CS-GY 6763: Lecture 10
Linear Programming, Singular Value Decomposition

NYU, Prof. Ainesh Bakshi

Dimension Dependent Convex Optimization

Consider a convex function $f(\mathbf{x})$ be bounded between $[-B, B]$ on a constraint set $\mathcal{S}$.

Theorem (Dimension Dependent Convex Optimization)

The Center-of-Gravity Method finds $\hat{\mathbf{x}}$ *satisfying* $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x}) + \epsilon$ *using* $O(d \log(B/\epsilon))$ *calls to a function and gradient oracle for convex* f*.*

Dimension Dependent Convex Optimization

Consider a convex function $f(\mathbf{x})$ be bounded between $[-B, B]$ on a constraint set $\mathcal{S}$.

Theorem (Dimension Dependent Convex Optimization)

The Center-of-Gravity Method finds $\hat{\mathbf{x}}$ *satisfying* $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x}) + \epsilon$ *using* $O(d \log(B/\epsilon))$ *calls to a function and gradient oracle for convex* f.

The center-of-gravity method is not computationally efficient, but inspired the polynomial time ellipsoid method.

Killer Application: Linear Programming

Linear programs (LPs) are one of the most basic convex constrained, convex optimization problems:

Let $\mathbf{c} \in \mathbb{R}^d, \mathbf{b} \in \mathbb{R}^n, \mathbf{A} \in \mathbb{R}^{n \times d}$ be fixed vectors that define the problem, and let $\mathbf{x}$ be our variable parameter.

$$\min f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}$$

$$\text{subject to } \mathbf{A}\mathbf{x} \geq \mathbf{b}.$$

Think about $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ as a union of half-space constraints:

$$\{\mathbf{x} : \mathbf{a}_1^T \mathbf{x} \geq b_1\}$$

$$\{\mathbf{x} : \mathbf{a}_2^T \mathbf{x} \geq b_2\}$$

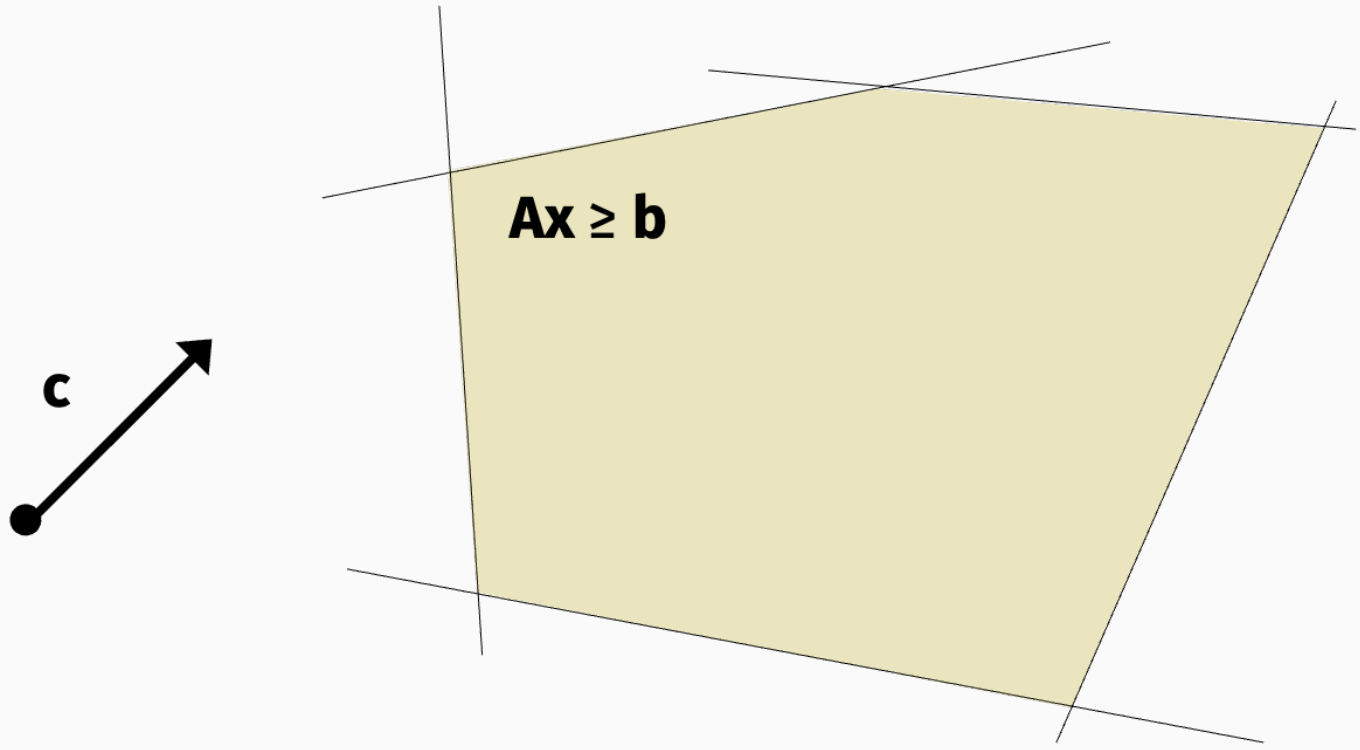
$$\vdots$$

$$\{\mathbf{x} : \mathbf{a}_n^T \mathbf{x} \geq b_n\}$$

Killer Application: Linear Programming

$$\min f(\mathbf{x}) = \mathbf{c}^T\mathbf{x}$$

$$\text{subject to } \mathbf{Ax} \geq \mathbf{b}.$$



Linear Programming Applications

- Classic optimization applications: industrial resource optimization problems were important original applications in the 70s.

Linear Programming Applications

- Classic optimization applications: industrial resource optimization problems were important original applications in the 70s.
- Robust regression: $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$.

Linear Programming Applications

- Classic optimization applications: industrial resource optimization problems were important original applications in the 70s.
- Robust regression: $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$.
- $L1$ constrained regression: $\min_{\mathbf{x}} \|\mathbf{x}\|_1$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$. Lots of applications in sparse recovery/compressed sensing.

Linear Programming Applications

- Classic optimization applications: industrial resource optimization problems were important original applications in the 70s.
- Robust regression: $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$.
- $L1$ constrained regression: $\min_{\mathbf{x}} \|\mathbf{x}\|_1$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$. Lots of applications in sparse recovery/compressed sensing.
- Solve $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_\infty$.

Linear Programming Applications

- Classic optimization applications: industrial resource optimization problems were important original applications in the 70s.
- Robust regression: $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$.
- $L1$ constrained regression: $\min_{\mathbf{x}} \|\mathbf{x}\|_1$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$. Lots of applications in sparse recovery/compressed sensing.
- Solve $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_\infty$.
- Polynomial time algorithms for Markov Decision Processes (reinforcement learning).

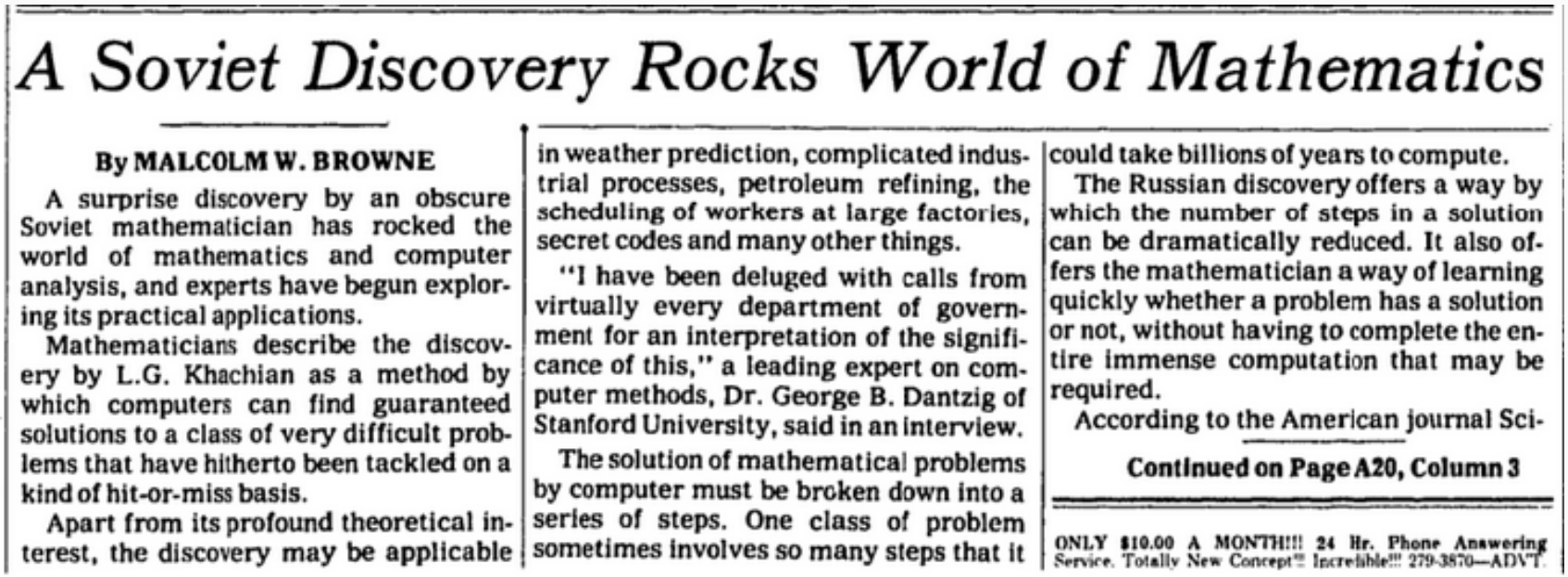
Linear Programming Applications

- Classic optimization applications: industrial resource optimization problems were important original applications in the 70s.
- Robust regression: $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$.
- $L1$ constrained regression: $\min_{\mathbf{x}} \|\mathbf{x}\|_1$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$. Lots of applications in sparse recovery/compressed sensing.
- Solve $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_\infty$.
- Polynomial time algorithms for Markov Decision Processes (reinforcement learning).
- **Many combinatorial optimization problems can be solved via LP relaxation.**

Linear Programming

Theorem (Khachiyan, 1979)

Assume $n = d$. The ellipsoid method solves any linear program with L-bit integer valued constraints exactly in $O(n^4 L)$ time.



A Soviet Discovery Rocks World of Mathematics

By MALCOLM W. BROWNE

A surprise discovery by an obscure Soviet mathematician has rocked the world of mathematics and computer analysis, and experts have begun exploring its practical applications.

Mathematicians describe the discovery by L.G. Khachian as a method by which computers can find guaranteed solutions to a class of very difficult problems that have hitherto been tackled on a kind of hit-or-miss basis.

Apart from its profound theoretical interest, the discovery may be applicable in weather prediction, complicated industrial processes, petroleum refining, the scheduling of workers at large factories, secret codes and many other things.

"I have been deluged with calls from virtually every department of government for an interpretation of the significance of this," a leading expert on computer methods, Dr. George B. Dantzig of Stanford University, said in an interview.

The solution of mathematical problems by computer must be broken down into a series of steps. One class of problem sometimes involves so many steps that it could take billions of years to compute.

The Russian discovery offers a way by which the number of steps in a solution can be dramatically reduced. It also offers the mathematician a way of learning quickly whether a problem has a solution or not, without having to complete the entire immense computation that may be required.

According to the American journal Sci-

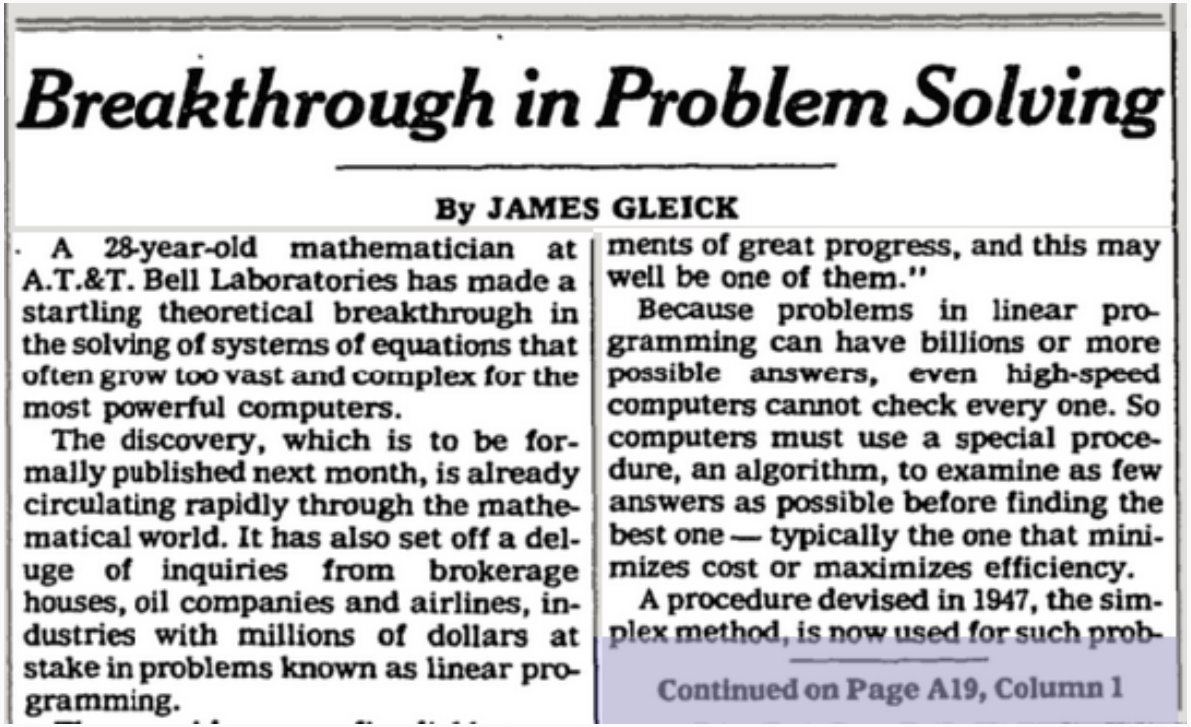
Continued on Page A20, Column 3

Front page of New York Times, November 9, 1979.

Interior Point Methods

Theorem (Karmarkar, 1984)

Assume $n = d$. The interior point method solves any linear program with L-bit integer valued constraints in $O(n^{3.5}L)$ time.



Breakthrough in Problem Solving

By JAMES GLEICK

A 28-year-old mathematician at A.T.&T. Bell Laboratories has made a startling theoretical breakthrough in the solving of systems of equations that often grow too vast and complex for the most powerful computers.

