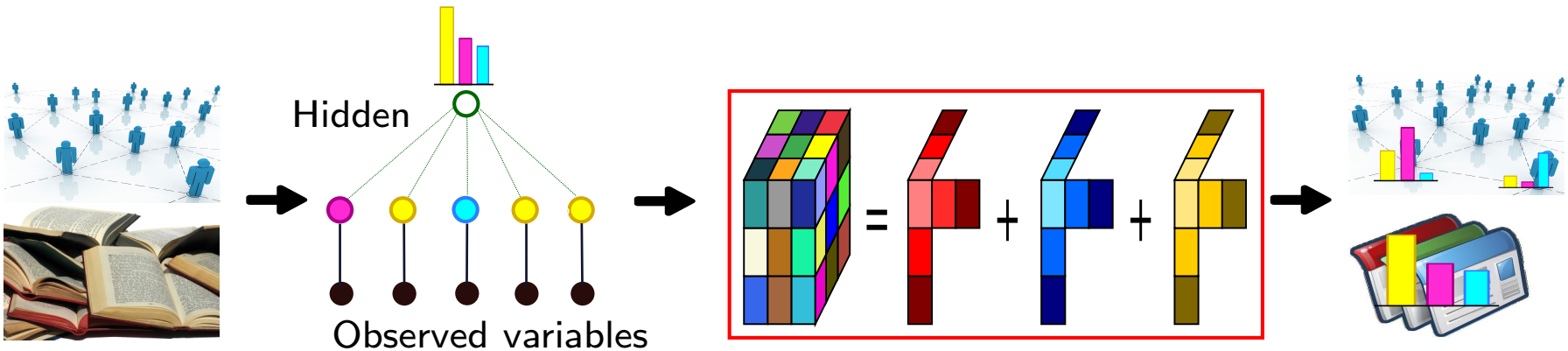
$$T(W, W, W) = \tilde{T}$$

Orthogonalization: invertible when v_i's linearly independent.

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Unsupervised Learning via Tensor Methods

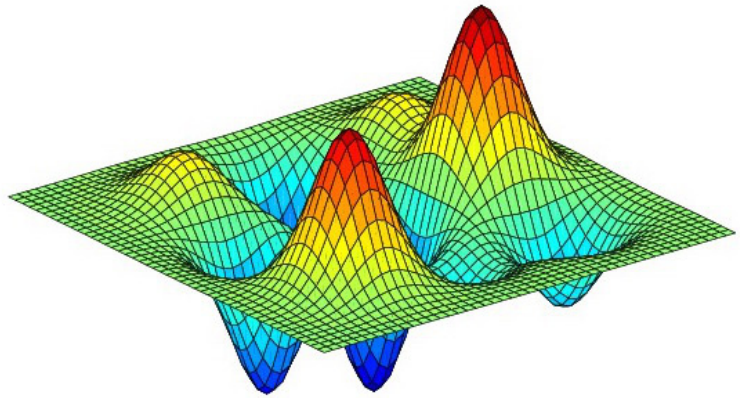
Data $\rightarrow$ Model $\rightarrow$ Learning Algorithm $\rightarrow$ Predictions



Guaranteed Learning through Tensor Methods

Replace the objective function

Max Likelihood vs. Best Tensor decomp.

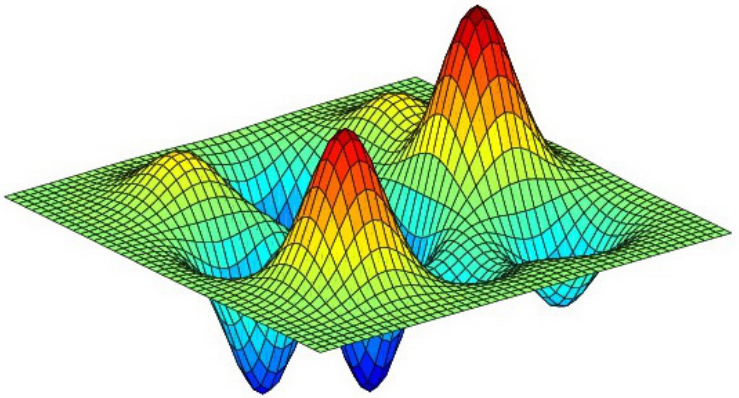


Preserves Global Optimum (infinite samples)

$$\arg\max_{\theta} p(x;\theta) = \arg\min_{\theta} \|\widehat{T}(x) - T(\theta)\|_{\mathbb{F}}^2$$

$\widehat{T}(x)$: empirical tensor, $T(\theta)$: low rank tensor based on θ.

Guaranteed Learning through Tensor Methods



Replace the objective function

Max Likelihood vs. Best Tensor decomp.

Preserves Global Optimum (infinite samples)

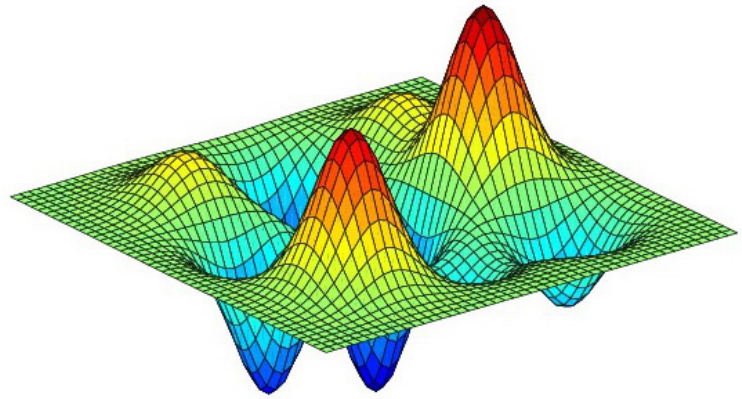
$$\arg\max_{\theta} p(x;\theta) = \arg\min_{\theta} \|\widehat{T}(x) - T(\theta)\|_{\mathbb{F}}^2$$

$\widehat{T}(x)$: empirical tensor, $T(\theta)$: low rank tensor based on θ.

Finding globally opt tensor decomposition

Simple algorithms succeed under mild and natural conditions for many learning problems.

Guaranteed Learning through Tensor Methods



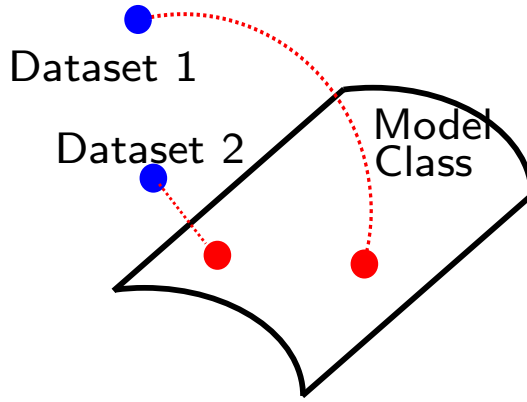
Replace the objective function

Max Likelihood vs. Best Tensor decomp.

