Guarantees for Alternating Least Squares in Overparameterized Tensor Decomposition

Vaidehi Srinivas

Dionysis Arvanitakis

Aravindan Vijayaraghavan

Northwestern University, Computer Science

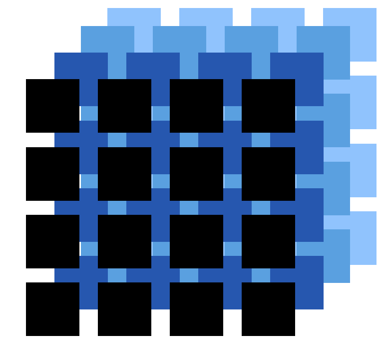
2025 Optimization Unplugged Workshop @ EPFL

Tensor Decomposition

CP (canonical polyadic) decomposition:

Given an **order-3** tensor $T \in \mathbb{R}^{n \times n \times n}$, for the minimum possible **rank** $r \in \mathbb{N}$, find $\underbrace{U, V, W}_{\text{factor matrices}} \in \mathbb{R}^{n \times r}$ such that

$$T = \sum_{i=1}^{r} u_i \otimes v_i \otimes w_i.$$



($\otimes$ the **tensor** or **outer product**)

TL;DR:

- NP-hard to compute rank in the worst-case
- Large body of work designing and analyzing algorithms for non-worst-case instances

So why do iterative optimization methods work so well in practice?

Why Tensor Decomposition?

Interesting on its own:

- Method of moments for latent variable models
  - Tensor components can correspond to components of mixture distributions, e.g. mixtures of Gaussians [Ge Huang Kakade '15] [Bafna Hsieh Kothari Xu '22]
- Scientific applications: mixture problems in chemistry and physics

Testbed for nonconvex optimization:

- Can formulate learning neural networks as a tensor problem [Ge Lee Ma '18]
  - For $f(x) = a^\top \sigma(W^\top x)$, σ is ReLU, bias is 0, Gaussian x, m number of hidden neurons:

$$L(\widetilde{a}, \widetilde{W}) = \sum_{k \geq 2, k \text{ even}} \frac{((k-3)!!)^2}{2\pi k} \left\| \sum_{i=1}^{m} a_i w_i^{\otimes k} - \sum_{i=1}^{m} \widetilde{a}_i \widetilde{w}_i^{\otimes k} \right\|_F^2$$

- "Simple" hard problem because of multilinearity

Non-convexity

Order-3 Tensors: For $T \in \mathbb{R}^{n \times n \times n}$,

$$\min_{X,Y,Z \in \mathbb{R}^{n \times k}} \left\| T - \sum_{i=1}^{k} x_i \otimes y_i \otimes z_i \right\|_F^2 .$$

Matrices (order-2 tensors): For $M \in \mathbb{R}^{n \times n}$,

$$\min_{X,Y \in \mathbb{R}^{n \times k}} \left\| M - XY^\top \right\|_F^2 .$$

Convexity in the space of matrices $UV^\top$ does not correspond to convexity in the space of factors (X, Y)!

- For matrices, no spurious second order critical points! (**benign** non-convexity)
- Third order tensors are not so nice… (e.g. local minima exist [Wang Wu Lee Ma Ge '20])
- Tensor decompositions tend to be **unique**

Poly-time Algorithms for Exact Tensor Decomposition

Worst case:

- NP-hard to compute the CP-rank [Hillar, Lim '09]
- Can be badly behaved (border-rank issues) and non-stable

Non-worst case:

- For random tensor $T = \sum_{i=1}^{r} u_i^{\otimes 3}$, with u_is drawn independently from the unit sphere, can recover factors for rank $r < n^{2/3}/\text{polylog}(n)$, via sum-of-squares algorithm [Ge Ma '15]
- For tensor $T = \sum_{i=1}^{r} u_i \otimes v_i \otimes w_i$ with **generic** factors, $U, V, W \in \mathbb{R}^{n \times r}$, can recover decomposition for $k \leq 2n - \varepsilon$ via "Koszul-Young Flattenings." [Koiran '24][Kothari Moitra Wein '24]