The discovery, which is to be formally published next month, is already circulating rapidly through the mathematical world. It has also set off a deluge of inquiries from brokerage houses, oil companies and airlines, industries with millions of dollars at stake in problems known as linear programming.

ments of great progress, and this may well be one of them."

Because problems in linear programming can have billions or more possible answers, even high-speed computers cannot check every one. So computers must use a special procedure, an algorithm, to examine as few answers as possible before finding the best one — typically the one that minimizes cost or maximizes efficiency.

A procedure devised in 1947, the simplex method, is now used for such prob-

Continued on Page A19, Column 1

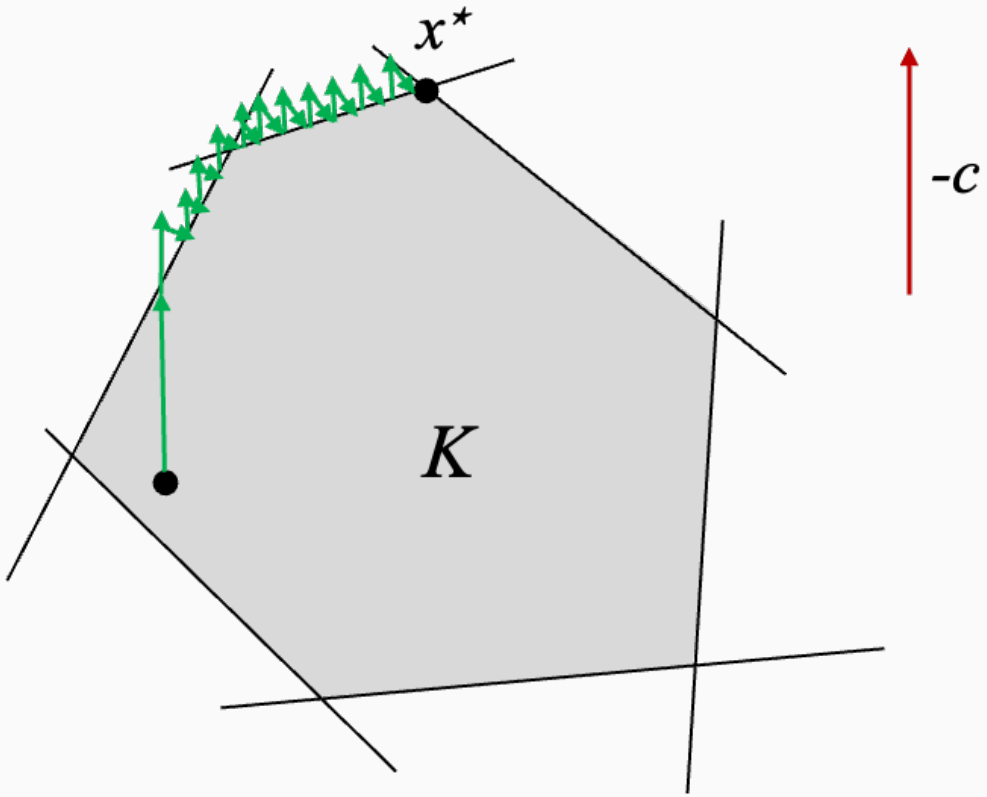
Front page of New York Times, November 19, 1984.

Interior Point Methods

- Lecture notes are posted on the website (optional reading).

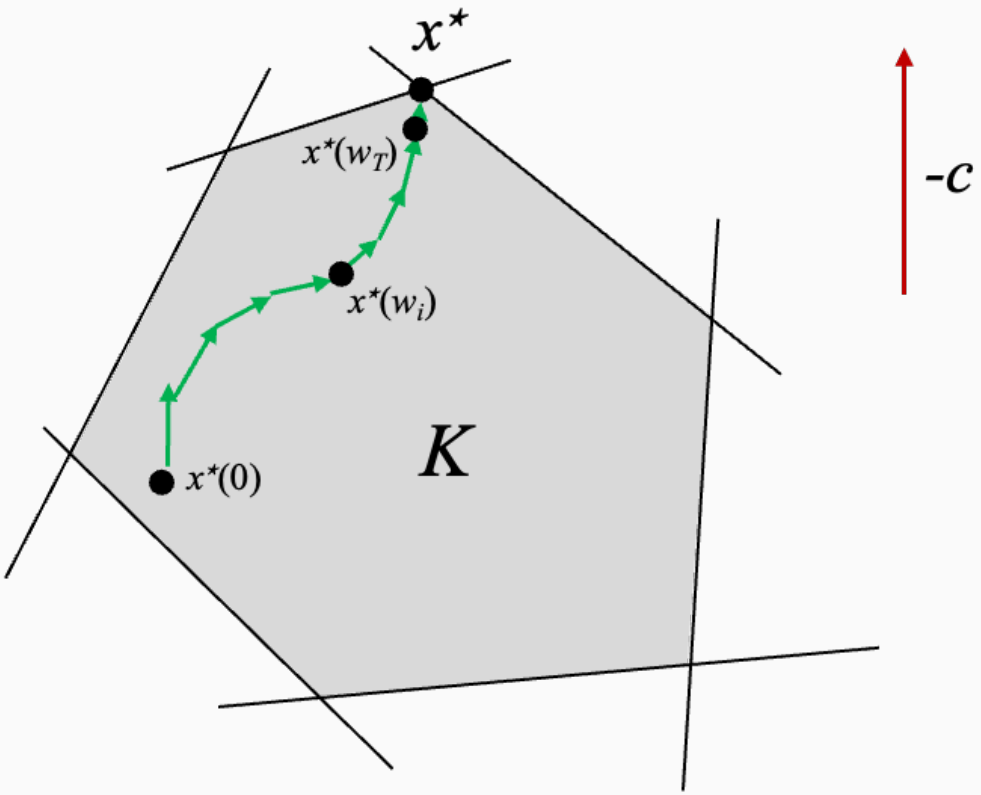
Interior Point Methods

- Lecture notes are posted on the website (optional reading).
- Very similar to projected gradient descent



Projected Gradient Descent Optimization Path

Interior Point Methods



Ideal Interior Point Optimization Path

Polynomial Time Linear Programming

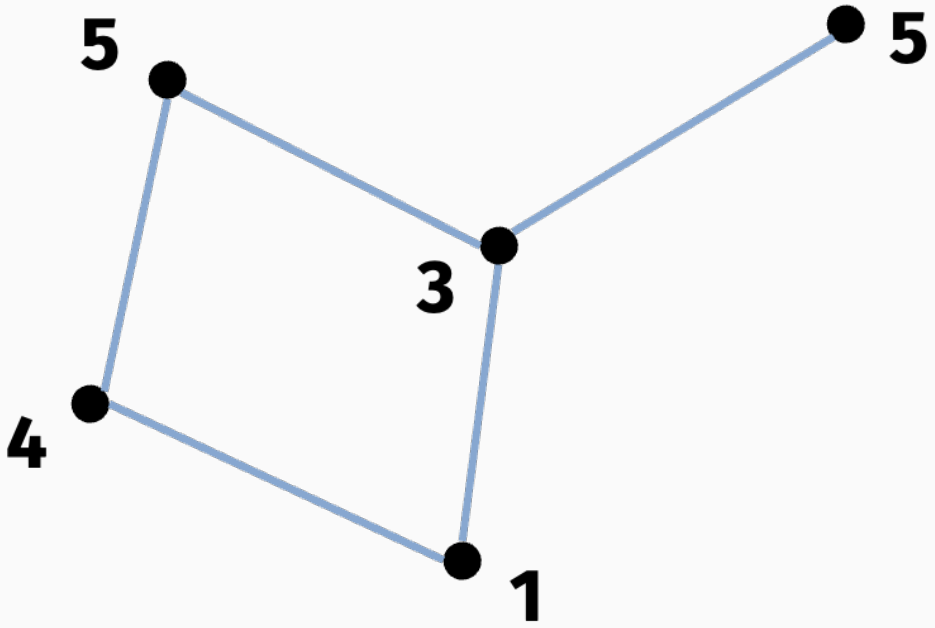
Both results had a huge impact on the theory of optimization, although at the time neither the ellipsoid method nor the interior point method were faster than a heuristic known as the Simplex Method.

These days, improved interior point methods compete with and often outperform simplex.

Polynomial time linear programming algorithms have also had a huge impact on combinatorial optimization. They are often the work-horse behind approximation algorithms for NP-hard problems.

Example: Vertex Cover

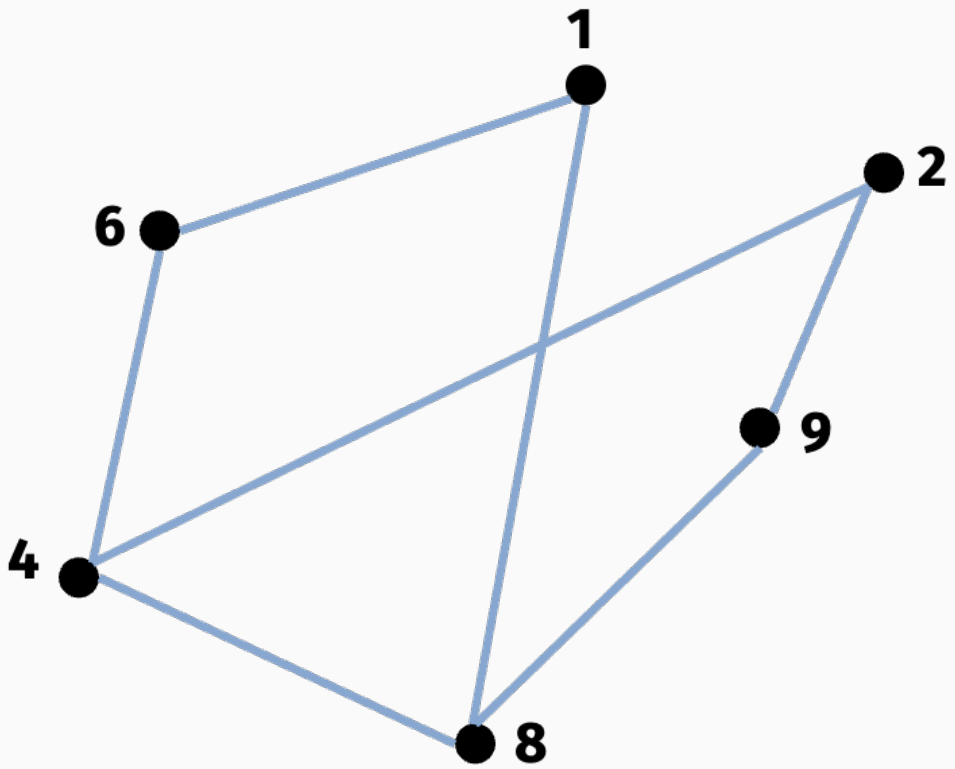
Given a graph G with n nodes and edge set E. Each node is assigned a weight $w_1, \ldots, w_n$.



Goal: Select subset of nodes with minimum total weight that covers all edges.

Example: Vertex Cover

NP-hard to solve exactly.



Example: Vertex Cover

Given a graph G with n nodes and edge set E. Each node is assigned a weight $w_1, \ldots, w_n$.

Formally: Denote if node i is selected by assigning variable x_i to 0 or 1. Let $\mathbf{x} = [x_1, \ldots, x_n]$.

$$\min_{\mathbf{x}} \sum_{i=1}^{n} x_i w_i \qquad \text{subject to} \qquad x_i \in \{0, 1\} \text{ for all } i$$

$$x_i + x_j \geq 1 \text{ for all } (i, j) \in E$$

Example: Vertex Cover

Given a graph G with n nodes and edge set E. Each node is assigned a weight $w_1, \ldots, w_n$.

Formally: Denote if node i is selected by assigning variable x_i to 0 or 1. Let $\mathbf{x} = [x_1, \ldots, x_n]$.

$$\min_{\mathbf{x}} \sum_{i=1}^{n} x_i w_i \qquad \text{subject to} \qquad x_i \in \{0, 1\} \text{ for all } i$$

$$x_i + x_j \geq 1 \text{ for all } (i, j) \in E$$

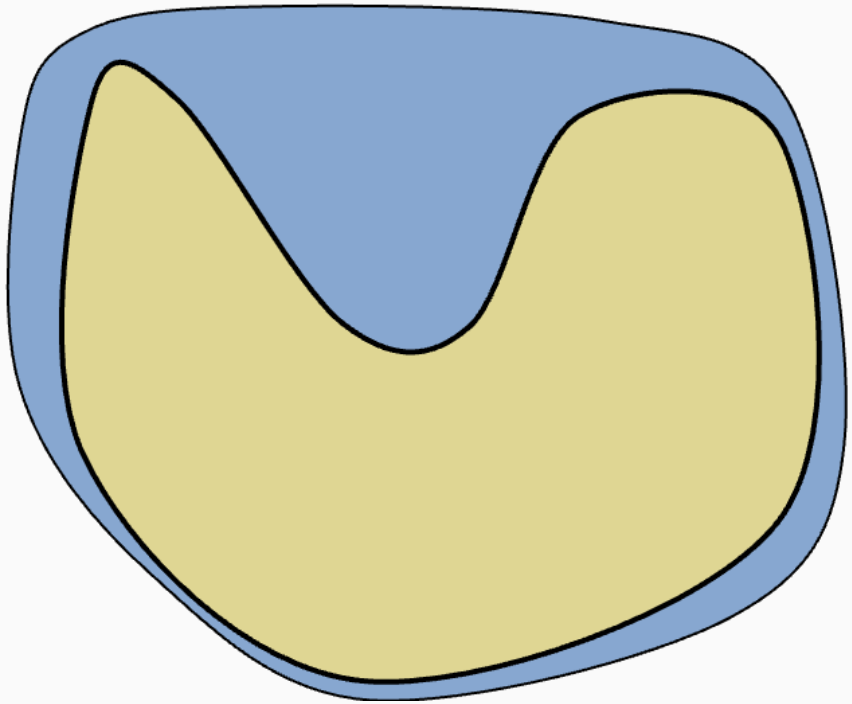
We will use convex optimization to give a 2-approximation in polynomial time.

Function to minimize is linear (so convex) but constraint set is not convex. Why?

Relax-and-Round

High level approach:

- Relax to a problem with convex constraints.
- Round optimal solution of convex problem back to original constraint set.



Relax-and-Round

High level approach:

- Relax to a problem with convex constraints.
- Round optimal solution of convex problem back to original constraint set.

Let $\bar{\mathcal{S}} \supseteq \mathcal{S}$ be the relaxed constraint set. Let $\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x})$ and let $\bar{\mathbf{x}}^* = \arg\min_{\mathbf{x} \in \bar{\mathcal{S}}} f(\mathbf{x})$. We always have that:

$$f(\bar{\mathbf{x}}^*) \leq f(\mathbf{x}^*).$$

Relax-and-Round

High level approach:

- Relax to a problem with convex constraints.
- Round optimal solution of convex problem back to original constraint set.