Preserves Global Optimum (infinite samples)

$$\arg\max_{\theta} p(x;\theta) = \arg\min_{\theta} \|\widehat{T}(x) - T(\theta)\|_{\mathbb{F}}^2$$

$\widehat{T}(x)$: empirical tensor, $T(\theta)$: low rank tensor based on θ.



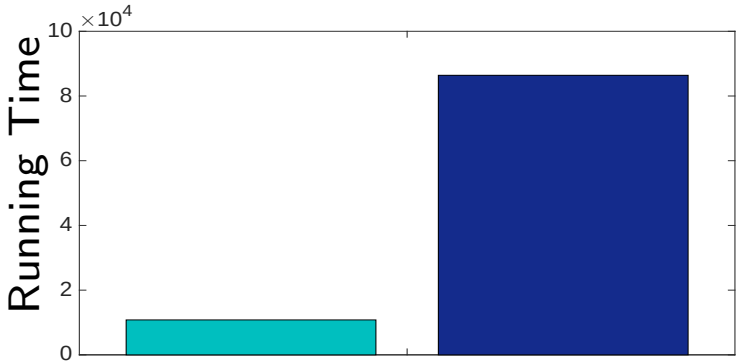
Finding globally opt tensor decomposition

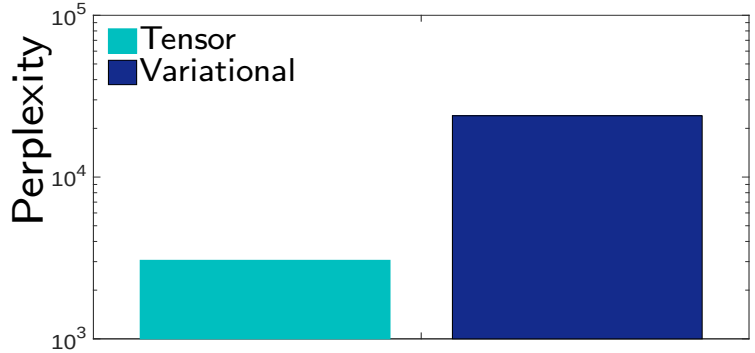
Simple algorithms succeed under mild and natural conditions for many learning problems.

Tensors vs. Variational Inference

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

Learning Topics from PubMed on Spark, 8mil articles



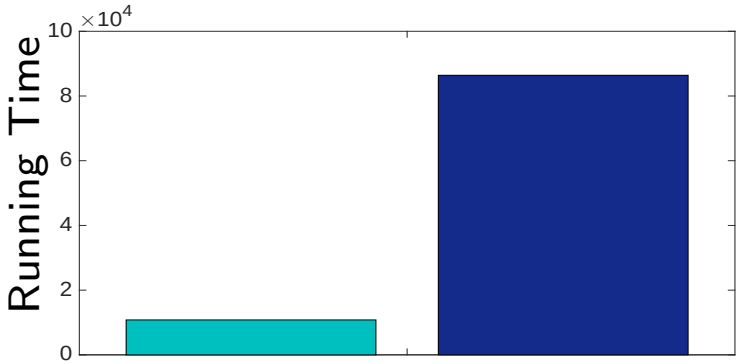


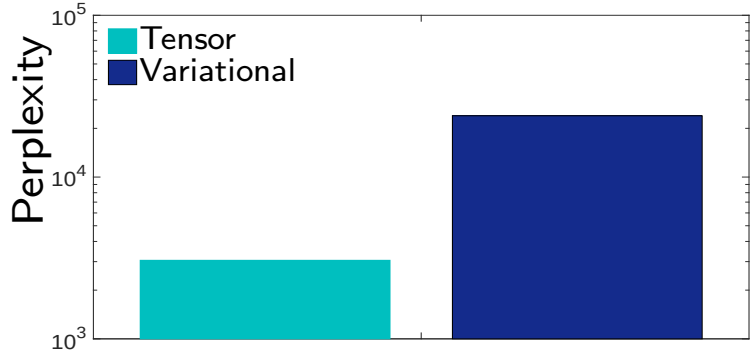
F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

Tensors vs. Variational Inference

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

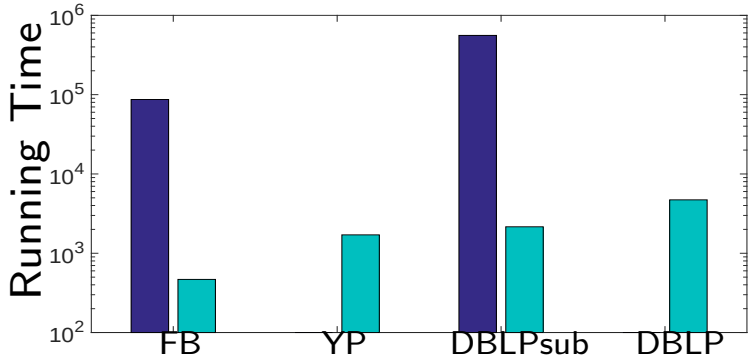
Learning Topics from PubMed on Spark, 8mil articles

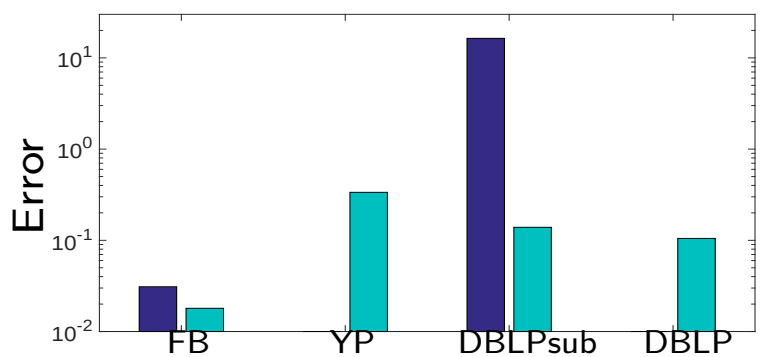




Learning network communities from social network data

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



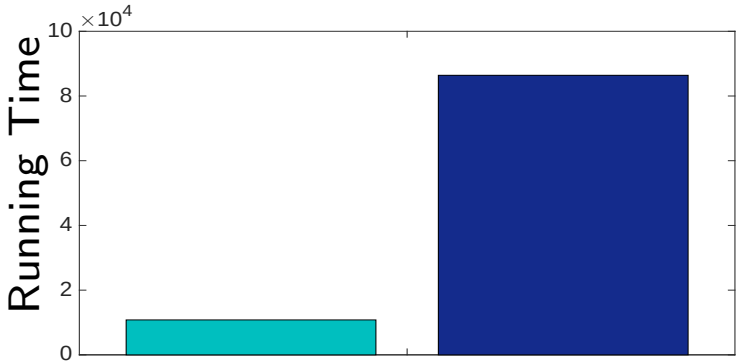


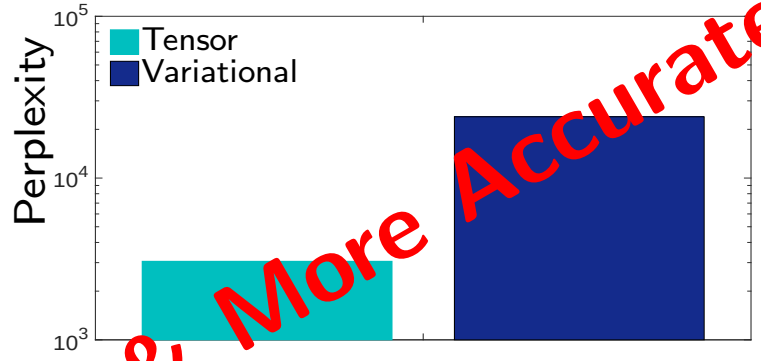
F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

