Dynamics of Tensor Approximation in Narrow Valleys

Martin Mohlenkamp, Nate McClatchey, Balázs Bárány, Xue Gong and Todd Young

Department of Mathematics

OHIO UNIVERSITY

Workshop on Dynamical Systems, Atlanta, August 2017

Supported by the US National Science Foundation under grant 1418787.

Thank you Shui-Nee for being a great adviser and inspiration.

Abstract

Approximating a multivariate tensor as a short sum of rank 1 tensors has many important potential uses, but common optimization algorithms applied to this problem can exhibit extremely slow progress in regions known informally as "swamps".

- We have identified one possible type of swamp as a narrow valley in the optimization landscape.
- We analyze the dynamics of one important class of algorithms, in typical valleys and identify several interesting and potentially useful properties.
- We have located robust narrow valleys in the optimization landscape of tensor problems.

Outline

1. The tensor approximation problem and alternating least squares (ALS)
2. Alternating methods in typical valleys
3. Valleys in tensor approximations problems

Approximation by Sums of Separable (Rank 1) Tensors

Consider approximation of a tensor T of the form

$$T(j_1, j_2, \ldots, j_d) \approx G(j_1, \ldots, j_d) = \sum_{l=1}^{r} G^l(j_1, \ldots, j_d) = \sum_{l=1}^{r} \bigotimes_{i=1}^{d} G_i^l$$

d is the number of "directions". If d is large, storing T is prohibitive.

r is the rank of the approximation.

Each G^l is a tensor product of d vectors.

The ability to construct such approximations enables many algorithms in high dimensions.

Alternating Least Squares

ALS is an implementation of Block Coordinate Descent (BCD) in the context of tensor approximation.
Considering:

$$T \approx G = \sum_{l=1}^{r} G^l = \sum_{l=1}^{r} \bigotimes_{i=1}^{d} G_i^l \,,$$

if one optimizes w.r.t. one direction i at a time, each step becomes a linear Least Squares problem. Alternating between directions is called ALS.

Each step is very efficient and accurate.

But, ...

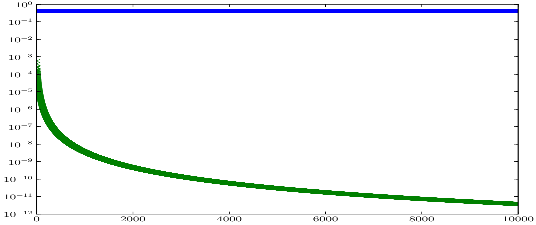
Current Approximation Algorithms are Unsatisfactory

Sometimes everything is fine, but sometimes unpleasant things happen.

ALS behavior:

Terminal swamp: 10000 iterations and the error is still decreasing 10^{-11} at each step.

Error and Δ Error



Current Approximation Algorithms are Unsatisfactory

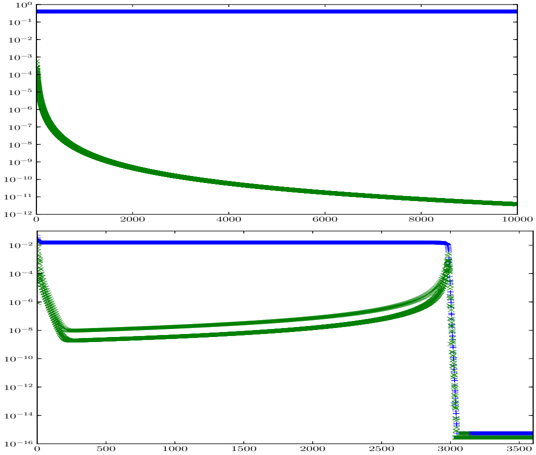
Sometimes everything is fine, but sometimes unpleasant things happen.

ALS behavior:

Terminal swamp: 10000 iterations and the error is still decreasing 10^{-11} at each step.

Transient swamp: 3000 iterations with error decreasing like 10^{-8} at each step, then rapid convergence.

Error and Δ Error



Goals of this Project

1. Construct an efficient, robust, and otherwise great approximation algorithm.

Goals of this Project

1. Construct an efficient, robust, and otherwise great approximation algorithm.
0. Understand why current algorithms have trouble.

Goals of this Project

1. Construct an efficient, robust, and otherwise great approximation algorithm.
0. Understand why current algorithms have trouble.
-1. Understand why current algorithms have trouble on a few interesting examples.