Let $\bar{\mathcal{S}} \supseteq \mathcal{S}$ be the relaxed constraint set. Let $\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x})$ and let $\bar{\mathbf{x}}^* = \arg\min_{\mathbf{x} \in \bar{\mathcal{S}}} f(\mathbf{x})$. We always have that:

$$f(\bar{\mathbf{x}}^*) \leq f(\mathbf{x}^*).$$

So typically the goal is to round $\bar{\mathbf{x}}^*$ to $\mathcal{S}$ in such a way that we don't increase the function value too much.

Relax-and-Round

High level approach:

- Relax to a problem with convex constraints.
- Round optimal solution of convex problem back to original constraint set.

Let $\bar{\mathcal{S}} \supseteq \mathcal{S}$ be the relaxed constraint set. Let $\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x})$ and let $\bar{\mathbf{x}}^* = \arg\min_{\mathbf{x} \in \bar{\mathcal{S}}} f(\mathbf{x})$. We always have that:

$$f(\bar{\mathbf{x}}^*) \leq f(\mathbf{x}^*).$$

So typically the goal is to round $\bar{\mathbf{x}}^*$ to $\mathcal{S}$ in such a way that we don't increase the function value too much.

$$\text{WTS:} \quad f(\text{round}(\bar{\mathbf{x}}^*)) \leq \alpha \cdot f(\bar{\mathbf{x}}^*) \leq \alpha \cdot f(\mathbf{x}^*)$$

Relaxing Vertex Cover

Vertex Cover:

$$\min_{\mathbf{x}} \sum_{i=1}^{n} x_i w_i \qquad \text{subject to} \qquad x_i \in \{0, 1\} \text{ for all } i$$

$$x_i + x_j \geq 1 \text{ for all } (i, j) \in E$$

Relaxed Vertex Cover:

$$\min_{\mathbf{x}} \sum_{i=1}^{n} x_i w_i \qquad \text{subject to} \qquad 0 \leq x_i \leq 1 \text{ for all } i$$

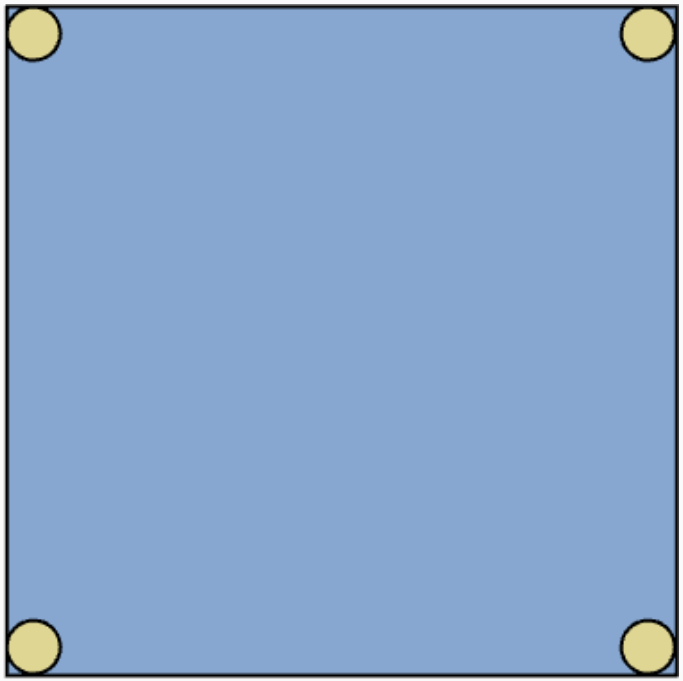
$$x_i + x_j \geq 1 \text{ for all } (i, j) \in E$$

The second problem is a linear program! It can be solved in poly(n) time!

Relax-and-Round

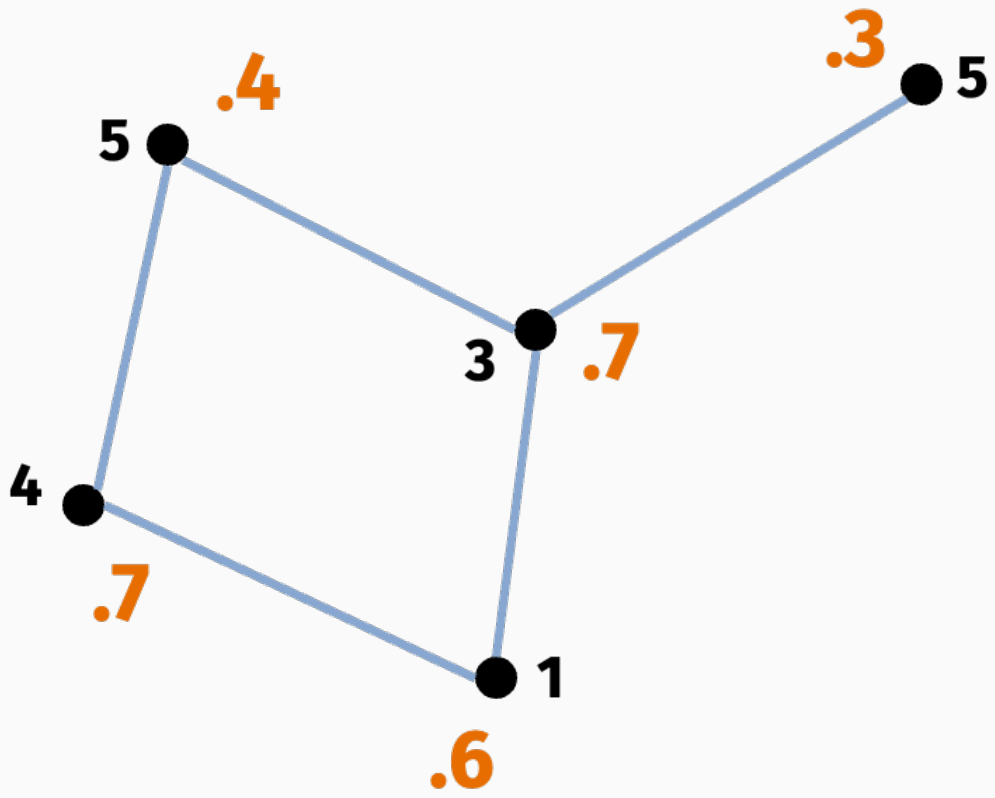
High level approach:

- Relax to a problem with convex constraints.
- Round optimal solution of convex problem back to original constraint set.



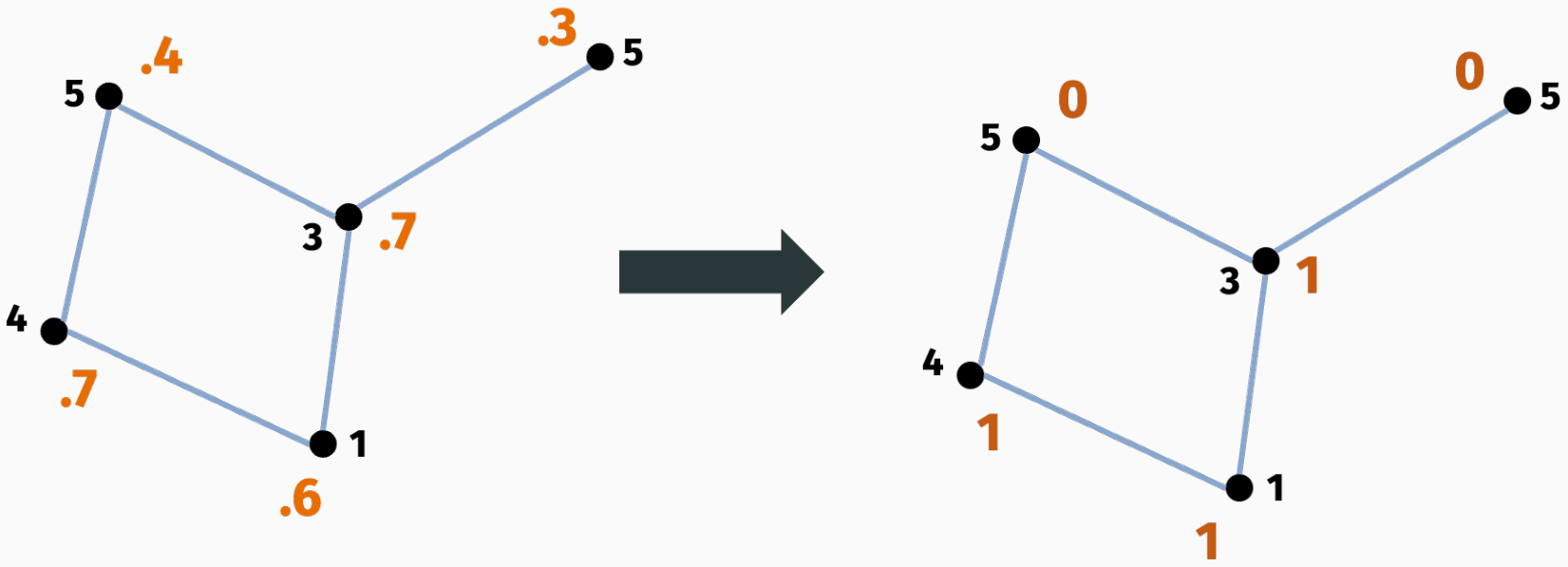
Rounding Vertex Cover

Any ideas on how to round this to a solution to the original problem? I.e., with constraints $x_i \in \{0, 1\}$ for all i.



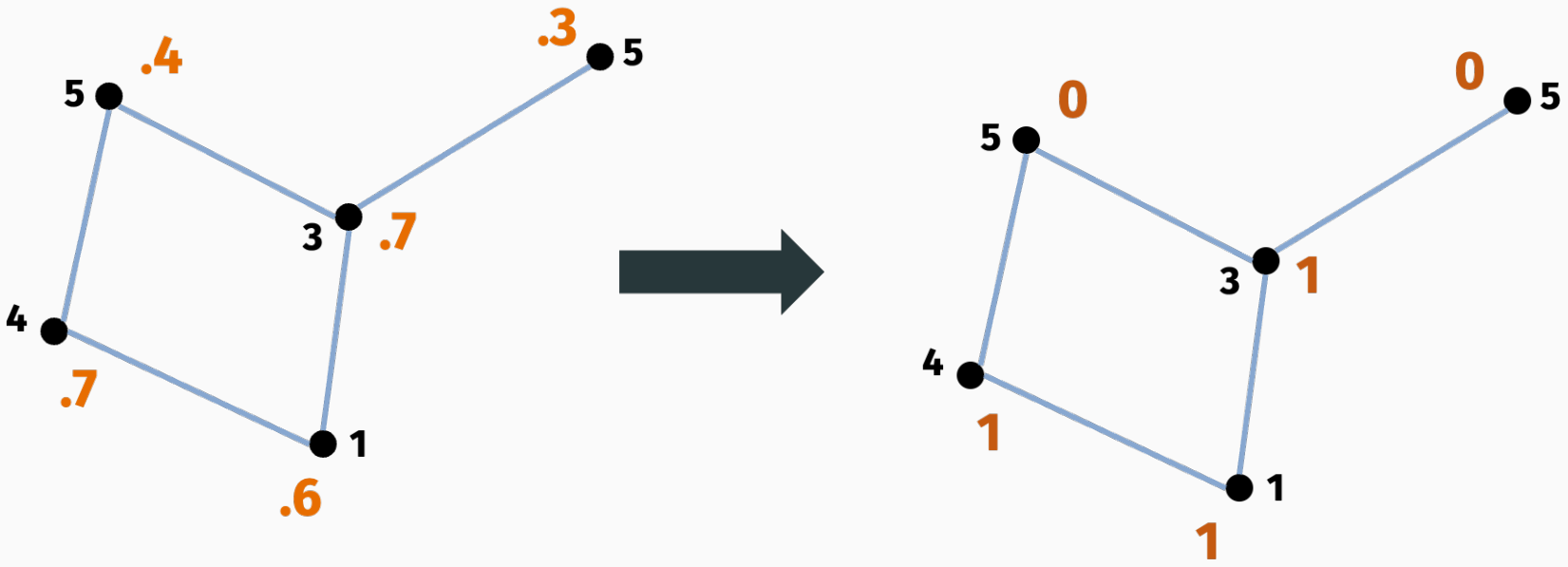
Rounding Vertex Cover

Simply set all variables $x_i = 1$ if $\bar{x}_i^* \geq 1/2$ and $x_i = 0$ otherwise.



Rounding Vertex Cover

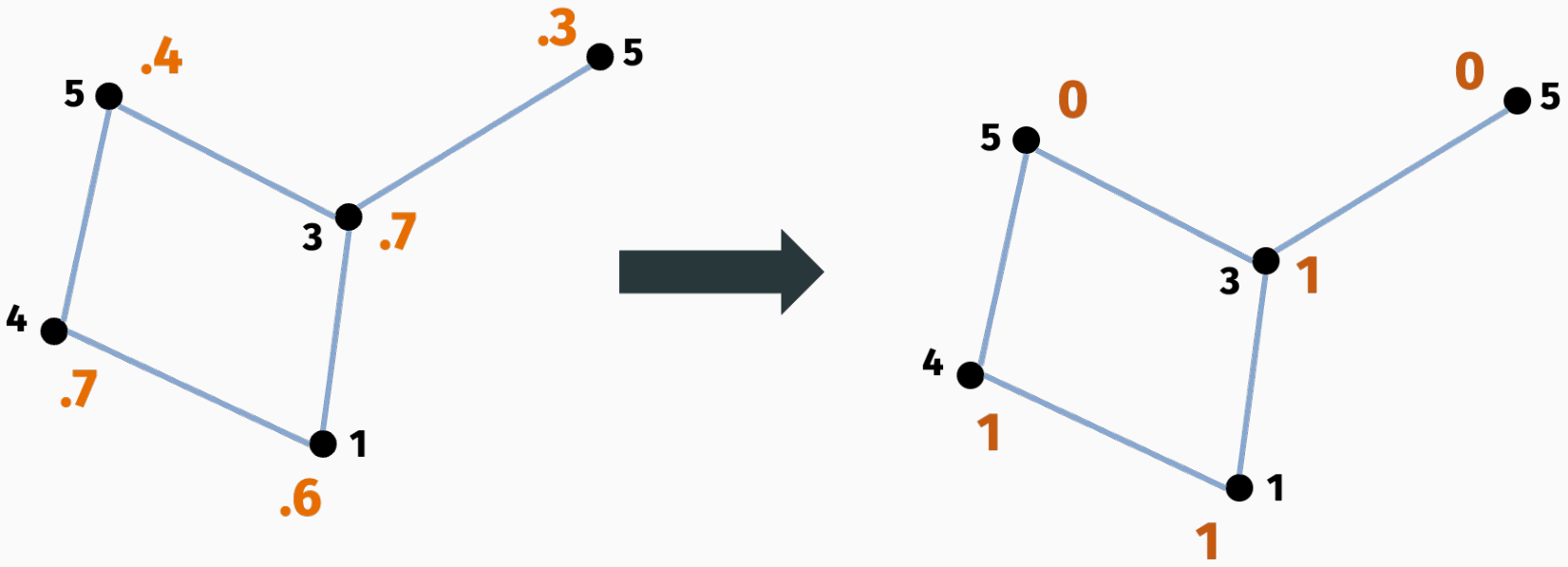
Simply set all variables $x_i = 1$ if $\bar{x}_i^* \geq 1/2$ and $x_i = 0$ otherwise.