Tensors vs. Variational Inference

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

Learning Topics from PubMed on Spark, 8mil articles

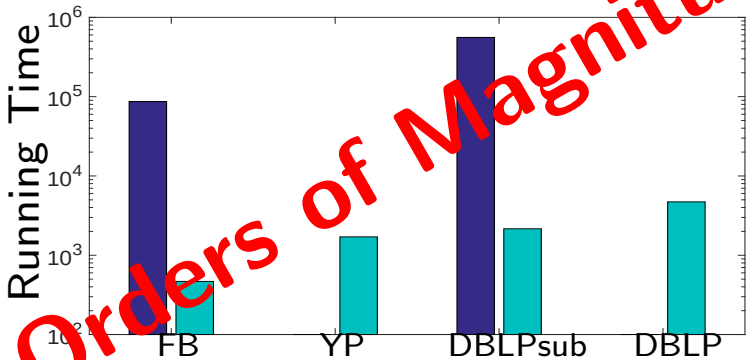


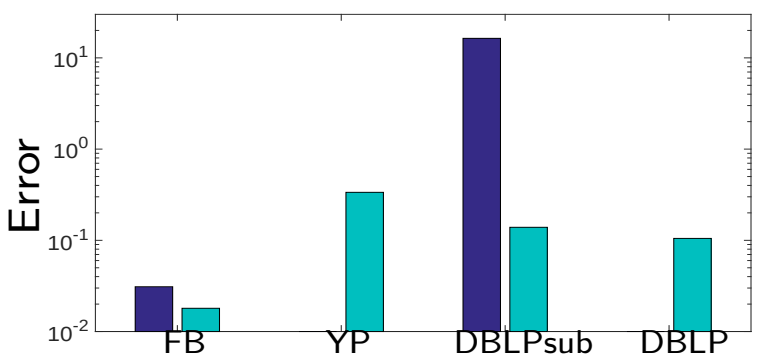


Orders of Magnitude Faster & More Accurate

Learning network communities from social network data

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

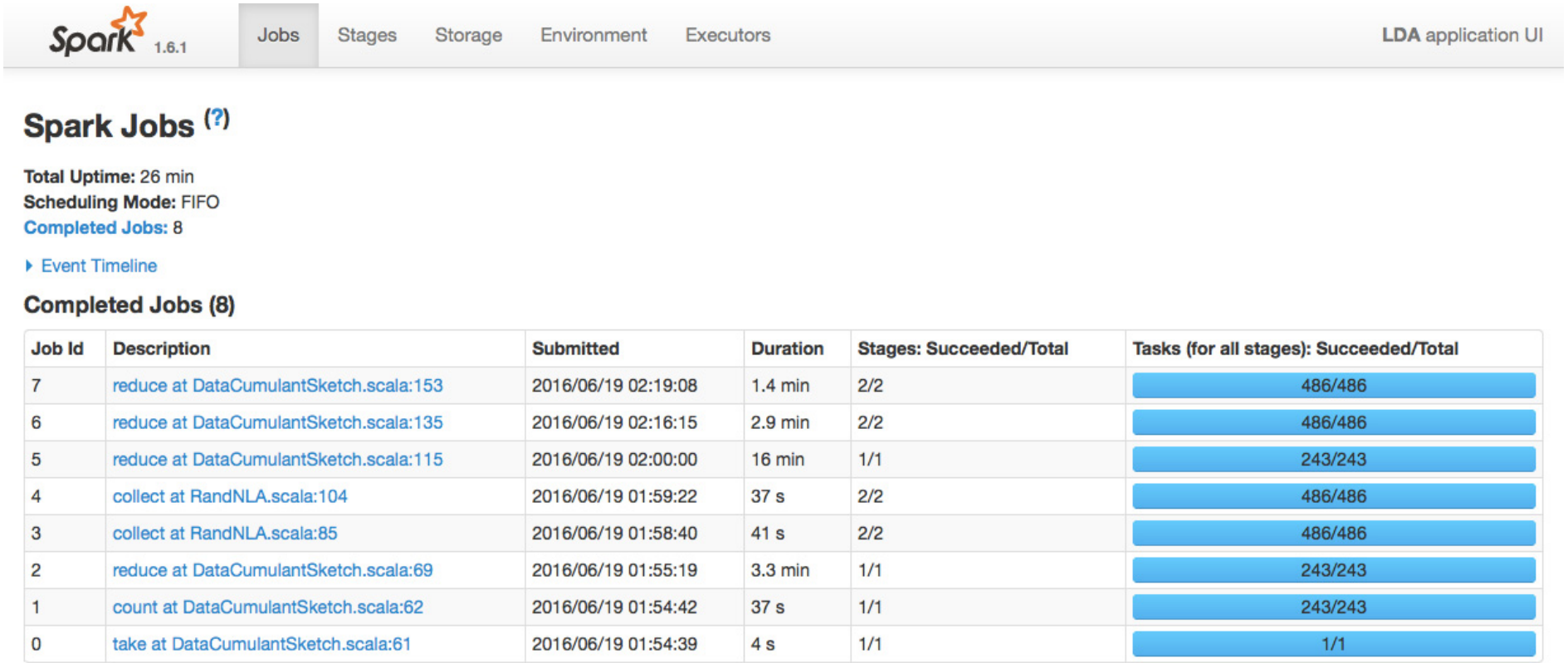




F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

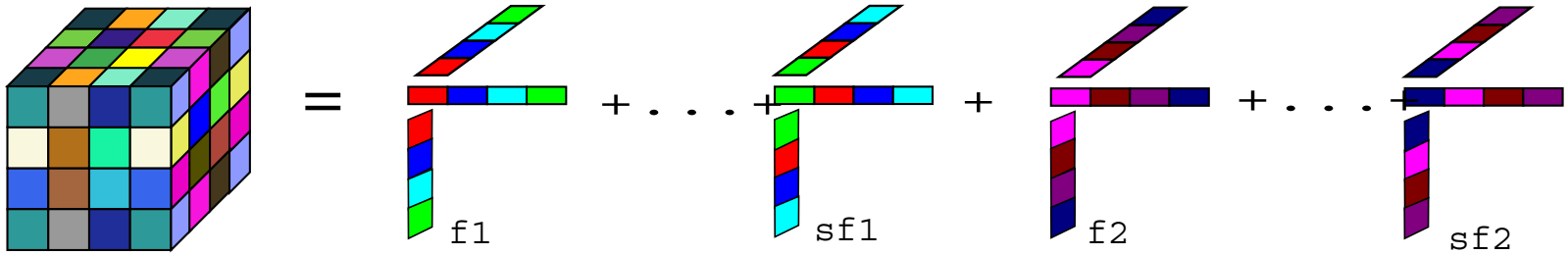
Sample Screenshot

- Infrastructure: AWS 6 nodes with 36 cores each.
- Data: PubMed 8.1 million articles, 140k vocabulary. 50 topics.
- Total runtime: 27 mins.
- https://github.com/FurongHuang/SpectralLDA-TensorSpark