Takeaway: Exact tensor decomposition is complicated

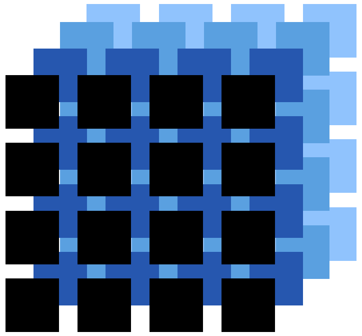
Practical Methods and Guarantees

Gradient Descent:

- For iteratively finding one component at a time, "better than random" initialization converges to good minima [Ge Ma '17]
- For standard least-squares objective, can show modified gradient descent on least-squares objective can recover tensor with **overparameterization** $k = r^{7.5 \log n}$ [Wang Wu Ge Lee Ma '20]
    - For rank-r tensor $T = \sum_{i=1}^{r} u_i \otimes v_i \otimes w_i$, fit **rank-$k$** model $\sum_{i=1}^{k} x_i \otimes y_i \otimes z_i$
    - Can find $k = r^2$ decomposition with SVD
    - Analogue of overparameterization in other settings like neural networks

Alternating Least-Squares (ALS):

- Most common method in practice
- Focus of this work!



Alternating Least-Squares (ALS)

For input tensor $T \in \mathbb{R}^{n \times n \times n}$ and **overparameterized rank** $k \in \mathbb{N}$, objective is

$$\min_{X,Y,Z \in \mathbb{R}^{n \times k}} \left\| T - \sum_{j=1}^{k} x_j \otimes y_j \otimes z_j \right\|_F^2 = \mathrm{Obj}(X, Y, Z)$$

- Initialize $X, Y, Z \sim \mathcal{N}(0,1)^{n \times k}$ **randomly**
- Alternately update each factor matrix by solving **linear** system w.r.t. that mode.

Let $T = \sum_{i=1}^{r} a_i \otimes a_i \otimes a_i$ for unknown $A \in \mathbb{R}^{n \times r}$. For Y, Z fixed, X update is

$$\operatorname*{argmin}_X \mathrm{Obj}(X, Y, Z) = \operatorname*{argmin}_X \left\| A(A \odot A)^\top - \underline{X(Y \odot Z)^\top} \right\|_F^2$$

$$= A(A \odot A)^\top (Y \odot Z)^{\top \dagger}$$

† the Moore-Penrose pseudoinverse

Notation just reformatting slices of tensor as rows of a matrix

Khatri-Rao Product:

For $A, B \in \mathbb{R}^{n \times r}$, $A \odot B = \begin{bmatrix} & \uparrow & \\ \ldots & \mathrm{vec}(a_i \otimes b_i) & \ldots \\ & \downarrow & \end{bmatrix} \in \mathbb{R}^{n^2 \times r}$.

Not a nice matrix product! More of a shorthand

ALS a.k.a. Block-Coordinate Descent

For input tensor $T \in \mathbb{R}^{n \times n \times n}$ and **overparameterized rank** $k \in \mathbb{N}$, objective is

$$\min_{X,Y,Z \in \mathbb{R}^{n \times k}} \left\| T - \sum_{j=1}^{k} x_j \otimes y_j \otimes z_j \right\|_F^2 = \mathsf{Obj}(X, Y, Z)$$

- ALS alternately sets

$$X^{(t+1)} \leftarrow \min_X \mathsf{Obj}\left(X, Y^{(t)}, Z^{(t)}\right)$$
$$Y^{(t+1)} \leftarrow \min_Y \mathsf{Obj}\left(X^{(t+1)}, Y, Z^{(t)}\right)$$
$$Z^{(t+1)} \leftarrow \min_Z \mathsf{Obj}\left(X^{(t+1)}, Y^{(t+1)}, Z\right)$$

- **Coordinate descent** treating each factor matrix X, Y, Z as a block
- How tensor (CP) decomposition is implemented in many applications and libraries

Result

Informal Theorem [Arvanitakis **S.** Vijayaraghavan '25]

Given a rank-r tensor $T \in \mathbb{R}^{n\times n\times n}$, with unknown factorization $T = \sum_{i=1}^{r} a_i \otimes b_i \otimes c_i$, with mildly conditioned the factor matrices A, B, C,

a parallel variant of Alternating Least Squares (ALS) with $k = \Omega(r^2)$ factors and random initialization converges to a global minimum X, Y, Z, i.e.,

$$\left\| T - \sum_{i=1}^{k} x_i \otimes y_i \otimes z_i \right\|_F^2 = 0,$$

with high probability.