Observation 1: All edges remain covered. I.e., the constraint $x_i + x_j \geq 1$ for all $(i, j) \in E$ is not violated.

Rounding Vertex Cover

Simply set all variables $x_i = 1$ if $\bar{x}_i^* \geq 1/2$ and $x_i = 0$ otherwise.



Observation 1: All edges remain covered. I.e., the constraint $x_i + x_j \geq 1$ for all $(i, j) \in E$ is not violated.

Why?

Rounding Vertex Cover

Observation 2: Let $\mathbf{x}$ be the rounded version of $\bar{\mathbf{x}}^*$. We have $f(\mathbf{x}) \leq 2 \cdot f(\bar{\mathbf{x}}^*)$, and thus $f(\mathbf{x}) \leq 2 \cdot f(\mathbf{x}^*)$.

Proof: Observe, for any i, if $\bar{x}_i^* \geq 1/2$ then $x_i = 1$. So, $\text{round}(\bar{x}_i) \leq 2\bar{x}_i^*$. Thus,

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{n} \text{round}(\bar{x}_i) w_i$$

Rounding Vertex Cover

Observation 2: Let $\mathbf{x}$ be the rounded version of $\bar{\mathbf{x}}^*$. We have $f(\mathbf{x}) \leq 2 \cdot f(\bar{\mathbf{x}}^*)$, and thus $f(\mathbf{x}) \leq 2 \cdot f(\mathbf{x}^*)$.

Proof: Observe, for any i, if $\bar{x}_i^* \geq 1/2$ then $x_i = 1$. So, $\text{round}(\bar{x}_i) \leq 2\bar{x}_i^*$. Thus,

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{n} \text{round}(\bar{x}_i) w_i \\ &\leq \sum_{i=1}^{n} 2 \cdot \bar{x}_i^* w_i = 2 \cdot f(\bar{\mathbf{x}}^*) \end{aligned}$$

Rounding Vertex Cover

Observation 2: Let $\mathbf{x}$ be the rounded version of $\bar{\mathbf{x}}^*$. We have $f(\mathbf{x}) \leq 2 \cdot f(\bar{\mathbf{x}}^*)$, and thus $f(\mathbf{x}) \leq 2 \cdot f(\mathbf{x}^*)$.

Proof: Observe, for any i, if $\bar{x}_i^* \geq 1/2$ then $x_i = 1$. So, $\text{round}(\bar{x}_i) \leq 2\bar{x}_i^*$. Thus,

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{n} \text{round}(\bar{x}_i) w_i \\ &\leq \sum_{i=1}^{n} 2 \cdot \bar{x}_i^* w_i = 2 \cdot f(\bar{\mathbf{x}}^*) \end{aligned}$$

Rounding Vertex Cover

Observation 2: Let $\mathbf{x}$ be the rounded version of $\bar{\mathbf{x}}^*$. We have $f(\mathbf{x}) \leq 2 \cdot f(\bar{\mathbf{x}}^*)$, and thus $f(\mathbf{x}) \leq 2 \cdot f(\mathbf{x}^*)$.

Proof: Observe, for any i, if $\bar{x}_i^* \geq 1/2$ then $x_i = 1$. So, $\text{round}(\bar{x}_i) \leq 2\bar{x}_i^*$. Thus,

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{n} \text{round}(\bar{x}_i) w_i \\ &\leq \sum_{i=1}^{n} 2 \cdot \bar{x}_i^* w_i = 2 \cdot f(\bar{\mathbf{x}}^*) \end{aligned}$$

Recall, $f(\bar{\mathbf{x}}^*) \leq f(\mathbf{x}^*)$, so we have $f(\mathbf{x}) \leq 2 \cdot f(\mathbf{x}^*)$.

Vertex Cover

So, a polynomial time algorithm for solving LPs immediately yields a 2-approximation algorithm for the NP-hard problem of vertex cover.

- Proven that it is NP-hard to do better than a 1.36 approximation in [Dinur, Safra, 2002].

Vertex Cover

So, a polynomial time algorithm for solving LPs immediately yields a 2-approximation algorithm for the NP-hard problem of vertex cover.

- Proven that it is NP-hard to do better than a 1.36 approximation in [Dinur, Safra, 2002].
- Recently improved to $\sqrt{2} \approx 1.41$ in [Khot, Minzer, Safra 2018], which proved the 2-to-2 games conjecture.

Vertex Cover

So, a polynomial time algorithm for solving LPs immediately yields a 2-approximation algorithm for the NP-hard problem of vertex cover.

- Proven that it is NP-hard to do better than a 1.36 approximation in [Dinur, Safra, 2002].
- Recently improved to $\sqrt{2} \approx 1.41$ in [Khot, Minzer, Safra 2018], which proved the 2-to-2 games conjecture.
- Widely believed that doing better than $2 - \epsilon$ is NP-hard for any $\epsilon > 0$, and this is implied by Subhash Khot's Unique Games Conjecture.

There is a simpler greedy 2-approximation algorithm that doesn't use optimization at all!

Spectral Methods

Next section of course: Spectral methods and numerical linear algebra.

Spectral methods generally refer to methods based on the "spectrum" of a matrix. I.e. on its eigenvectors/eigenvalues and singular vectors/singular values. We will look at

- Applications to low-rank approximation and dimensionality reduction.
- Applications to graph problems.
- Fast algorithms for computing spectral information.

Spectral Methods

Reminder: A vector $\mathbf{v} \in \mathbb{R}^d$ is an eigenvector of a matrix $\mathbf{X} \in \mathbb{R}^{d \times d}$, if there exists a scalar λ such that

$$\mathbf{X}\mathbf{v} = \lambda \mathbf{v}$$

The scalar λ is called the eigenvalue associated with $\mathbf{v}$.

Spectral Methods

Reminder: A vector $\mathbf{v} \in \mathbb{R}^d$ is an eigenvector of a matrix $\mathbf{X} \in \mathbb{R}^{d \times d}$, if there exists a scalar λ such that

$$\mathbf{X}\mathbf{v} = \lambda \mathbf{v}$$

The scalar λ is called the eigenvalue associated with $\mathbf{v}$.

Matrices can often be written completely in terms of their eigenvectors and eigenvalues. This is called eigendecomposition.

Spectral Methods

Reminder: A vector $\mathbf{v} \in \mathbb{R}^d$ is an eigenvector of a matrix $\mathbf{X} \in \mathbb{R}^{d \times d}$, if there exists a scalar λ such that

$$\mathbf{X}\mathbf{v} = \lambda \mathbf{v}$$

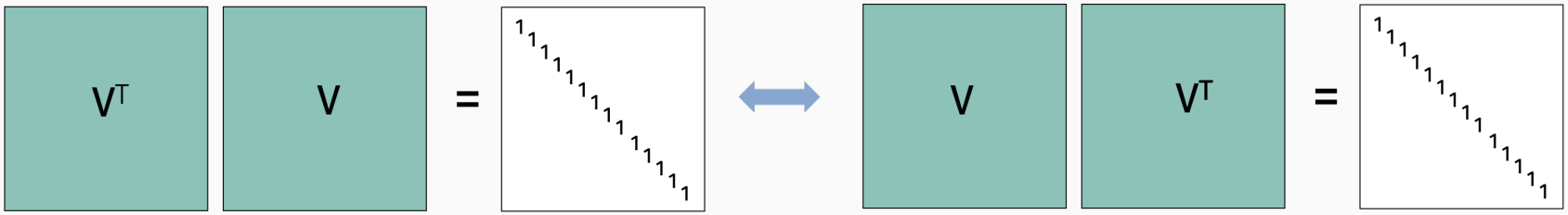
The scalar λ is called the eigenvalue associated with $\mathbf{v}$.

Matrices can often be written completely in terms of their eigenvectors and eigenvalues. This is called eigendecomposition.

We will actually focus on a related tool called singular value decomposition.

Linear Algebra Reminder

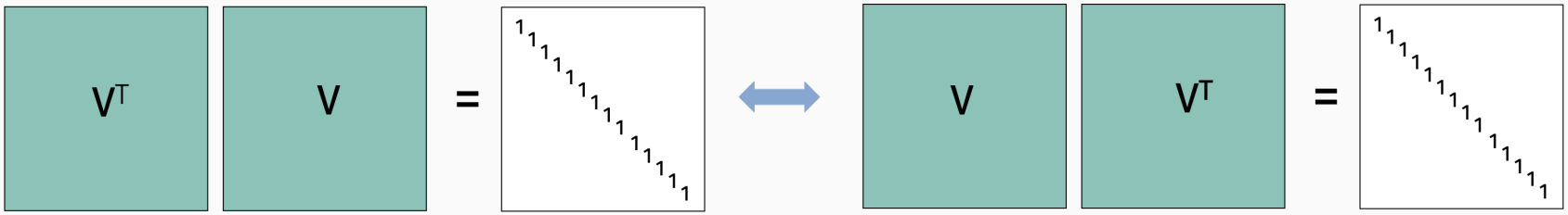
If a square matrix has orthonormal rows, it also has orthonormal columns:



$$\mathbf{V}^T\mathbf{V} = \mathbf{I} = \mathbf{V}\mathbf{V}^T$$

Linear Algebra Reminder

If a square matrix has orthonormal rows, it also has orthonormal columns:



$$\mathbf{V}^T\mathbf{V} = \mathbf{I} = \mathbf{V}\mathbf{V}^T$$

$$\begin{bmatrix} -0.62 & 0.78 & -0.11 \\ -0.28 & -0.35 & -0.89 \\ -0.73 & -0.52 & 0.44 \end{bmatrix} \cdot \begin{bmatrix} -0.62 & -0.28 & -0.73 \\ 0.78 & -0.35 & -0.52 \\ -0.11 & -0.89 & 0.44 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Linear Algebra Reminder

For V with orthonormal columns and any $\mathbf{x}$, $\|\mathbf{V}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$

Linear Algebra Reminder

For V with orthonormal columns and any $\mathbf{x}$, $\|\mathbf{V}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$

Proof:

$$\|\mathbf{V}\mathbf{x}\|_2^2 = (\mathbf{V}\mathbf{x})^T(\mathbf{V}\mathbf{x}) = \mathbf{x}^T\mathbf{V}^T\mathbf{V}\mathbf{x} = \|\mathbf{x}\|_2^2$$

Linear Algebra Reminder

For V with orthonormal columns and any $\mathbf{x}$, $\|\mathbf{V}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$

Proof:

$$\|\mathbf{V}\mathbf{x}\|_2^2 = (\mathbf{V}\mathbf{x})^T(\mathbf{V}\mathbf{x}) = \mathbf{x}^T\mathbf{V}^T\mathbf{V}\mathbf{x} = \|\mathbf{x}\|_2^2$$

Linear Algebra Reminder

For V with orthonormal columns and any $\mathbf{x}$, $\|\mathbf{V}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$

Proof:

$$\|\mathbf{V}\mathbf{x}\|_2^2 = (\mathbf{V}\mathbf{x})^T(\mathbf{V}\mathbf{x}) = \mathbf{x}^T\mathbf{V}^T\mathbf{V}\mathbf{x} = \|\mathbf{x}\|_2^2$$

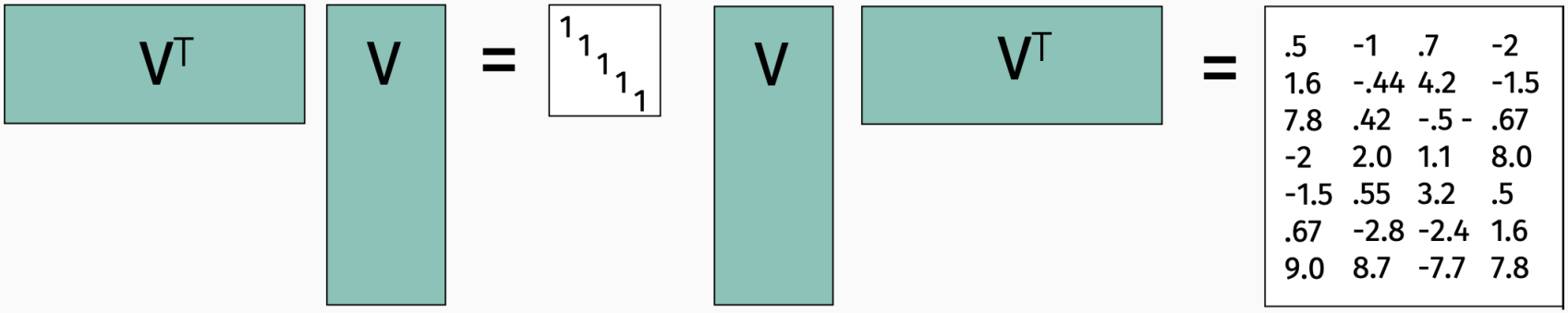
Same thing goes for Frobenius norm: for any matrix $\mathbf{X}$, $\|\mathbf{V}\mathbf{X}\|_F^2 = \|\mathbf{X}\|_F^2$.

Proof: Let $\mathbf{x}_i$ be the i-th row of $\mathbf{X}$. Then,

$$\|\mathbf{V}\mathbf{X}\|_F^2 = \sum_{i=1}^{d} \|\mathbf{V}\mathbf{x}_i\|_2^2 = \sum_{i=1}^{d} \|\mathbf{x}_i\|_2^2 = \|\mathbf{X}\|_F^2$$

Linear Algebra Reminder

The same is not true for rectangular matrices.