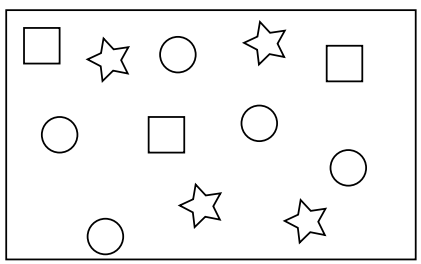


Learning Representations using Spectral Methods

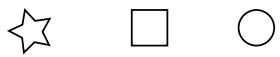
Efficient Tensor Decomposition with Shifted Components



Shift-invariant Dictionary

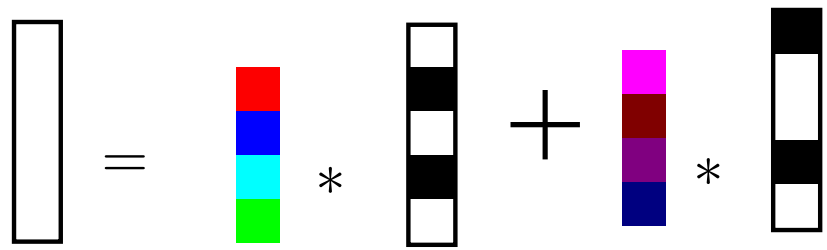


Image

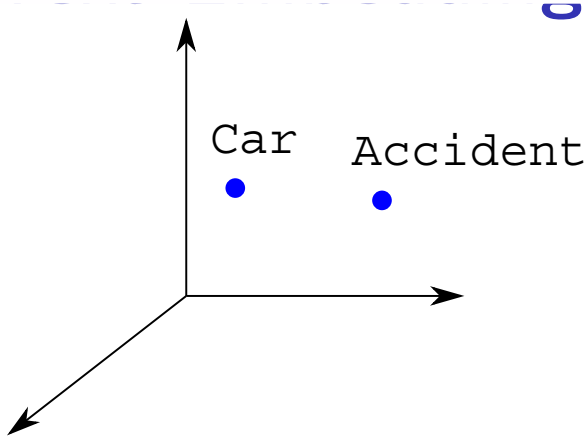


Dictionary

Convolutional Model



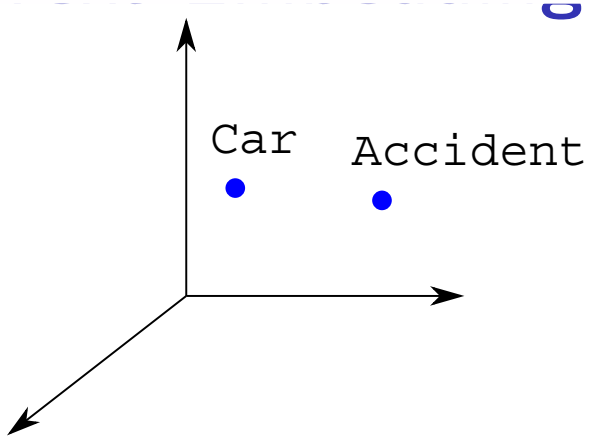
Fast Text Embeddings through Tensor Methods



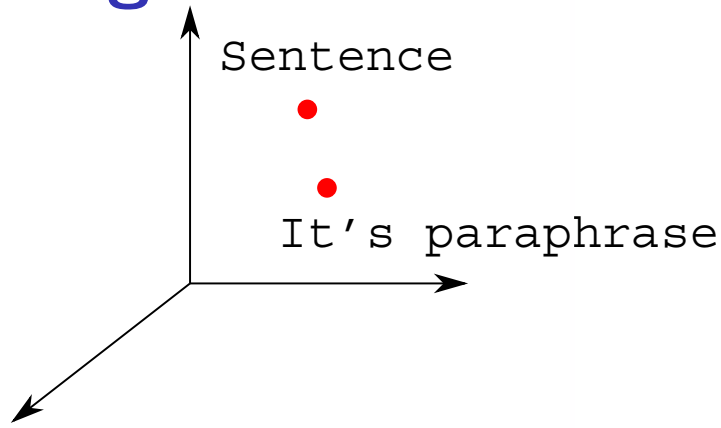
Word Embeddings

F. Huang, A. Unsupervised Learning of Word-Sequence Representations from Scratch via Convolutional Tensor Decomposition. 2016.