Warm-Up: Matrix Decomposition

Let $T = AA^\top$ for unknown $A \in \mathbb{R}^{n\times r}$.

$$\min_{X,Y\in\mathbb{R}^{n\times r}} \left\| T - \sum_{i=1}^{r} x_i \otimes y_i \right\|_F^2 = \min_{X,Y\in\mathbb{R}^{n\times k}} \left\| AA^\top - XY^\top \right\|_F^2 = \mathsf{Obj}(X, Y).$$

Initialize: $X^{(0)}, Y^{(0)} \sim \mathcal{N}(0,1)^{n\times r}$

Step 1a: $X^{(1)} \leftarrow (AA^\top)(Y^{(0)})^{\top\dagger}$

Step 1b: $Y^{(1)} \leftarrow (AA^\top)(X^{(1)})^{\top\dagger}$

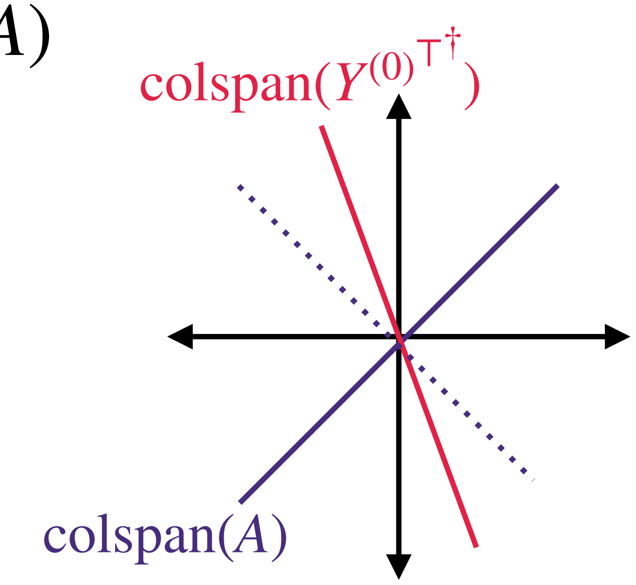
Converge!

Proof:

- $\mathrm{colspan}(X^{(1)}) \subseteq \mathrm{colspan}(A)$
- $\mathrm{colspan}\big((Y^{(0)})^{\top\dagger}\big)$ fully random r-dim. space

  $\implies \mathrm{colspan}(X^{(1)})$ has dimension r
- $\mathrm{colspan}(X^{(1)}) = \mathrm{colspan}(A)$

$$X^{(1)}Y^{(1)\top} = X^{(1)}\, X^{(1)\dagger} AA^\top$$
$$= \Pi_{X^{(1)}} AA^\top$$
$$= AA^\top$$



Trouble with Tensors

Khatri-Rao Product:

$$A \odot B = \begin{array}{c} \xleftrightarrow{\; r \text{ columns} \;} \\ \begin{bmatrix} \uparrow \\ \ldots \mathrm{vec}(a_i \otimes b_i) \ldots \\ \downarrow \end{bmatrix} \end{array}$$

Let $T = \sum_{i=1}^{r} a_i \otimes a_i \otimes a_i$ for unknown $A \in \mathbb{R}^{n \times r}$.

$$\min_{X,Y,Z \in \mathbb{R}^{n \times r}} \left\| T - \sum_{i=1}^{r} x_i \otimes y_i \otimes z_i \right\|_F^2 = \mathrm{Obj}(X, Y).$$

Initialize: $X^{(0)}, Y^{(0)}, Z^{(0)} \sim \mathcal{N}(0,1)^{n \times r}$

Step 1a: $X^{(1)} \leftarrow A(A \odot A)^\top (Y^{(0)} \odot Z^{(0)})^{\top\dagger}$

Step 1b: $Y^{(1)} \leftarrow A(A \odot A)^\top (X^{(1)} \odot Z^{(0)})^{\top\dagger}$

Step 1c: $Z^{(1)} \leftarrow A(A \odot A)^\top (X^{(1)} \odot Y^{(1)})^{\top\dagger}$

Converge?

Proof???