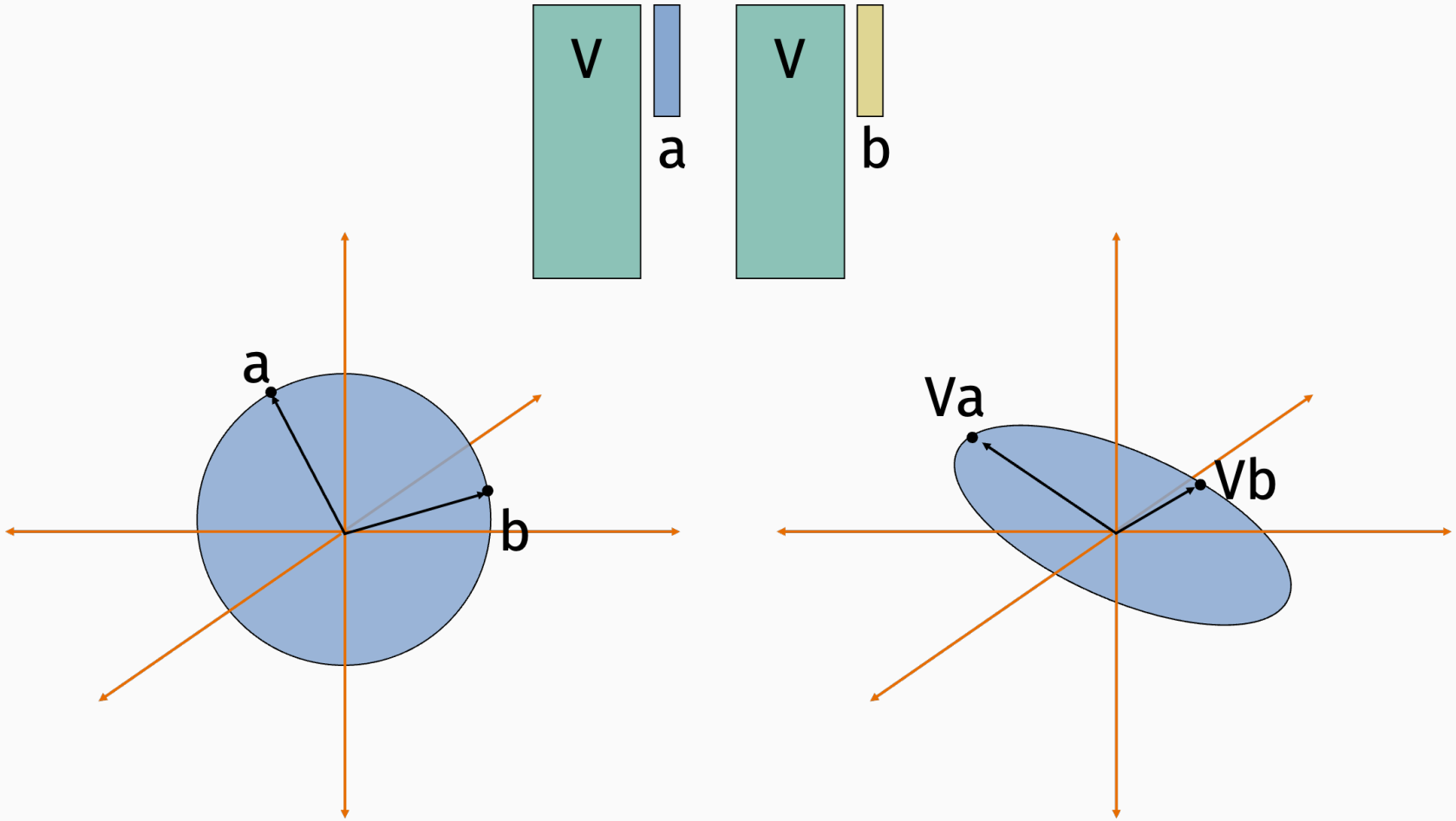


$$\mathbf{V}^T\mathbf{V} = \mathbf{I} \qquad \text{but} \qquad \mathbf{V}\mathbf{V}^T \neq \mathbf{I}$$

For any $\mathbf{x}$, $\|\mathbf{V}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$ but $\|\mathbf{V}^T\mathbf{x}\|_2^2 \neq \|\mathbf{x}\|_2^2$ in general.

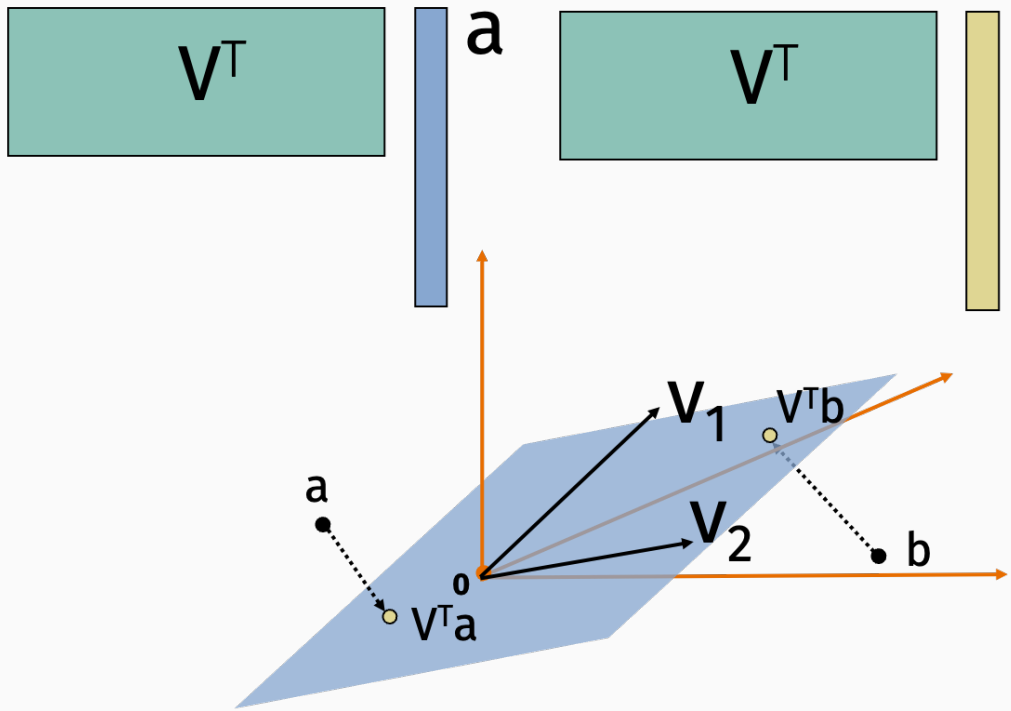
Linear Algebra Reminder

Multiplying a vector by **V** with orthonormal columns rotates and/or reflects the vector.



Linear Algebra Reminder

Multiplying a vector by a rectangular matrix $\mathbf{V}^T$ with orthonormal rows projects the vector (representing it as coordinates in the lower dimensional space).

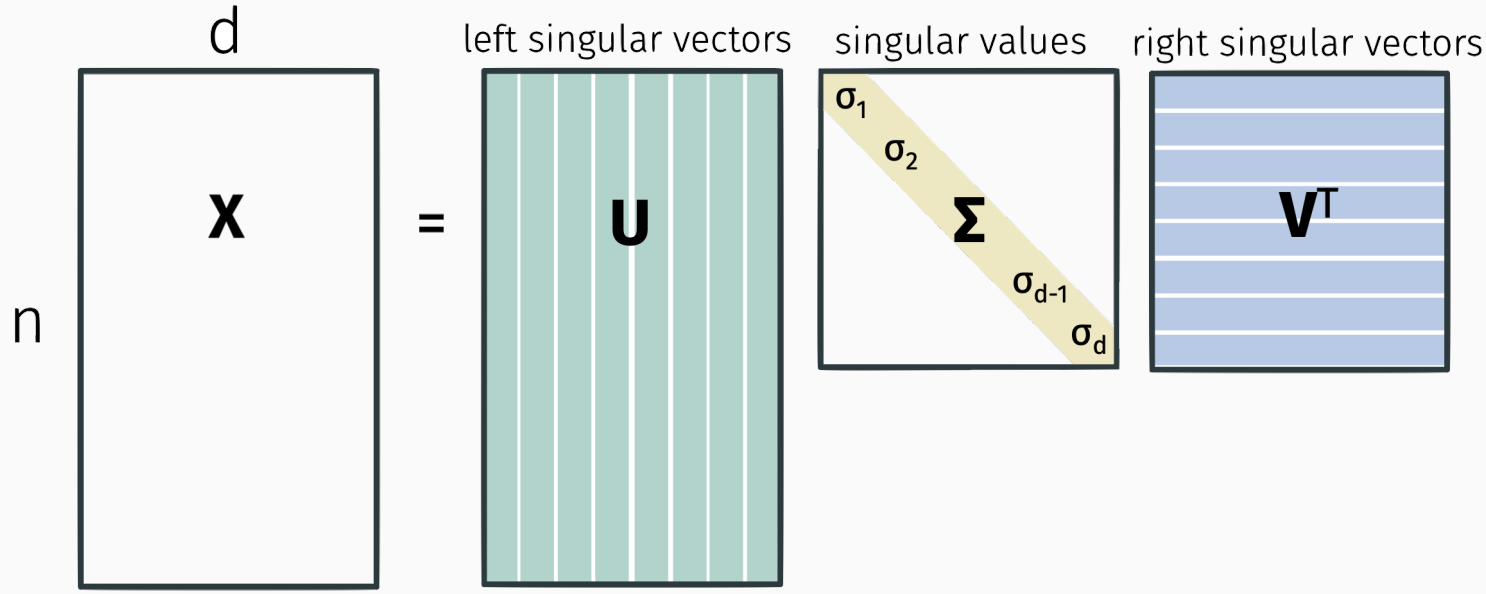


So we always have that $\|\mathbf{V}^T\mathbf{x}\|_2 \leq \|\mathbf{x}\|_2$.

Singular Value Decomposition

One of the most fundamental results in linear algebra.

Any matrix $\mathbf{X}$ can be written:



Where $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, and $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_d \geq 0$.

Singular values are unique. Factors are not. E.g. would still get a valid SVD by multiplying both i^{th} column of $\mathbf{V}$ and $\mathbf{U}$ by -1.

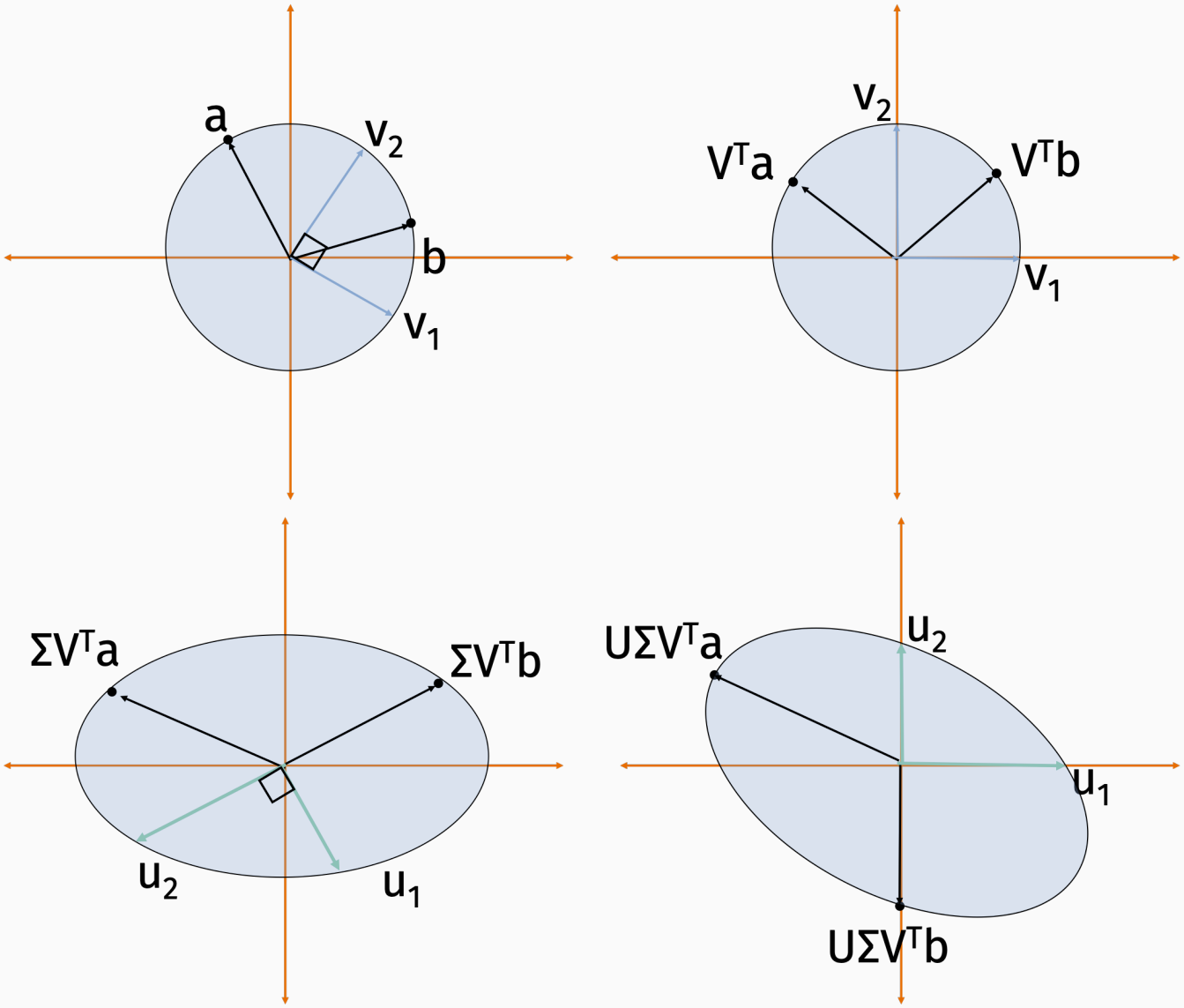
Singular Value Decomposition

Important take away from singular value decomposition.

Multiplying any vector $\mathbf{a}$ by a matrix $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ to form $\mathbf{X}\mathbf{a}$ can be viewed as a composition of 3 operations:

1. Rotate/reflect the vector (multiplication by $\mathbf{V}^T$).
2. Scale the coordinates (multiplication by $\mathbf{\Sigma}$).
3. Rotate/reflect the vector again (multiplication by $\mathbf{U}$).

Singular Value Decomposition: Rotate/Reflect



Comparison to Eigendecomposition

A square matrix has at most d linearly independent eigenvectors. If a matrix has a full set of d eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ with eigenvalues $\lambda_1, \ldots, \lambda_d$ it is called "diagonalizable" and can be written as:

$$\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}.$$

$\mathbf{V}$'s columns are $\mathbf{v}_1, \ldots, \mathbf{v}_d$.

Comparison to Eigendecomposition

Singular Value Decomposition

- Exists for all matrices, square or rectangular.
- Singular values are always positive.
- Factors $\mathbf{U}$ and $\mathbf{V}$ are orthogonal.