Fast Text Embeddings through Tensor Methods



Word Embeddings

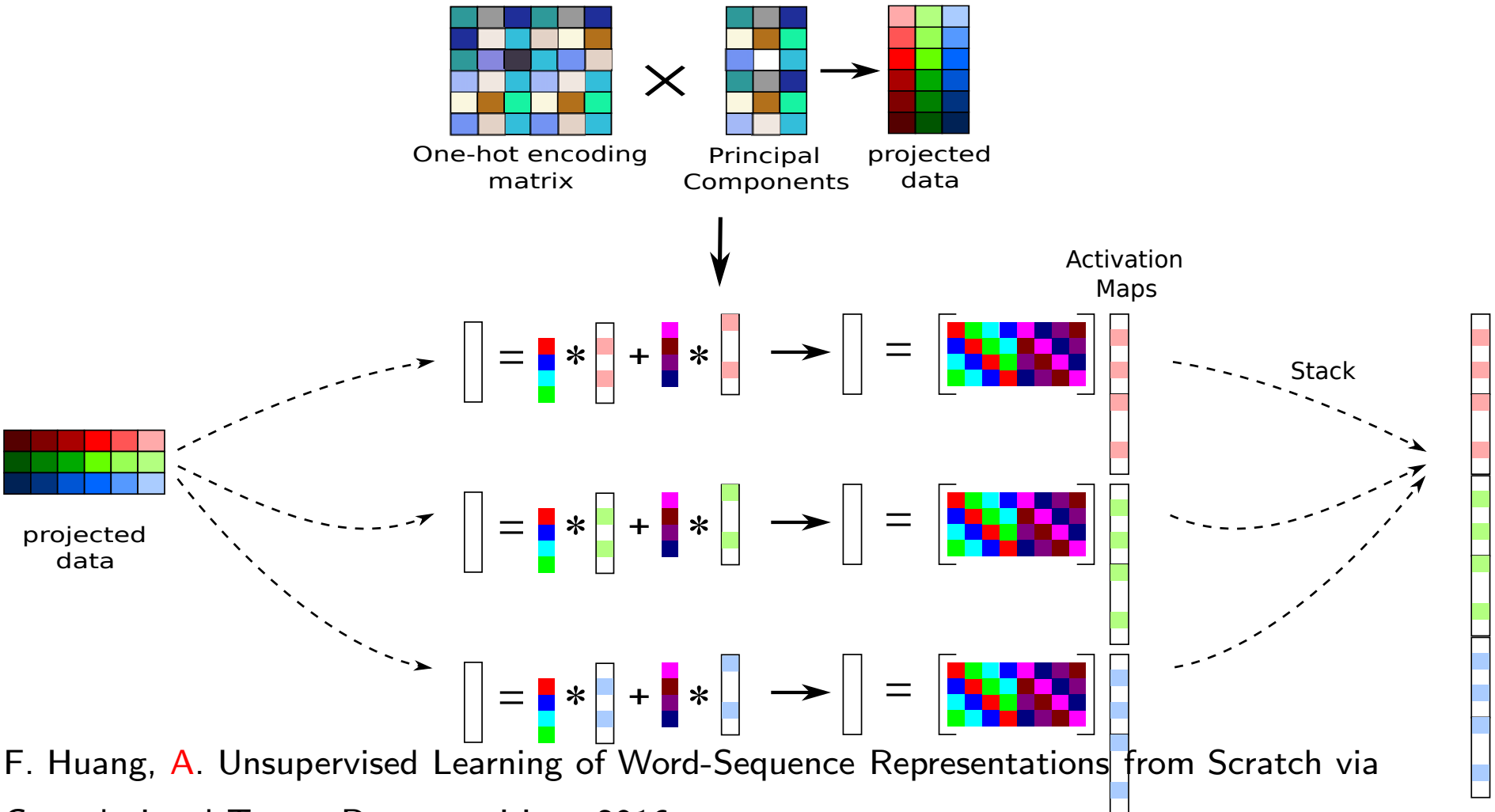


Sentence Embeddings

F. Huang, A. Unsupervised Learning of Word-Sequence Representations from Scratch via Convolutional Tensor Decomposition. 2016.

Fast Text Embeddings through Tensor Methods

Paraphrase Detection on MSR corpus with ~ 5000 Sentences



F. Huang, A. Unsupervised Learning of Word-Sequence Representations from Scratch via Convolutional Tensor Decomposition. 2016.

Fast Text Embeddings through Tensor Methods

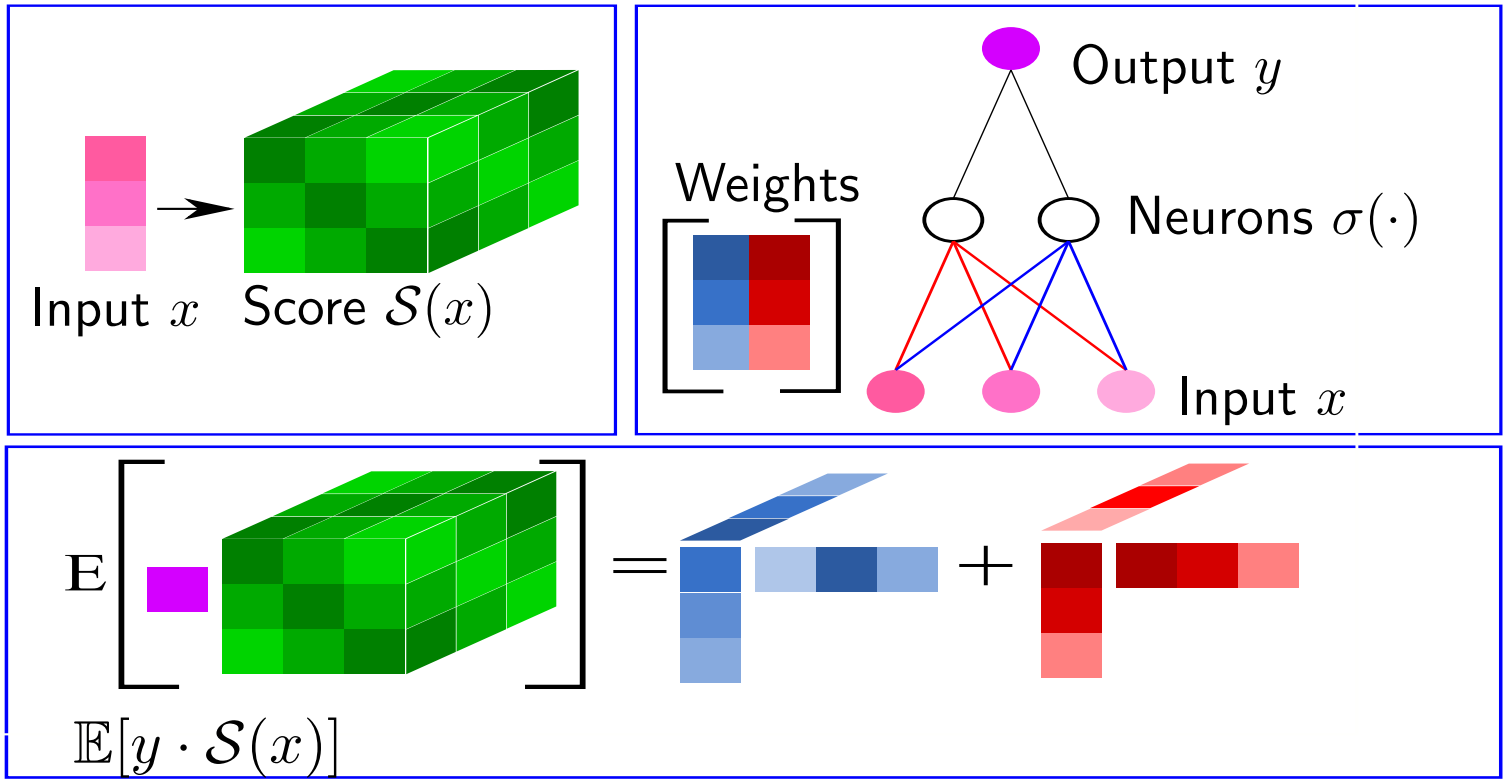
Paraphrase Detection on MSR corpus with ~ 5000 Sentences

Method	F score	No. of samples
Vector Similarity (Baseline)	75%	$\sim 4k$
Tensor (Proposed)	**81%**	$\sim 4k$
Skipthought (RNN)	82%	~ 74mil

- Unsupervised learning of embeddings.
- No outside info for tensor vs. large book corpus (74 million) for skipthought vectors.