- $\mathrm{colspan}(X^{(1)}), \mathrm{colspan}(Y^{(1)}) \subseteq \mathrm{colspan}(A)$
- $\mathrm{colspan}(X^{(1)}), \mathrm{colspan}(Y^{(1)})$ have dimension r w.h.p.
- $\mathrm{colspan}(X^{(1)}) = \mathrm{colspan}(Y^{(1)}) = \mathrm{colspan}(A)$
- $\mathrm{colspan}(X^{(1)} \odot Y^{(1)}) = \mathrm{colspan}(A \odot A)$???

No! The Khatri-Rao product isn't so nice

Random Tensors

Choice of basis really matters for the Khatri-Rao product!

Random Matrix Fact: The space of matrices spanned by a few random rank-1 matrices will not contain any other rank-1 matrices.

For $X^{(1)}, Y^{(1)}, A \in \mathbb{R}^{n \times r}$, compare:

$$X^{(1)} \odot Y^{(1)} \quad \text{vs.} \quad A \odot A$$

- View columns of Khatri-Rao product as vectorized **rank-1 matrices**
- Even if $\mathrm{colspan}(X^{(1)}) = \mathrm{colspan}(Y^{(1)}) = \mathrm{colspan}(A)$,
  for random looking $X^{(1)}$ and $Y^{(1)}$, $\mathrm{colspan}(X^{(1)} \odot Y^{(1)})$ will not contain $a_1 \otimes a_1$
- Makes sense, because recovering span of A does not suffice for tensor decomposition!
  - Because of uniqueness, for this to work, $X^{(1)}$, $Y^{(1)}$ must recover **vectors** of A, not just the span!
  - (Unlike for matrices)

Khatri-Rao Product:

$$A \odot B = \begin{array}{c} \xleftrightarrow{\quad r \text{ columns} \quad} \\ \begin{bmatrix} & \uparrow & \\ \ldots & \mathrm{vec}(a_i \otimes b_i) & \ldots \\ & \downarrow & \end{bmatrix} \end{array}$$

Enter, the Kronecker Product

Khatri-Rao product doesn't behave nicely because it doesn't correspond to the **tensor product** of spaces:

$$\mathrm{colspan}(A \otimes B) = \mathrm{span}\left\{a \otimes b : a \in \mathrm{colspan}(A), b \in \mathrm{colspan}(B)\right\}.$$

Kronecker Product:

For $A, B \in \mathbb{R}^{n \times r}$, $A \otimes B = \begin{bmatrix} \cdots & \Bigg| \begin{matrix} \uparrow \\ \mathrm{vec}(a_i \otimes b_j) \\ \downarrow \end{matrix} \Bigg| & \cdots \end{bmatrix} \in \mathbb{R}^{n^2 \times r^2}$.

Very friendly matrix product with many nice properties!

Basis doesn't matter for the (span of the) Kronecker product!

Useful fact: For any $A, B \in \mathbb{R}^{n \times k}$,

$$\mathrm{colspan}(A \odot B) \subseteq \mathrm{colspan}(A \otimes B)$$

because Khatri-Rao product columns are subset of Kronecker product columns

Khatri-Rao Product:

$$\begin{array}{cc} & \overset{r \text{ columns}}{\longleftrightarrow} \\ A \odot B = & \begin{bmatrix} \uparrow \\ \ldots \mathrm{vec}(a_i \otimes b_i) \ldots \\ \downarrow \end{bmatrix} \end{array}$$

Connection to Overparameterization

Khatri-Rao Product: r columns

$$A \odot B = \begin{bmatrix} & \uparrow & \\ \dots & \mathrm{vec}(a_i \otimes b_i) & \dots \\ & \downarrow & \end{bmatrix}$$

Kronecker Product: r^2 columns

$$A \otimes B = \left[\begin{array}{c|c|c} & \uparrow & \\ \dots & \mathrm{vec}(a_i \otimes b_j) & \dots \\ & \downarrow & \end{array}\right]$$

Insight: Maybe overparameterization helps because it allows the model to learn the bigger Kronecker space, which is easier to capture (basis independent)

Old goal: For $X^{(1)}, Y^{(1)}$ with r columns, show that

$$\mathrm{colspan}(X^{(1)} \odot Y^{(1)}) = \mathrm{colspan}(A \odot A)$$

Issue: Would need columns of $X^{(1)}$ to match columns of A, which is unlikely.