Eigendecomposition

- Exists for some square matrices.
- Eigenvalues can be positive, negative, or imaginary. Real if $\mathbf{X}$ is symmetric.
- Factor $\mathbf{V}$ is orthogonal if and only if $\mathbf{X}$ is symmetric.

Connection to Eigendecomposition

- $\mathbf{U}$ contains the orthogonal eigenvectors of $\mathbf{X}\mathbf{X}^T$.
- $\mathbf{V}$ contains the orthogonal eigenvectors of $\mathbf{X}^T\mathbf{X}$.
- $\sigma_i^2 = \lambda_i(\mathbf{X}\mathbf{X}^T) = \lambda_i(\mathbf{X}^T\mathbf{X})$

Connection to Eigendecomposition

- $\mathbf{U}$ contains the orthogonal eigenvectors of $\mathbf{X}\mathbf{X}^T$.
- $\mathbf{V}$ contains the orthogonal eigenvectors of $\mathbf{X}^T\mathbf{X}$.
- $\sigma_i^2 = \lambda_i(\mathbf{X}\mathbf{X}^T) = \lambda_i(\mathbf{X}^T\mathbf{X})$

Proof: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Then,

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T$$

Connection to Eigendecomposition

- $\mathbf{U}$ contains the orthogonal eigenvectors of $\mathbf{X}\mathbf{X}^T$.
- $\mathbf{V}$ contains the orthogonal eigenvectors of $\mathbf{X}^T\mathbf{X}$.
- $\sigma_i^2 = \lambda_i(\mathbf{X}\mathbf{X}^T) = \lambda_i(\mathbf{X}^T\mathbf{X})$

Proof: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Then,

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T$$

Connection to Eigendecomposition

- $\mathbf{U}$ contains the orthogonal eigenvectors of $\mathbf{X}\mathbf{X}^T$.
- $\mathbf{V}$ contains the orthogonal eigenvectors of $\mathbf{X}^T\mathbf{X}$.
- $\sigma_i^2 = \lambda_i(\mathbf{X}\mathbf{X}^T) = \lambda_i(\mathbf{X}^T\mathbf{X})$

Proof: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Then,

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T$$

And

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T$$

SVD Applications

Lots of applications.

- Compute pseudoinverse $\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T$.
- Read off condition number of $\mathbf{X}$, σ_1^2/σ_d^2.
- Compute matrix norms. E.g. $\|\mathbf{X}\|_2 = \sigma_1$, $\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^{d} \sigma_i^2}$.
- Compute matrix square root – i.e. find a matrix $\mathbf{B}$ such that $\mathbf{B}\mathbf{B}^T = \mathbf{X}$. Used e.g. in sampling from Gaussian with covariance $\mathbf{X}$.
- Principal component analysis.

SVD Applications

Lots of applications.

- Compute pseudoinverse $\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T$.
- Read off condition number of $\mathbf{X}$, σ_1^2/σ_d^2.
- Compute matrix norms. E.g. $\|\mathbf{X}\|_2 = \sigma_1$, $\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^{d} \sigma_i^2}$.
- Compute matrix square root – i.e. find a matrix $\mathbf{B}$ such that $\mathbf{B}\mathbf{B}^T = \mathbf{X}$. Used e.g. in sampling from Gaussian with covariance $\mathbf{X}$.
- Principal component analysis.

Killer app: Read off optimal low-rank approximation for $\mathbf{X}$.

Rank

- **Column span** of $\mathbf{X} \in \mathbb{R}^{n \times d}$: set of all vectors $\mathbf{Xa}$ for some $\mathbf{a} \in \mathbb{R}^d$.
- D_c = dimension of column span = max number of linearly independent columns.

Rank

- **Column span** of $\mathbf{X} \in \mathbb{R}^{n \times d}$: set of all vectors $\mathbf{Xa}$ for some $\mathbf{a} \in \mathbb{R}^d$.
- D_c = dimension of column span = max number of linearly independent columns.
- **Row span** of $\mathbf{X} \in \mathbb{R}^{n \times d}$: set of all vectors $\mathbf{b}^T\mathbf{X}$ for some $\mathbf{b} \in \mathbb{R}^n$.
- D_r = dimension of row span = max number of linearly independent rows.

Rank

For a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ we have:

$$D_c \leq d$$
$$D_r \leq n$$
$$D_c = D_r.$$

We call the value of $D_c = D_r$ the rank of $\mathbf{X}$.

Observation: $\text{rank}(\mathbf{X}) \leq \min(n, d)$.

Rank

For a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ we have:

$$D_c \leq d$$
$$D_r \leq n$$
$$D_c = D_r.$$

We call the value of $D_c = D_r$ the rank of $\mathbf{X}$.

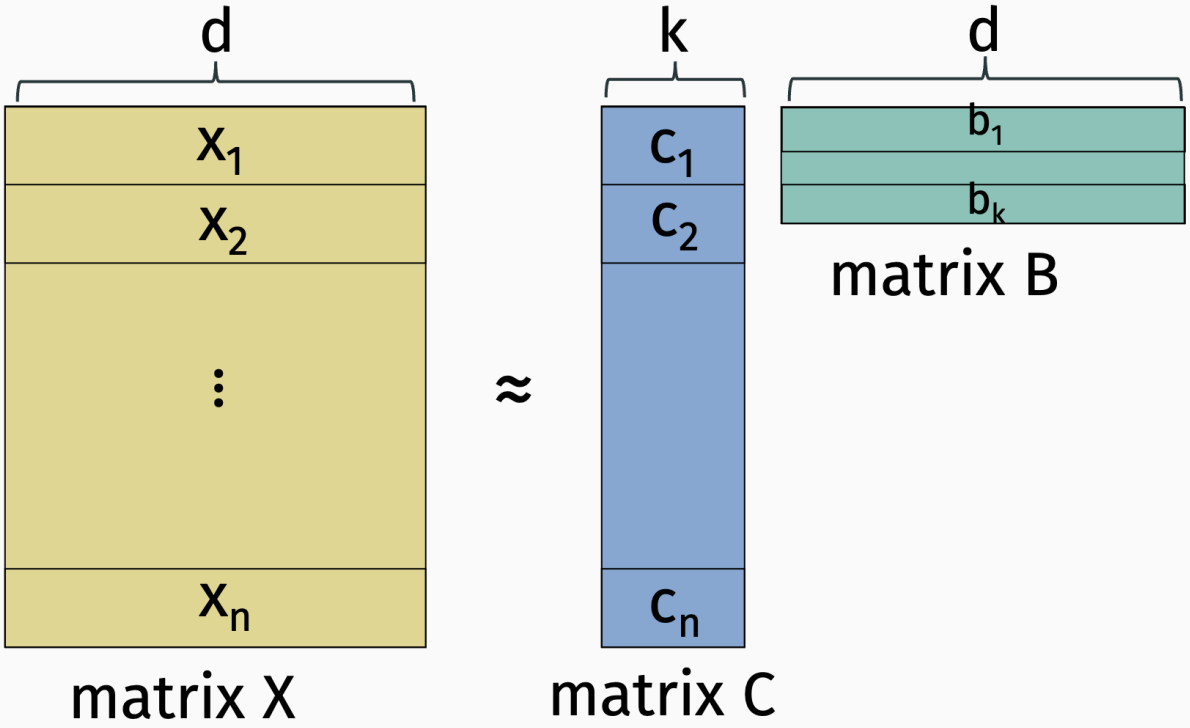
Observation: $\text{rank}(\mathbf{X}) \leq \min(n, d)$.

We always have that:

$$\text{rank}\left(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C} \cdot \ldots\right) \leq \min\left(\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B}), \text{rank}(\mathbf{C}), \ldots\right).$$

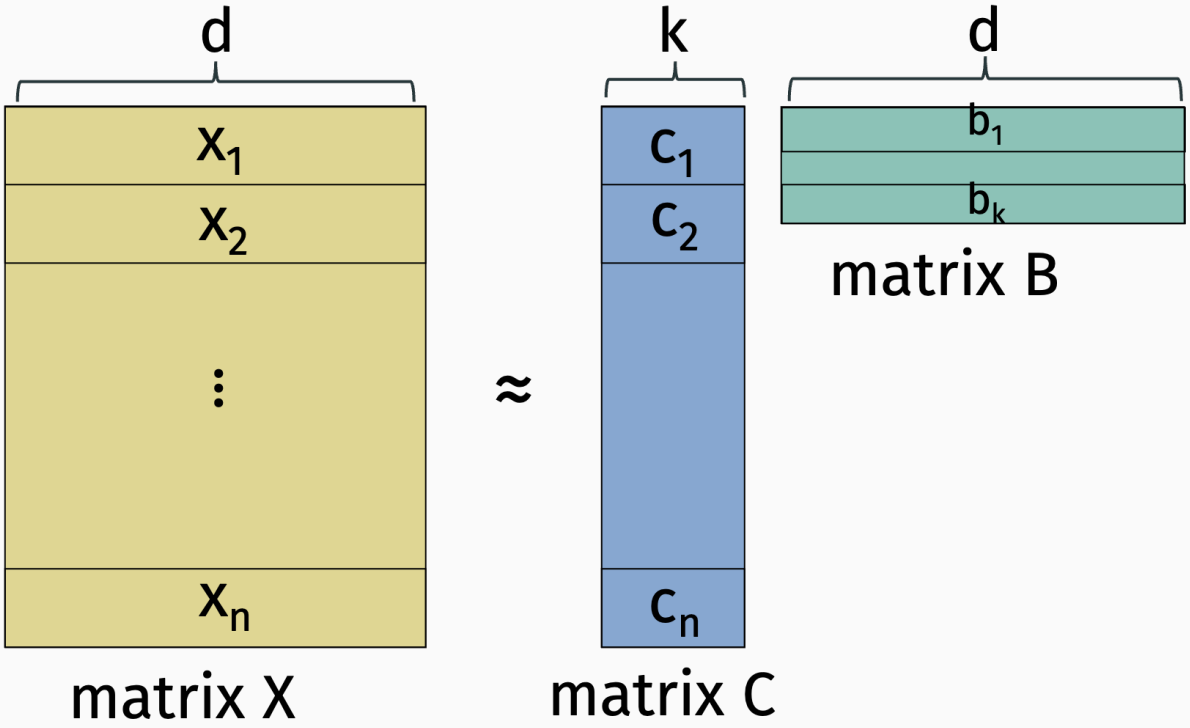
Low-Rank Approximation

Approximate $\mathbf{X}$ as a rank k matrix:



Low-Rank Approximation

Approximate $\mathbf{X}$ as a rank k matrix:

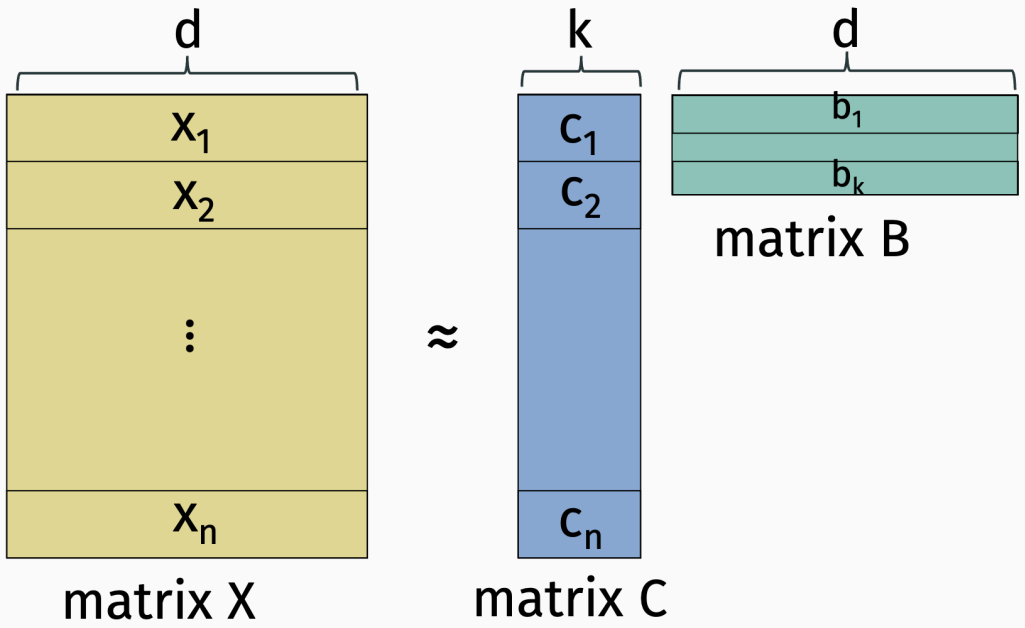


Choose $\mathbf{C}$ and $\mathbf{B}$ to minimize:

$$\min_{\mathbf{B},\mathbf{C}} \|\mathbf{X} - \mathbf{CB}\|$$

for some matrix norm. Common choice is $\|\mathbf{X} - \mathbf{CB}\|_F^2$.

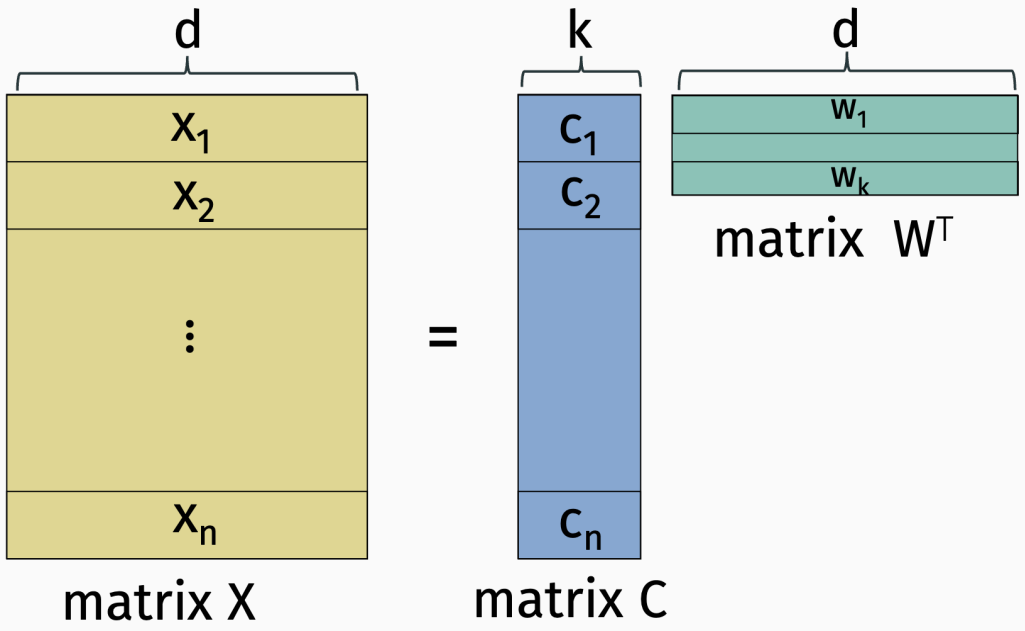
Applications of Low-Rank Approximation



- **CB** takes $O(k(n+d))$ space to store instead of $O(nd)$.
  - Important in many applications, including e.g. LoRA: Low-Rank Adaptation of Large Language Models
- Many linear algebraic problems involving **CB** can be solved in $O(nk^2)$ instead of $O(nd^2)$ time.