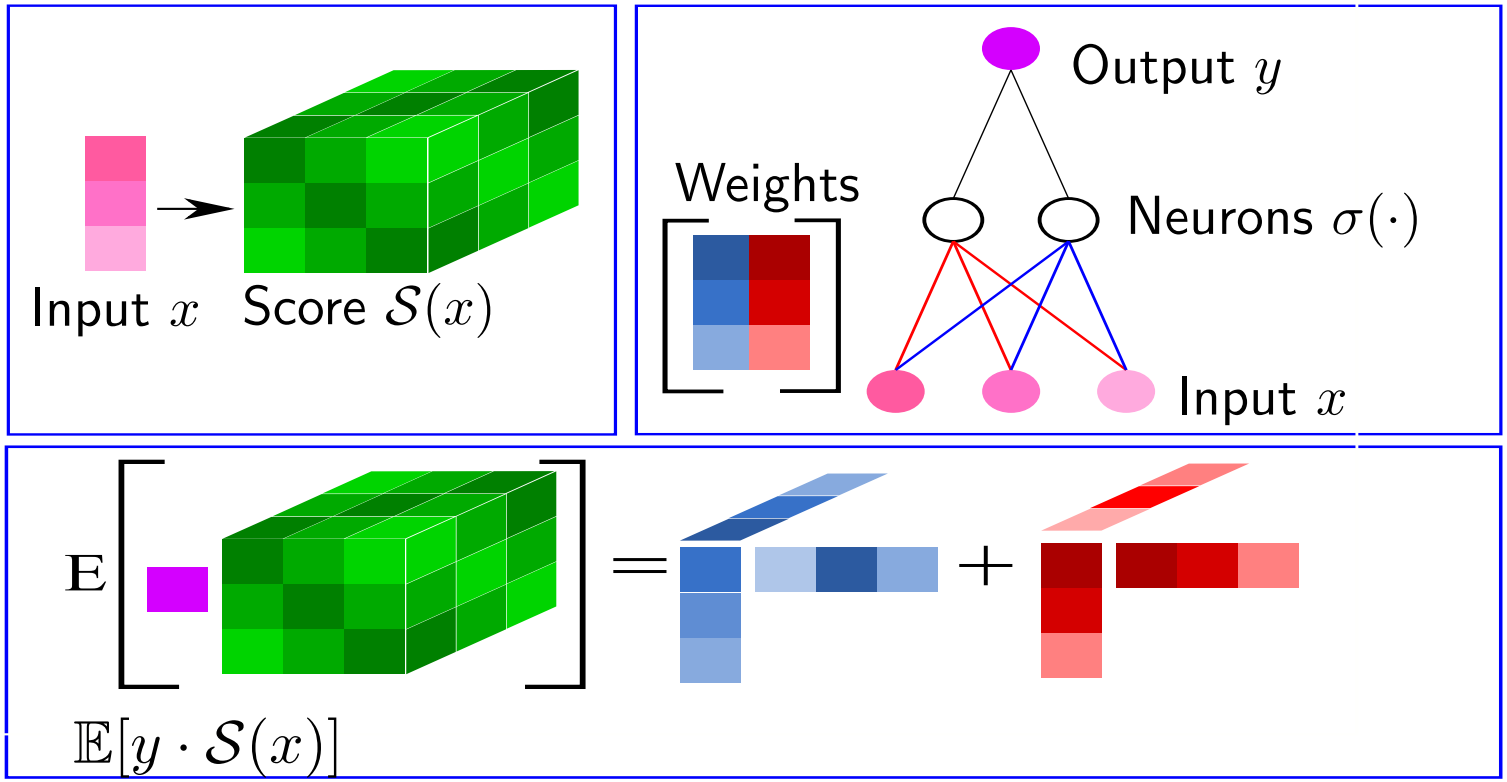
F. Huang, A. Unsupervised Learning of Word-Sequence Representations from Scratch via Convolutional Tensor Decomposition. 2016.

Training Neural Networks with Tensors

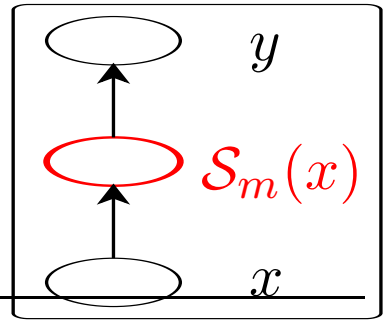


M. Janzamin, H. Sedghi, and A., "Beating the Perils of Non-Convexity: Guaranteed Training of Neural Networks using Tensor Methods," June. 2015.

Training Neural Networks with Tensors



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



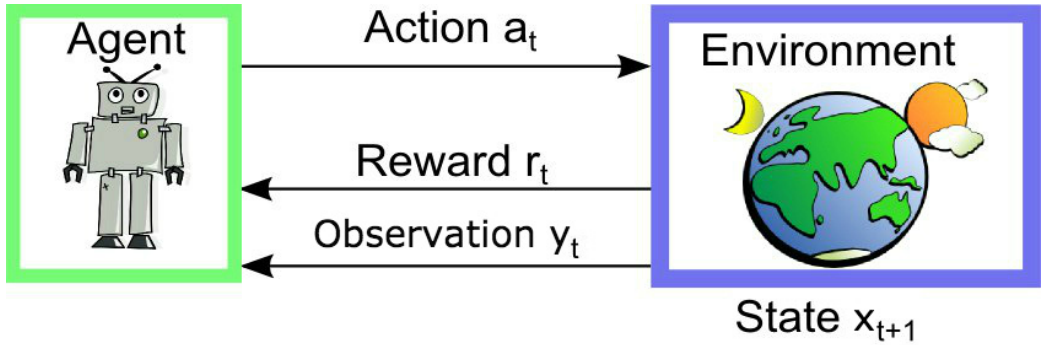
Gaussian $x \Rightarrow$ Hermite polynomials.

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

Reinforcement Learning of POMDPs

Learning in Adaptive Environments

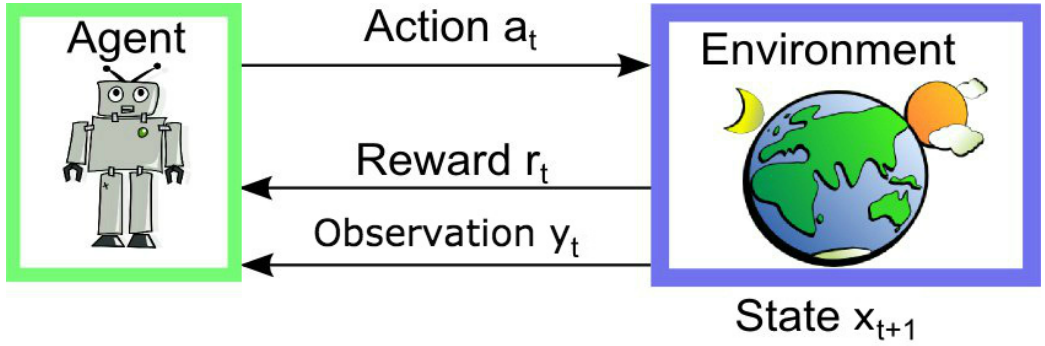
- Rewards from hidden state.
- Actions drive hidden state evolution.