Updated goal: For $X^{(1)}, Y^{(1)}$ now with r^2 columns, show that

$$\begin{aligned} \mathrm{colspan}(X^{(1)} \odot Y^{(1)}) &= \mathrm{colspan}(A \otimes A) \\ &\supseteq \mathrm{colspan}(A \odot A)\,. \end{aligned}$$

Khatri-Rao Product:

$$\begin{array}{c} \overset{r \text{ columns}}{\longleftrightarrow} \\ A \odot B = \begin{bmatrix} & \uparrow & \\ \ldots & \mathrm{vec}(a_i \otimes b_i) & \ldots \\ & \downarrow & \end{bmatrix} \end{array}$$

Proof Overview

Kronecker Product:

$$\begin{array}{c} \overset{r^2 \text{ columns}}{\longleftrightarrow} \\ A \otimes B = \left[\begin{array}{c|c|c} & \uparrow & \\ \ldots & \mathrm{vec}(a_i \otimes b_j) & \ldots \\ & \downarrow & \end{array}\right] \end{array}$$

Initialize: $X^{(0)}, Y^{(0)}, Z^{(0)} \sim \mathcal{N}(0,1)^{n \times r^2}$

Step 1a: $X^{(1)} \leftarrow A(A \odot A)^\top (Y^{(0)} \odot Z^{(0)})^{\top\dagger}$ **Step 1b:** $Y^{(1)} \leftarrow A(A \odot A)^\top (\underline{X^{(0)}} \odot Z^{(0)})^{\top\dagger}$

Step 2c: $Z^{(2)} \leftarrow A(A \odot A)^\top (X^{(1)} \odot Y^{(1)})^{\top\dagger}$

("Parallel" ALS updates for simplicity.)

(1) Use tools from random matrix theory to approximate

$$\left(Y^{(0)} \odot Z^{(0)}\right)^{\top\dagger} \approx \left(Y^{(0)} \odot Z^{(0)}\right)$$

(2) By (1) can, treat $\left(X^{(1)} \odot Y^{(1)}\right)$ as **polynomial** of Gaussian entries.

- **Matrix anticoncentration** via Carbery-Wright inequality tells us that columns span r^2-dimensional space w.h.p.
- Thus $\mathrm{colspan}\left(X^{(1)} \odot Y^{(1)}\right) = \mathrm{colspan}(A \otimes A)$, and step 2c **converges!**

Result

Informal Theorem [Arvanitakis **S.** Vijayaraghavan '25]

Given a rank-r tensor $T \in \mathbb{R}^{n\times n\times n}$, with unknown factorization $T = \sum_{i=1}^{r} a_i \otimes b_i \otimes c_i$, with mildly conditioned the factor matrices A, B, C, a parallel variant of Alternating Least Squares (ALS) with $k = \Omega(r^2)$ factors and random initialization converges to a global minimum X, Y, Z, such that