Goals of this Project

1. Construct an efficient, robust, and otherwise great approximation algorithm.
0. Understand why current algorithms have trouble.
-1. Understand why current algorithms have trouble on a few interesting examples.
-2. Understand the approximation problem itself on a few interesting examples.

Today we work toward goals -1 and -2.

A "typical" valley

Narrow valleys are known to cause problems for optimization.
Suppose that $\mathbf{v}$ is a unit vector and $\Delta_{\mathbf{v}} = \mathrm{span}(\mathbf{v})$.
Consider a quadratic error function for which $\Delta_{\mathbf{v}}$ is the valley floor:

$$E = C + \frac{a}{2} d^2(\mathbf{x}, \Delta_{\mathbf{v}}) + \epsilon \mathbf{v} \cdot \mathbf{x} = \frac{a}{2} \left(|\mathbf{x}|^2 - (\mathbf{v} \cdot \mathbf{x})^2 \right) + \epsilon \mathbf{v} \cdot \mathbf{x}.$$

$$\nabla E = a\mathbf{x} - a(\mathbf{v} \cdot \mathbf{x})\mathbf{v} + \epsilon \mathbf{v}.$$

On $\Delta_{\mathbf{v}}$, we have $\nabla E = \epsilon \mathbf{v}$. Further,

$$H = \nabla^2 E = aI - a\mathbf{v}\mathbf{v}^T.$$

We see that $H\mathbf{v} = \mathbf{0}$ and if $\mathbf{u} \perp \mathbf{v}$ then $H\mathbf{u} = a\mathbf{u}$.
a measures the steepness of the sides of the valley.

Alternating method (BCD) with partitioned variables

Suppose that $\mathbf{v}$ and $\mathbf{x}$ are partitioned into d compatible sets of variables:

$$\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_d)^T \quad \text{and} \quad \mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d)^T,$$

where $\dim \mathbf{x}_i = \dim \mathbf{v}_i$.
The gradient ∇E in partitioned variables reads:

$$\nabla E = a \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_d \end{pmatrix} - a(\mathbf{x}_1 \cdot \mathbf{v}_1 + \mathbf{x}_2 \cdot \mathbf{v}_2 + \ldots + \mathbf{x}_d \cdot \mathbf{v}_d) \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_d \end{pmatrix} + \epsilon \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_d \end{pmatrix}$$

and so $\partial E / \partial \mathbf{x}_1 = 0$ is:

$$a\mathbf{x}_1 - a(\mathbf{x}_1 \cdot \mathbf{v}_1 + \mathbf{x}_2 \cdot \mathbf{v}_2 + \ldots + \mathbf{x}_d \cdot \mathbf{v}_d)\mathbf{v}_1 + \epsilon \mathbf{v}_1 = 0.$$

ALS $\rightarrow$ recursion on coefficients

This has a solution $\mathbf{x}_1 = c_1^1 \mathbf{v}_1$, where

$$c_1^1 = \frac{1}{1 - |\mathbf{v}_1|^2} \left(\mathbf{x}_2 \cdot \mathbf{v}_2 + \ldots + \mathbf{x}_d \cdot \mathbf{v}_d - \frac{\epsilon}{a} \right).$$

After one full round of ALS we will have:

$$\mathbf{x}^1 = (c_1^1 \mathbf{v}_1, c_2^1 \mathbf{v}_2, \ldots, c_d^1 \mathbf{v}_d).$$

ALS is thereafter just a recurrence on $(c_1^k, c_2^k, \ldots, c_d^k)$:

$$c_i^{k+1} = \frac{|\mathbf{v}_1|^2 c_1^{k+1} + \ldots + |\mathbf{v}_{i-1}|^2 c_{i-1}^{k+1} + |\mathbf{v}_{i+1}|^2 c_{i+1}^k + \ldots + |\mathbf{v}_d|^2 c_d^k - \frac{\epsilon}{a}}{1 - |\mathbf{v}_i|^2}.$$