Reinforcement Learning of POMDPs

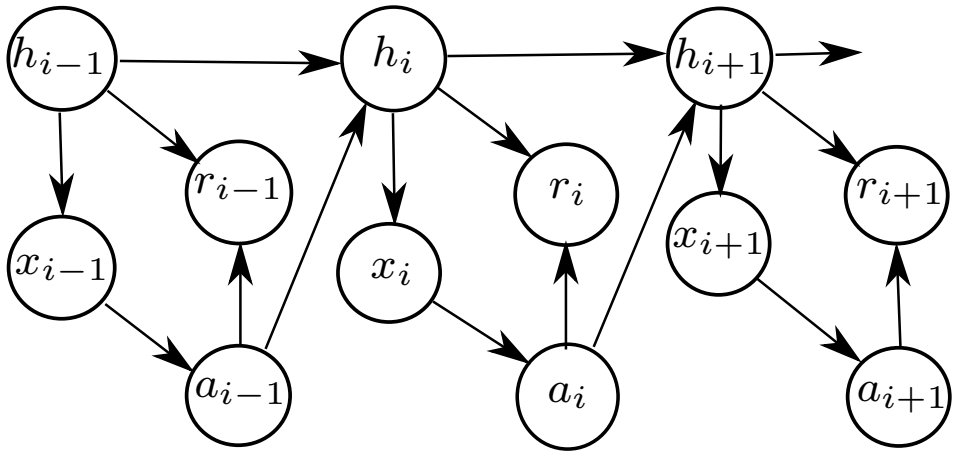
Learning in Adaptive Environments

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



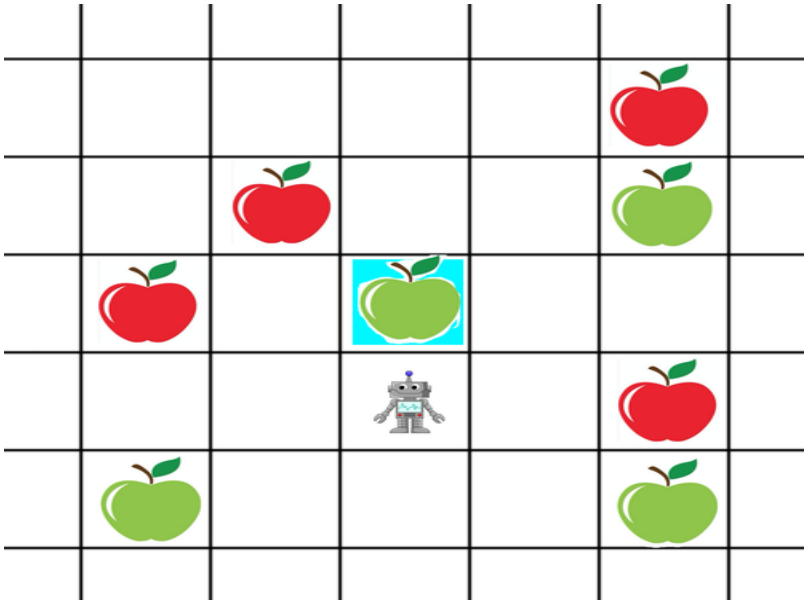
Partially Observable Markov Decision Process

Learning using tensor methods under memoryless policies

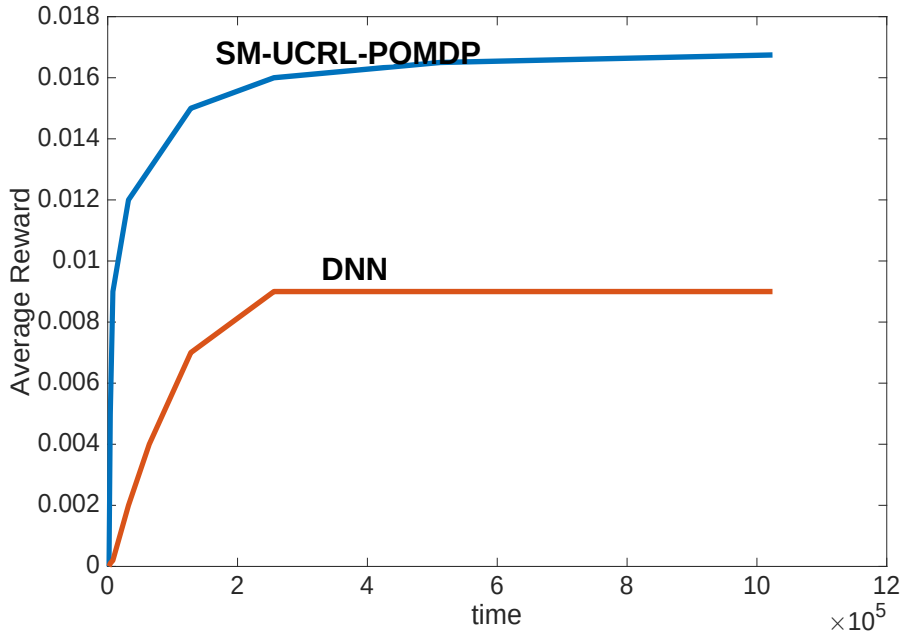


Playing Atari Game

Observation Window



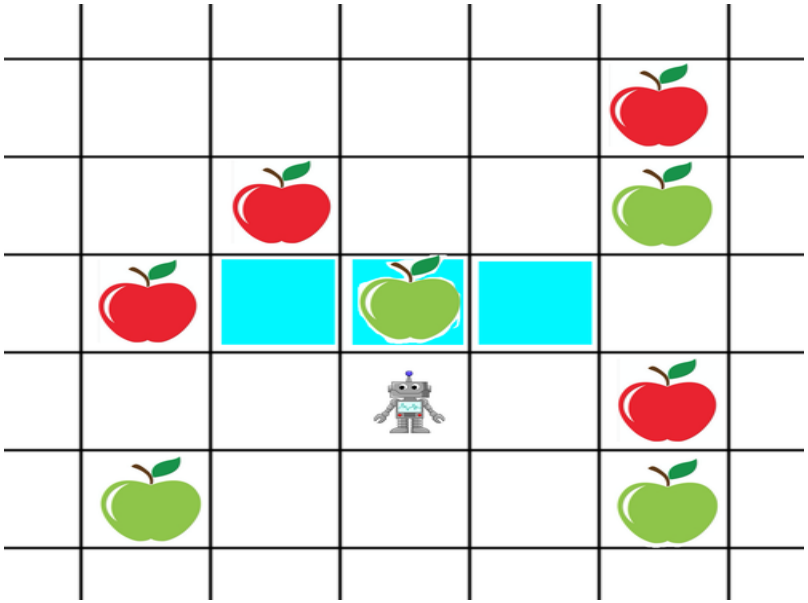
Average Reward vs. Time.