$$\left\| T - \sum_{i=1}^{k} x_i \otimes y_i \otimes z_i \right\|_F^2 = 0,$$

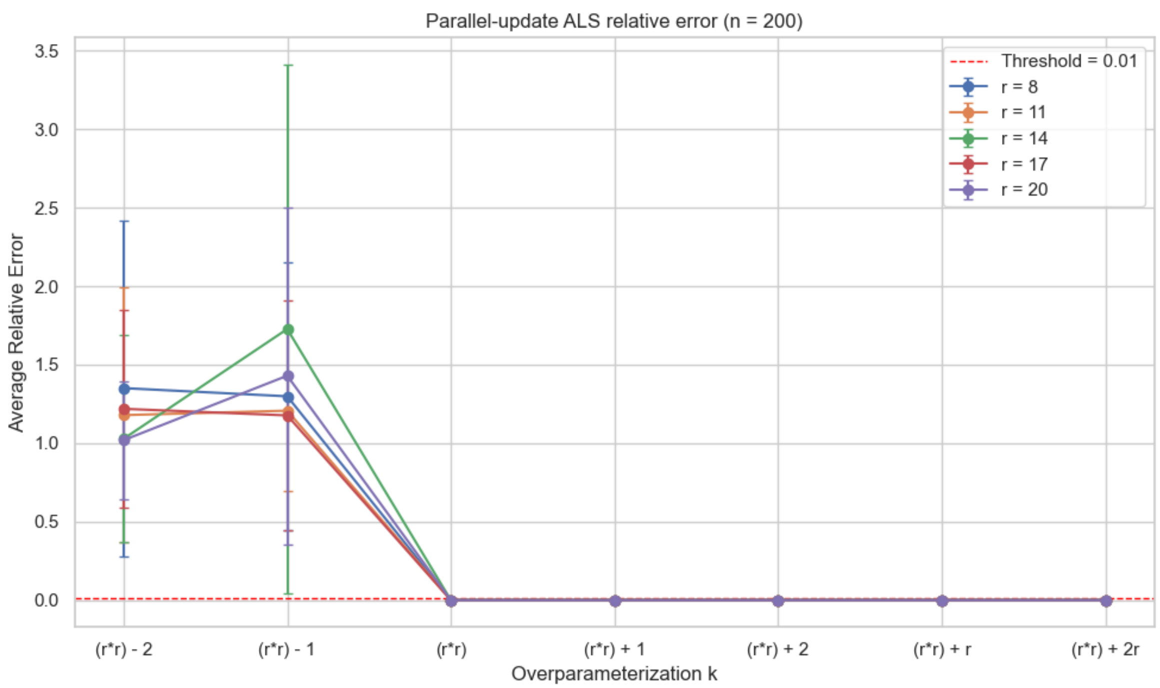
with high probability.

Can be extended to more general **low-rank approximation** problem

- Recover a tensor of rank $O(r^2)$ with least-squares objective competitive with best tensor of rank at most r

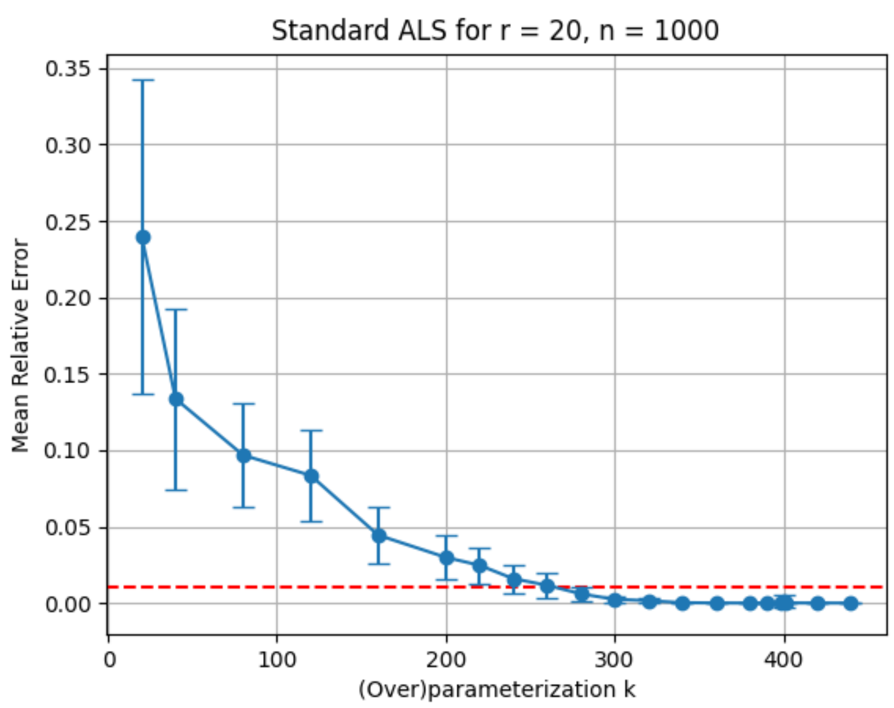
Is overparameterization necessary?

Simulations decomposing randomly generated tensors



Parallel-update variant that we analyze:

- Converges in 1 iteration for $k \geq r^2$
- Not converge for $k < r^2$



Standard sequential-update ALS:

- Smoother tradeoff in overparameterized rank k
- Definitely requires rank $k > r$

Future Directions

- Guarantees for standard sequential update ALS, and higher-order tensors
- Can we use these ideas to analyze gradient descent?
  - Seem to have unlocked some interesting structural properties of random initialization, that might make it easier to reason about
  - Interesting interaction between projecting to the span of true factors, and wanting to preserve randomness in the span of the true factors

THANKS!

vaidehi@u.northwestern.edu