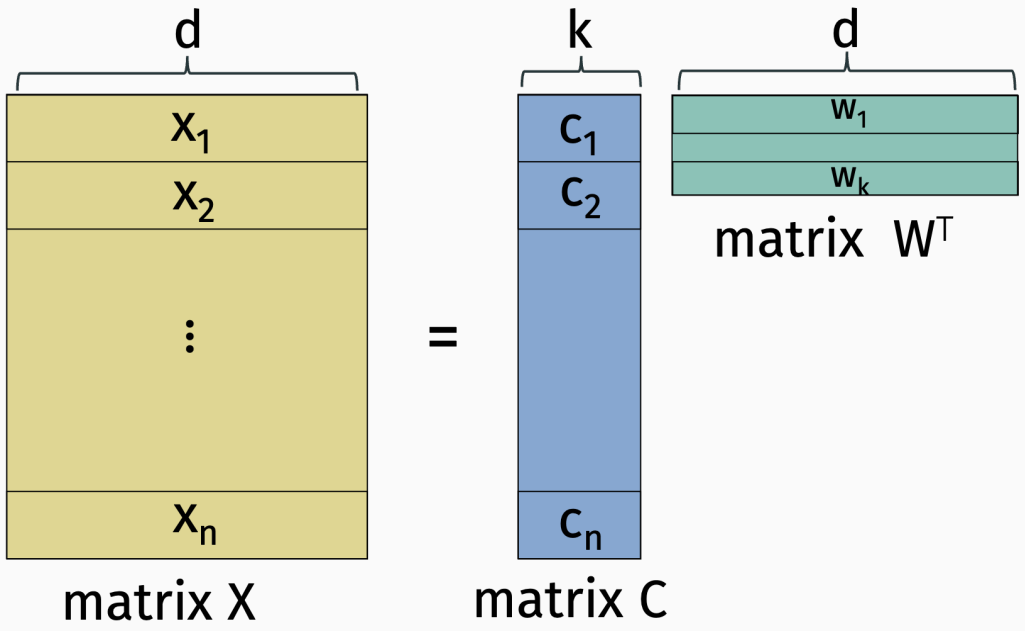
Low-Rank Approximation

Without loss of generality can assume that the right matrix is orthogonal. I.e. $\mathbf{W}^T$ with $\mathbf{W}^T\mathbf{W} = \mathbf{I}$



Low-Rank Approximation

Without loss of generality can assume that the right matrix is orthogonal. I.e. $\mathbf{W}^T$ with $\mathbf{W}^T\mathbf{W} = \mathbf{I}$



If we assume $\mathbf{W}$ is fixed we should choose left matrix $\mathbf{C}$:

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{W}^T\|_F^2$$

This is just n least squares regression problems!

Low-Rank Approximation

Let $\mathbf{x}_i$ be the i-th row of $\mathbf{X}$ and $\mathbf{c}_i$ be the i-th row of $\mathbf{C}$.

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{W}^T\|_F^2 = \min_{\mathbf{C}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{c}_i\mathbf{W}^T\|_2^2$$

Low-Rank Approximation

Let $\mathbf{x}_i$ be the i-th row of $\mathbf{X}$ and $\mathbf{c}_i$ be the i-th row of $\mathbf{C}$.

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{W}^T\|_F^2 = \min_{\mathbf{C}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{c}_i\mathbf{W}^T\|_2^2$$

Observation: The optimal $\mathbf{c}_i$'s can be computed independently:

$$\mathbf{c}_i = \arg\min_{\mathbf{c}} \|\mathbf{x}_i - \mathbf{c}_i\mathbf{W}^T\|_2^2$$

Low-Rank Approximation

Let $\mathbf{x}_i$ be the i-th row of $\mathbf{X}$ and $\mathbf{c}_i$ be the i-th row of $\mathbf{C}$.

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{W}^T\|_F^2 = \min_{\mathbf{C}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{c}_i\mathbf{W}^T\|_2^2$$

Observation: The optimal $\mathbf{c}_i$'s can be computed independently:

$$\mathbf{c}_i = \arg\min_{\mathbf{c}} \|\mathbf{x}_i - \mathbf{c}_i\mathbf{W}^T\|_2^2$$

Using normal equations we have

$$\mathbf{c}_i = \mathbf{x}_i\mathbf{W}\left(\mathbf{W}^T\mathbf{W}\right)^{-1} = \mathbf{x}_i\mathbf{W}$$

and therefore $\mathbf{C} = \mathbf{X}\mathbf{W}$.

Low-Rank Approximation

Let $\mathbf{x}_i$ be the i-th row of $\mathbf{X}$ and $\mathbf{c}_i$ be the i-th row of $\mathbf{C}$.

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{W}^T\|_F^2 = \min_{\mathbf{C}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{c}_i\mathbf{W}^T\|_2^2$$

Observation: The optimal $\mathbf{c}_i$'s can be computed independently:

$$\mathbf{c}_i = \arg\min_{\mathbf{c}} \|\mathbf{x}_i - \mathbf{c}_i\mathbf{W}^T\|_2^2$$

Using normal equations we have

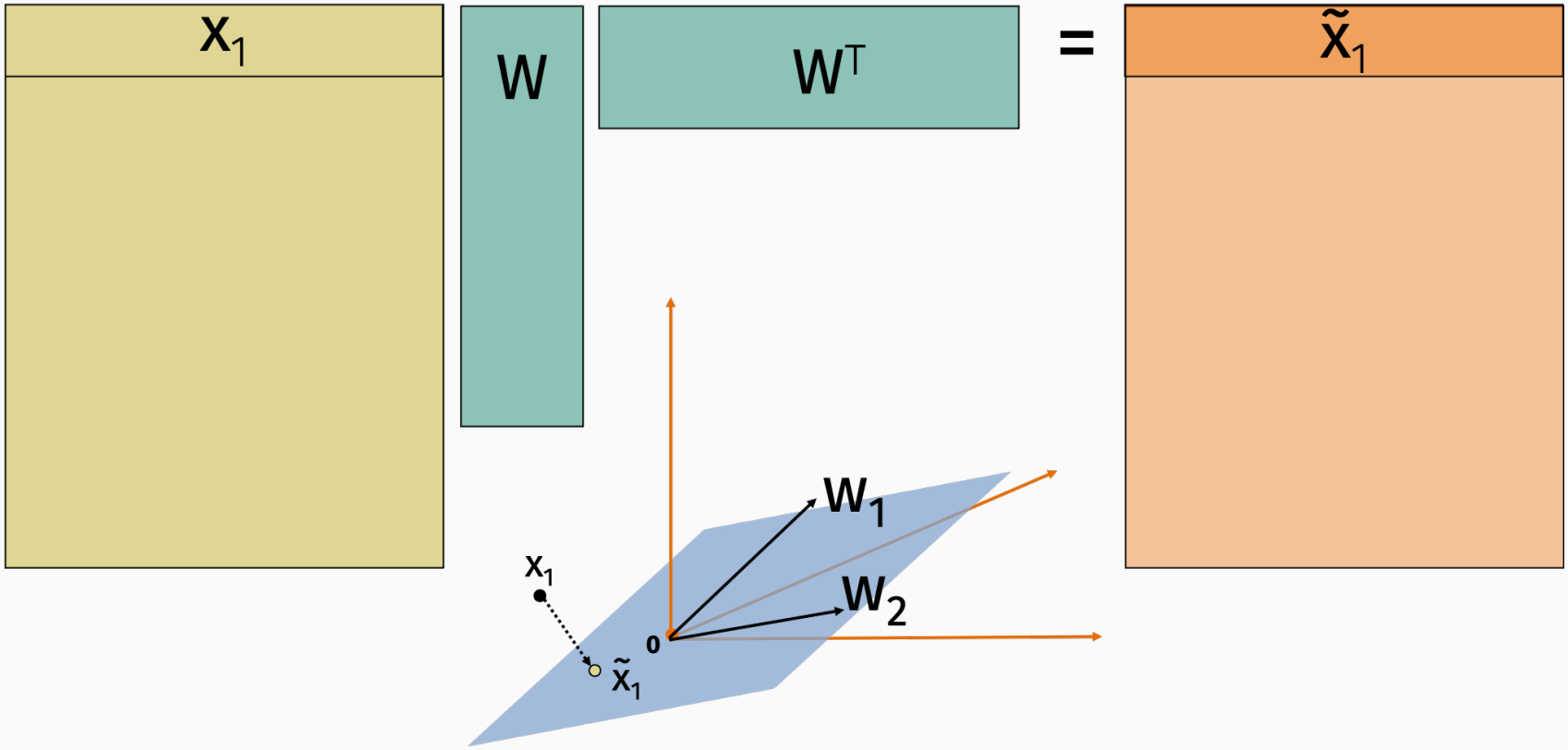
$$\mathbf{c}_i = \mathbf{x}_i\mathbf{W}\left(\mathbf{W}^T\mathbf{W}\right)^{-1} = \mathbf{x}_i\mathbf{W}$$

and therefore $\mathbf{C} = \mathbf{X}\mathbf{W}$.

Our optimal low-rank approximation always has the form: $\mathbf{X}\mathbf{W}\mathbf{W}^T$

Projection Matrices

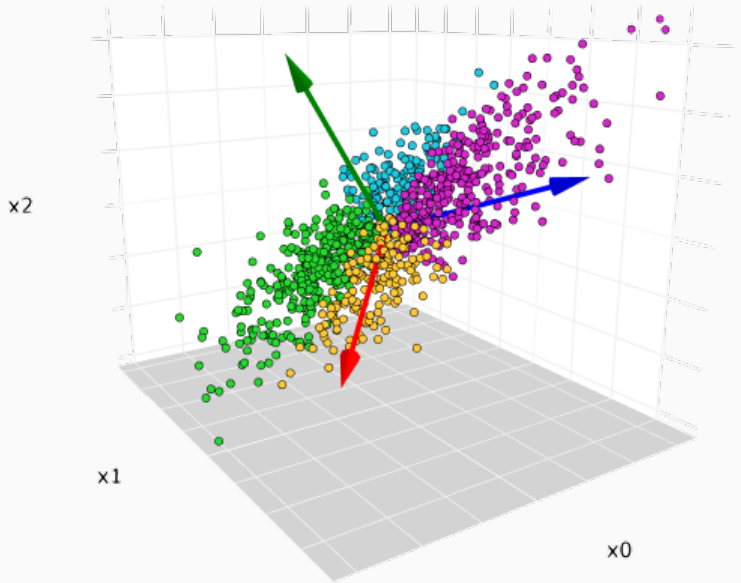
$\mathbf{W}\mathbf{W}^T$ is a symmetric projection matrix.

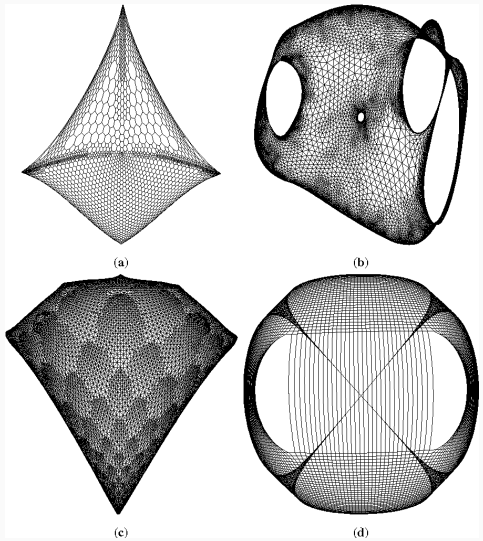


Applications of Low-Rank Approximation

Also useful in:

- Data visualization when $k = 2$ or 3.





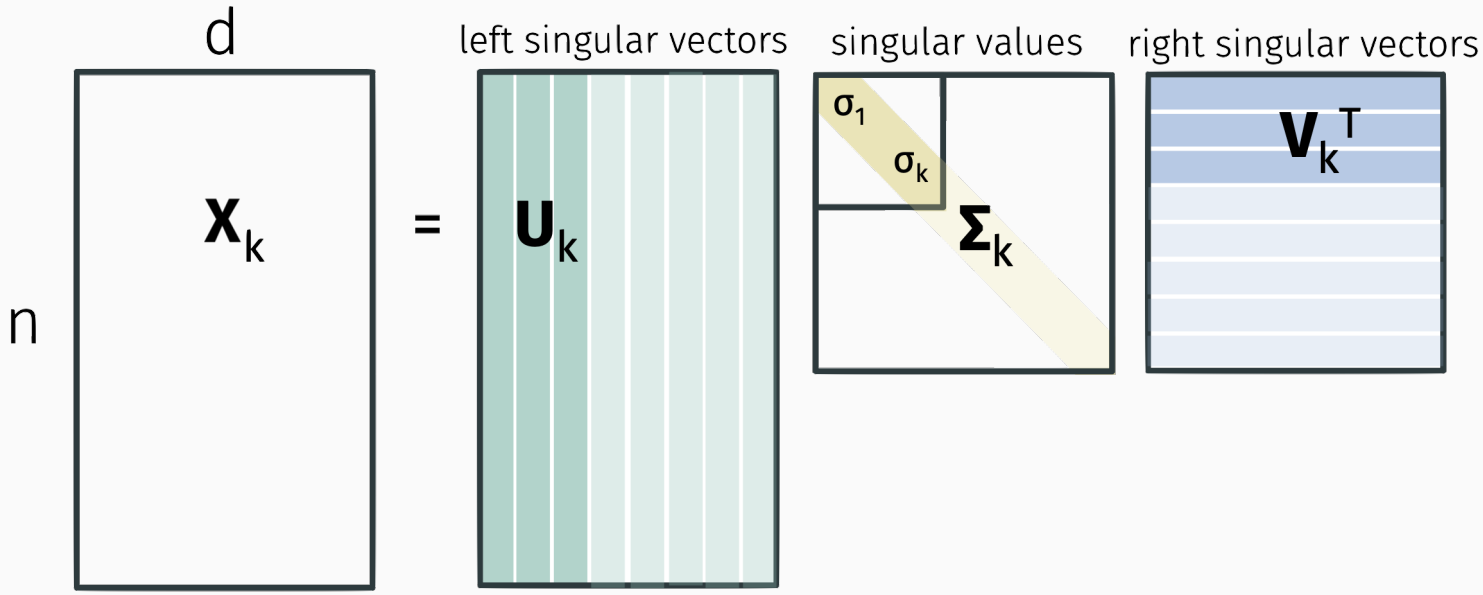
- Data denoising (e.g. distance triangulation).
- Feature selection.