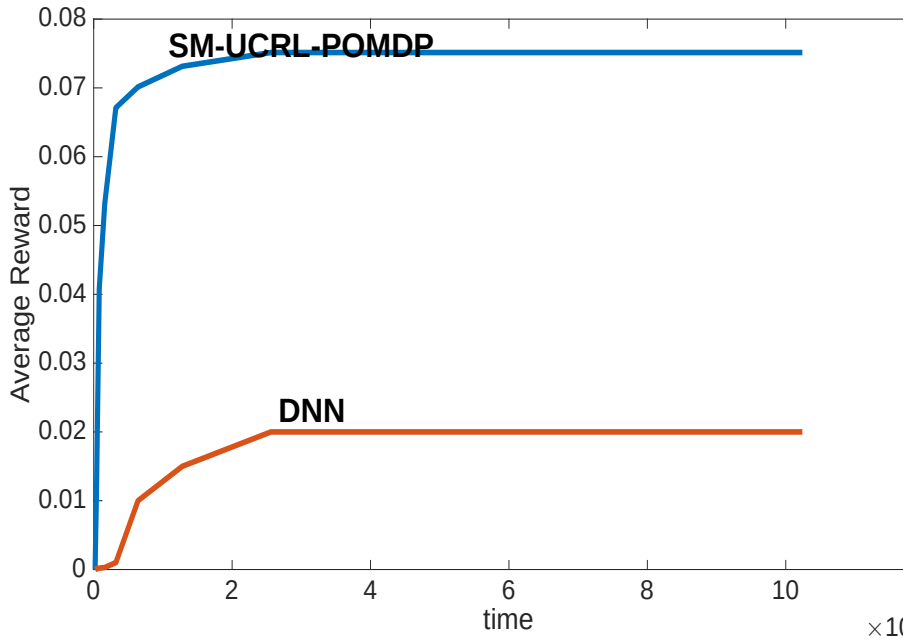
- POMDP model with 3 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 10 neurons each (trained using RmsProp).

Playing Atari Game

Observation Window



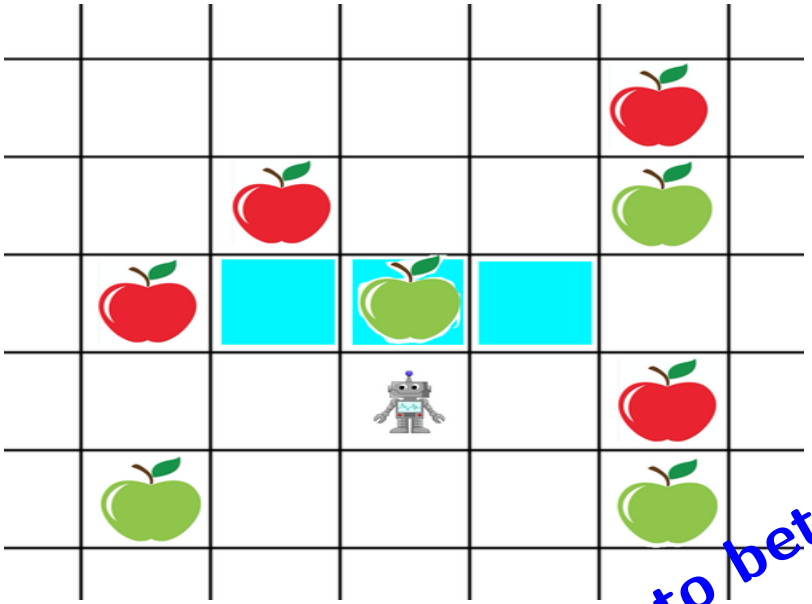
Average Reward vs. Time.



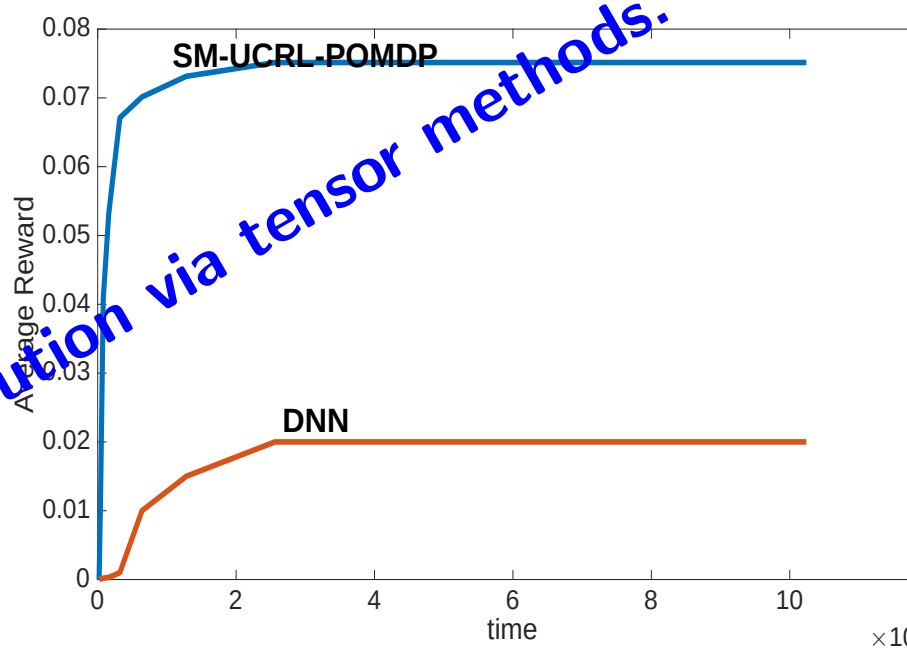
- POMDP model with 8 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 30 neurons each (trained using RmsProp).

Playing Atari Game

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).

Outline

Conclusion

Guaranteed Non-convex Optimization

- Non-convex optimization requires new theoretical frameworks.
- Escaping saddle points an important challenge for non-convex optimization.
  - Symmetry and overspecification lead to explosion of saddle points.
  - SGD can escape non-degenerate points in bounded time.
  - Efficient algorithms for escape under degeneracy.
- Matrix and tensor methods have desirable guarantees on reaching global optimum.
  - Applicable to unsupervised, supervised and reinforcement learning.
  - Polynomial computational and sample complexity.
  - Faster and better performance in practice.

Steps Forward

- How to analyze saddle point structure of well known problems, e.g. deep learning.
- Scaling up tensor methods: sketching algorithms, extended BLAS, ...
- Unified conditions on when non-convex optimization is tractable?

Resources and Research Connections

- http://www.offconvex.org/ blog.
- https://www.facebook.com/nonconvex group.
- http://newport.eecs.uci.edu/anandkumar/
- ICML and NIPS workshops. (upcoming one on thursday).

Resources and Research Connections

- http://www.offconvex.org/ blog.
- https://www.facebook.com/nonconvex group.
- http://newport.eecs.uci.edu/anandkumar/
- ICML and NIPS workshops. (upcoming one on thursday).

Collaborators

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