Example $d = 2$

$\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2)$, $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$, $|\mathbf{v}| = 1$.
After the first full round of ALS, $\mathbf{x}^k = (c_1^k \mathbf{v}_1, c_2^k \mathbf{v}_2)$, where:

$$c_1^k = c_2^{k-1} - \frac{1}{\beta^2}\frac{\epsilon}{a},$$

$$c_2^k = c_1^k - \frac{1}{\alpha^2}\frac{\epsilon}{a} = c_2^{k-1} - \frac{1}{\alpha^2\beta^2}\frac{\epsilon}{a}.$$

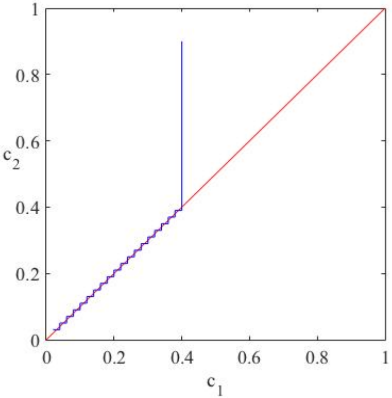
Each full pass moves a distance:

$$|\Delta \mathbf{v}| = \frac{1}{\alpha^2\beta^2}\frac{\epsilon}{a} = \frac{1}{\cos^2\theta \sin^2\theta}\frac{\epsilon}{a} = 4\csc^2 2\theta \frac{\epsilon}{a}.$$

Here $\alpha = |\mathbf{v}_1| = \cos\theta$ and $\beta = |\mathbf{v}_2| = \sin\theta$.

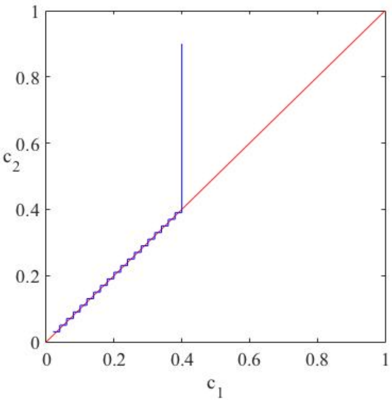
$d = 2$

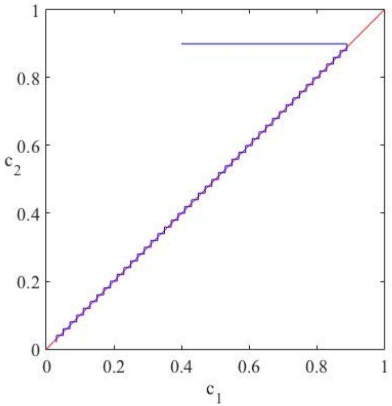
Iterations Zig-Zig on lines $\sim \frac{\epsilon}{a}$ from valley floor.



$d = 2$

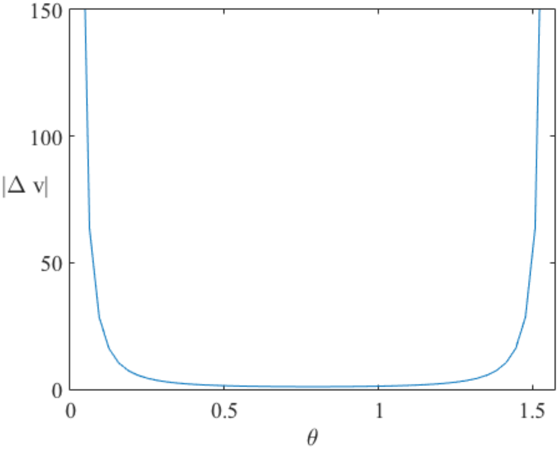
Iterations Zig-Zig on lines $\sim \frac{\epsilon}{a}$ from valley floor.





The direction you optimize first may matter a lot!

Dependence on the ‘Angle’



Graph of $\csc^2 2\theta$. Progress is slow except for shallow angles θ.

Some Lessons from $d = 2$

- Iterations zig-zag along lines $\sim \epsilon/a$ from the valley floor.
- Iterations move a distance $\sim \epsilon/a$.
- Iterations move a distance proportional to $\csc^2 2\theta$
- A greedy first step is important.

Example $d = 3$

$\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, $\alpha = |\mathbf{v}_1|$, $\beta = |\mathbf{v}_2|$, $\gamma = |\mathbf{v}_3|$.

After the first full step, $\mathbf{x}^k = (c_1^k \mathbf{v}_1, c_2^k \mathbf{v}_2, c_3^k \mathbf{v}_3)$, where:

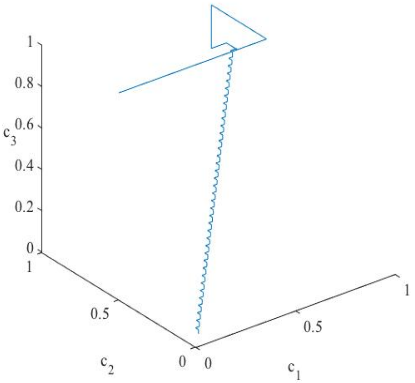
$$
\begin{aligned}
c_1^{k+1} &= \tfrac{1}{1-\alpha^2} \left(\beta^2 c_2^k + \gamma^2 c_3^k - \tfrac{\epsilon}{a} \right). \\
c_2^{k+1} &= \tfrac{1}{1-\beta^2} \left(\alpha^2 c_1^{k+1} + \gamma^2 c_3^k - \tfrac{\epsilon}{a} \right). \\
c_3^{k+1} &= \tfrac{1}{1-\gamma^2} \left(\alpha^2 c_1^{k+1} + \beta^2 c_2^{k+1} - \tfrac{\epsilon}{a} \right).
\end{aligned}
$$

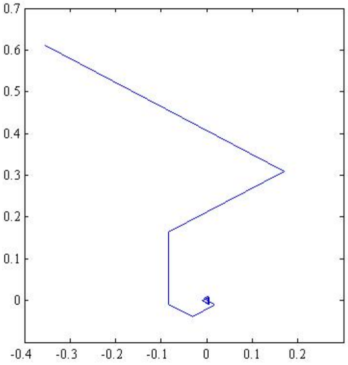
This can be solved. Full iterations move by: $\dfrac{1}{\alpha^2\beta^2 + \alpha^2\gamma^2 + \beta^2\gamma^2} \dfrac{\epsilon}{a}$

This is small unless *TWO* of α, β, γ are small.

Example $d = 3$

Iterations converge to a nbhd of the valley floor.





The rate of attraction to the valley floor is: $\frac{\alpha^2\beta^2\gamma^2}{(1-\alpha^2)(1-\beta^2)(1-\gamma^2)}$.

Some Lessons from $d = 3$

- Iterations are attracted to a nbhd of the valley floor at a rate independent of ϵ and a.
- Iterations zig-zag $\sim \epsilon/a$ from the valley floor.
- Full Iterations move a distance $\sim \epsilon/a$.
- As dimensions grow almost all directions are 'diagonal' for ALS.

Are there narrow valleys in tensor approximation problems?

Generic Rank-2 Target Tensor

Applying a separable unitary change of basis, choice of orientation, and normalization, any rank-2 tensor can be put in the form

$$T = C_{z,\bar{\phi}} \left(\bigotimes_{i=1}^{d} \mathbf{u}(0) + z \bigotimes_{i=1}^{d} \mathbf{u}(\phi_i) \right)$$

where

- $\mathbf{u}(\phi) = \cos(\phi)\mathbf{e}_1 + \sin(\phi)\mathbf{e}_2$ for orthonormal basis vectors $\{\mathbf{e}_1, \mathbf{e}_2\}$,
- $0 \leq \phi_i \leq \pi/2$ controls the angle of factors in the second summand,
- $|z| \leq 1$ controls the relative size and direction of the two terms, and
- the scalar makes T have norm 1.

These transformations commute with, and don't affect, ALS.

The Approximation

Consider approximations with the correct number of terms (rank 2):

$$G_2 = a \bigotimes_{i=1}^{d} \mathbf{u}(\alpha_i) + b \bigotimes_{i=1}^{d} \mathbf{u}(\beta_i) \,.$$

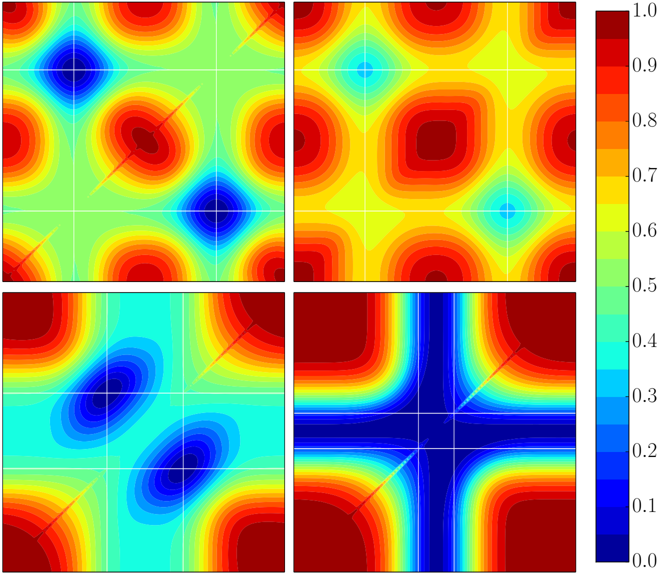
Regularized error:

$$\begin{aligned} E_\lambda(G) = E_\lambda(G^1, \ldots, G^r) &= \| T - G \|^2 + \lambda \sum_{l=1}^{r} \| G^l \|^2 \\ &= \| T - G_2 \|^2 + \lambda (a^2 + b^2) \,. \end{aligned}$$

When $\lambda = 0$ this is ordinary least-squares error, while $\lambda > 0$ ensures the problem is well-posed and controls loss-of-significance error.

Can make a and b 'fast variables' and eliminate them from the analysis. *Plot the error landscape in the symmetric case* $\phi_i = \phi$, $\alpha_i = \alpha$, $\beta_i = \beta$.

$$E_\lambda(G_2) \text{ for } \begin{array}{cc} d = 6, & z = 1, \\ \alpha_i = \alpha, & \beta_i = \beta, \end{array} \quad \phi, \lambda = \begin{bmatrix} \frac{\pi}{2}, 0 & \frac{\pi}{2}, \frac{1}{2} \\ 0.27\pi, 0 & \frac{\pi}{8}, 0 \end{bmatrix}.$$



An Essential Singularity

$\alpha_1 = \cdots = \alpha_d$	$\alpha_2 = \cdots = \alpha_d$	$\alpha_2 = \cdots = \alpha_d$	$\alpha_2 = \cdots = \alpha_d$
$(\vec{\alpha}, \vec{\beta}) = (\alpha\vec{1}, \beta\vec{1})$	and $\vec{\beta} \to \vec{\alpha}$	and $\vec{\beta} = \phi - \vec{\alpha}$	and $\vec{\beta} = \phi - \vec{\alpha}$
$E_0(G_2) \in [0, 1]$	$[0, 0.1]$	$[0, 0.1]$	$[0, 0.1]$

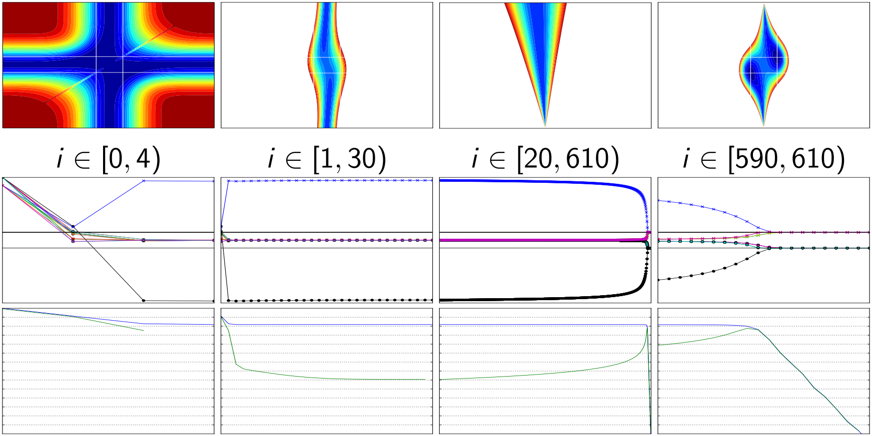


Illustration of a transient swamp flow path for $(d, z, \phi, \lambda) = (6, 1, \pi/8, 0)$

Conclusions

- Diagonally oriented valleys occur.
- Essential singularities exist on the boundary and tend to attract orbits.
- Progress in a valley depends on the gradient and transverse Hessian.
- Iterations are attracted to the valley at a rate independent of ϵ & a.
- Greedy first steps are important.
- Other implications for algorithm development are under consideration.
- A visualization webpage is available at `http://www.ohiouniversityfaculty.com/mohlenka/DSoTA/visual/`

Thank you for your attention!