Partial SVD

Key result: Can find the best projection from the singular value decomposition: $\mathbf{X}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$,

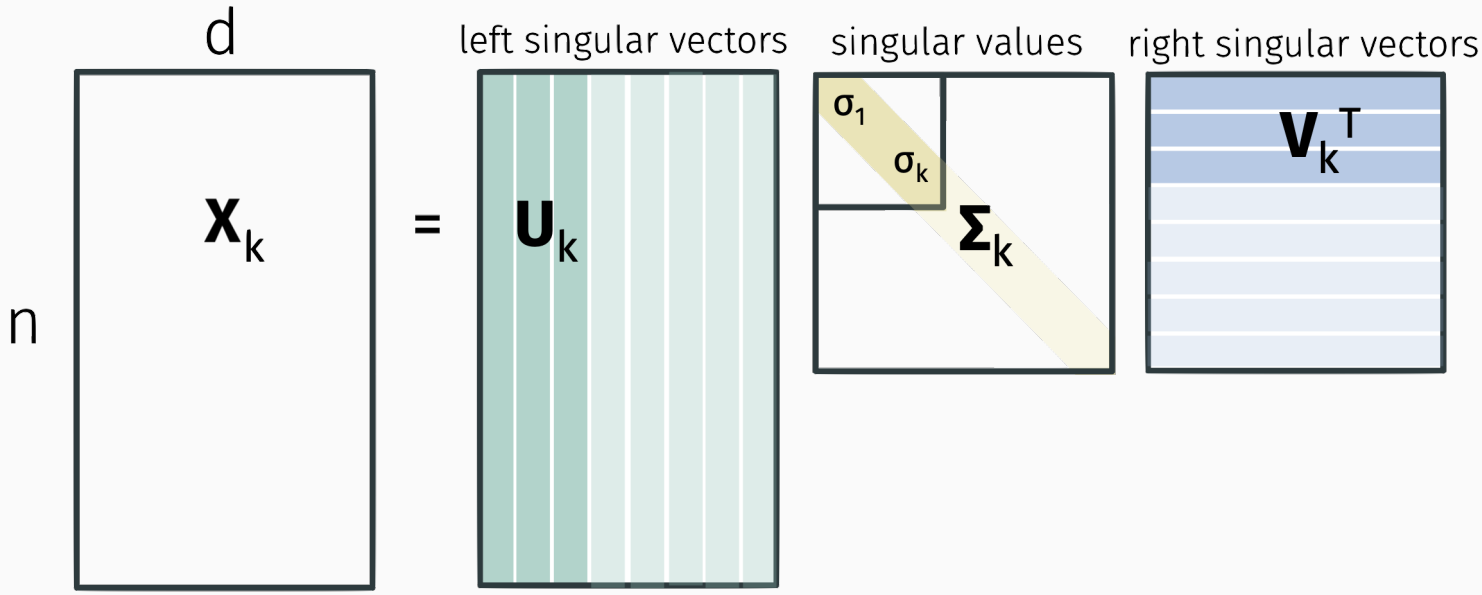
Partial SVD

Key result: Can find the best projection from the singular value decomposition: $\mathbf{X}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$,



Partial SVD

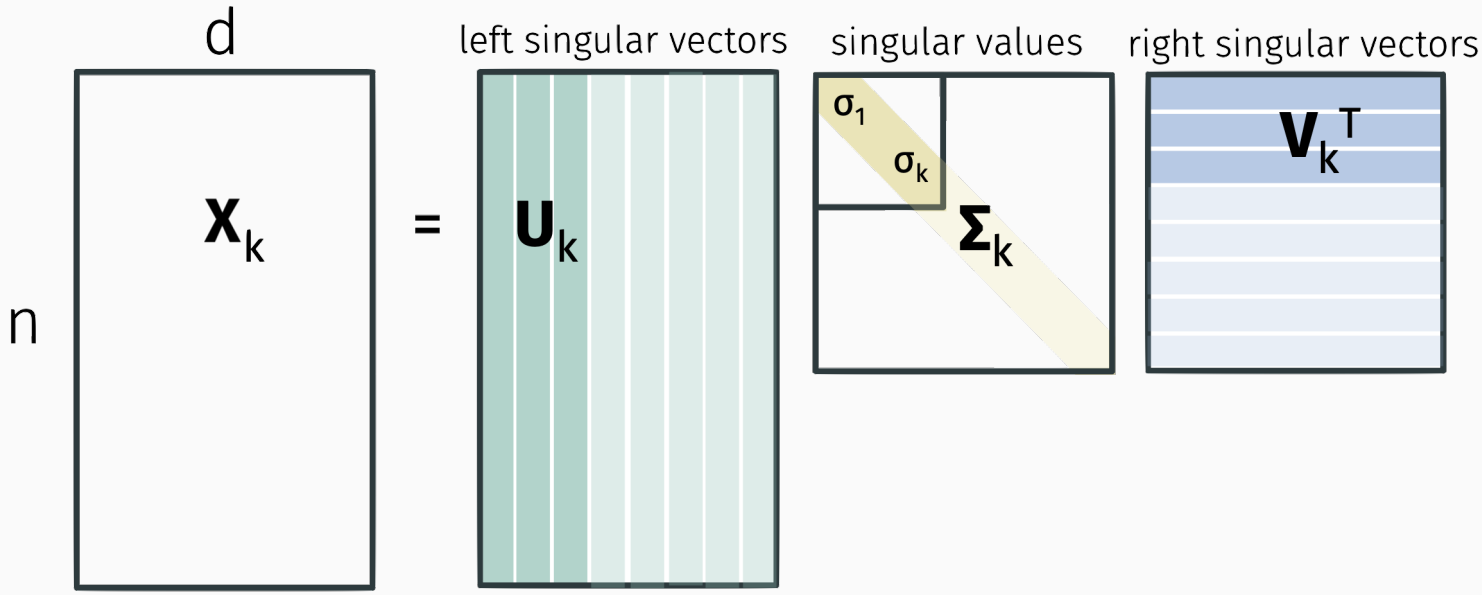
Key result: Can find the best projection from the singular value decomposition: $\mathbf{X}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$,



$$\mathbf{V}_k = \underset{\text{orthogonal } \mathbf{W} \in \mathbb{R}^{d \times k}}{\arg\min} \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2$$

Partial SVD

Key result: Can find the best projection from the singular value decomposition: $\mathbf{X}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$,



$$\mathbf{V}_k = \underset{\text{orthogonal } \mathbf{W} \in \mathbb{R}^{d \times k}}{\arg\min} \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2$$

$$\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}_k\mathbf{V}_k^T = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$$

How do we prove that the partial SVD gives the optimal low-rank approximation?

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{B}\|_F$.

Claim 1: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where V is square. Without loss of generality, can assume $\mathbf{B} = \mathbf{U}\mathbf{Z}\mathbf{V}^T$ for some other rank k matrix $\mathbf{Z}$.

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{B}\|_F$.

Claim 1: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where V is square. Without loss of generality, can assume $\mathbf{B} = \mathbf{U}\mathbf{Z}\mathbf{V}^T$ for some other rank k matrix $\mathbf{Z}$.

Proof:

$$\|\mathbf{X} - \mathbf{B}\|_F = \|(\mathbf{X} - \mathbf{B})\mathbf{V}\|_F = \|\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V}\|_F$$

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{B}\|_F$.

Claim 1: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where V is square. Without loss of generality, can assume $\mathbf{B} = \mathbf{U}\mathbf{Z}\mathbf{V}^T$ for some other rank k matrix $\mathbf{Z}$.

Proof:

$$\begin{aligned}\|\mathbf{X} - \mathbf{B}\|_F &= \|(\mathbf{X} - \mathbf{B})\mathbf{V}\|_F = \|\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V}\|_F \\ &\geq \|\mathbf{U}^T(\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V})\|_F = \|\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\|_F\end{aligned}$$

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{B}\|_F$.

Claim 1: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where V is square. Without loss of generality, can assume $\mathbf{B} = \mathbf{U}\mathbf{Z}\mathbf{V}^T$ for some other rank k matrix $\mathbf{Z}$.

Proof:

$$\begin{aligned}\|\mathbf{X} - \mathbf{B}\|_F &= \|(\mathbf{X} - \mathbf{B})\mathbf{V}\|_F = \|\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V}\|_F \\ &\geq \|\mathbf{U}^T(\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V})\|_F = \|\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\|_F \\ &= \|\mathbf{U}\left(\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\right)\mathbf{V}^T\|_F\end{aligned}$$

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{B}\|_F$.

Claim 1: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where V is square. Without loss of generality, can assume $\mathbf{B} = \mathbf{U}\mathbf{Z}\mathbf{V}^T$ for some other rank k matrix $\mathbf{Z}$.

Proof:

$$\begin{aligned}
\|\mathbf{X} - \mathbf{B}\|_F &= \|(\mathbf{X} - \mathbf{B})\mathbf{V}\|_F = \|\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V}\|_F \\
&\geq \|\mathbf{U}^T(\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V})\|_F = \|\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\|_F \\
&= \|\mathbf{U}\left(\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\right)\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{U}^T\mathbf{X}\mathbf{V}\mathbf{V}^T - \mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F
\end{aligned}$$

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{B}\|_F$.

Claim 1: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where V is square. Without loss of generality, can assume $\mathbf{B} = \mathbf{U}\mathbf{Z}\mathbf{V}^T$ for some other rank k matrix $\mathbf{Z}$.

Proof:

$$\begin{aligned}
\|\mathbf{X} - \mathbf{B}\|_F &= \|(\mathbf{X} - \mathbf{B})\mathbf{V}\|_F = \|\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V}\|_F \\
&\geq \|\mathbf{U}^T(\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V})\|_F = \|\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\|_F \\
&= \|\mathbf{U}\left(\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\right)\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{U}^T\mathbf{X}\mathbf{V}\mathbf{V}^T - \mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F
\end{aligned}$$

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{B}\|_F$.

Claim 1: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where V is square. Without loss of generality, can assume $\mathbf{B} = \mathbf{U}\mathbf{Z}\mathbf{V}^T$ for some other rank k matrix $\mathbf{Z}$.

Proof:

$$\begin{aligned}
\|\mathbf{X} - \mathbf{B}\|_F &= \|(\mathbf{X} - \mathbf{B})\mathbf{V}\|_F = \|\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V}\|_F \\
&\geq \|\mathbf{U}^T(\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V})\|_F = \|\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\|_F \\
&= \|\mathbf{U}\left(\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\right)\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{U}^T\mathbf{X}\mathbf{V}\mathbf{V}^T - \mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F = \|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F
\end{aligned}$$

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{B}\|_F$.

Claim 1: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where V is square. Without loss of generality, can assume $\mathbf{B} = \mathbf{U}\mathbf{Z}\mathbf{V}^T$ for some other rank k matrix $\mathbf{Z}$.

Proof:

$$\begin{aligned}
\|\mathbf{X} - \mathbf{B}\|_F &= \|(\mathbf{X} - \mathbf{B})\mathbf{V}\|_F = \|\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V}\|_F \\
&\geq \|\mathbf{U}^T(\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V})\|_F = \|\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\|_F \\
&= \|\mathbf{U}\left(\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\right)\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{U}^T\mathbf{X}\mathbf{V}\mathbf{V}^T - \mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F = \|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F
\end{aligned}$$

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{B}\|_F$.

Claim 1: Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where V is square. Without loss of generality, can assume $\mathbf{B} = \mathbf{U}\mathbf{Z}\mathbf{V}^T$ for some other rank k matrix $\mathbf{Z}$.

Proof:

$$\begin{aligned}
\|\mathbf{X} - \mathbf{B}\|_F &= \|(\mathbf{X} - \mathbf{B})\mathbf{V}\|_F = \|\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V}\|_F \\
&\geq \|\mathbf{U}^T(\mathbf{X}\mathbf{V} - \mathbf{B}\mathbf{V})\|_F = \|\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\|_F \\
&= \|\mathbf{U}\left(\mathbf{U}^T\mathbf{X}\mathbf{V} - \mathbf{U}^T\mathbf{B}\mathbf{V}\right)\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{U}^T\mathbf{X}\mathbf{V}\mathbf{V}^T - \mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F \\
&= \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\mathbf{B}\mathbf{V}\mathbf{V}^T\|_F = \|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F
\end{aligned}$$

where $Z = \mathbf{U}^T\mathbf{B}\mathbf{V}$. So it only helps to insist that the low-rank approximation is of the form $\mathbf{U}\mathbf{Z}\mathbf{V}^T$. Solve $\min_Z \|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F$!

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F$.

Claim 2: Should choose $\mathbf{Z}$ to be the best rank k approximation to $\mathbf{\Sigma}$. (We will then show this equals $\mathbf{\Sigma}_k$.)

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F$.

Claim 2: Should choose $\mathbf{Z}$ to be the best rank k approximation to $\mathbf{\Sigma}$. (We will then show this equals $\mathbf{\Sigma}_k$.)

Proof:

$$\min_{\text{rank}(Z)=k} \|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F = \min_{\text{rank}(Z)=k} \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F$$

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F$.

Claim 2: Should choose $\mathbf{Z}$ to be the best rank k approximation to $\mathbf{\Sigma}$. (We will then show this equals $\mathbf{\Sigma}_k$.)

Proof:

$$\begin{aligned}\min_{\text{rank}(Z)=k} \|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F &= \min_{\text{rank}(Z)=k} \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F \\ &= \min_{\text{rank}(Z)=k} \|\mathbf{U}(\mathbf{\Sigma} - \mathbf{Z})\mathbf{V}^T\|_F\end{aligned}$$

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F$.

Claim 2: Should choose $\mathbf{Z}$ to be the best rank k approximation to $\mathbf{\Sigma}$. (We will then show this equals $\mathbf{\Sigma}_k$.)

Proof:

$$\begin{aligned}\min_{\text{rank}(Z)=k} \|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F &= \min_{\text{rank}(Z)=k} \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F \\ &= \min_{\text{rank}(Z)=k} \|\mathbf{U}(\mathbf{\Sigma} - \mathbf{Z})\mathbf{V}^T\|_F \\ &= \min_{\text{rank}(Z)=k} \|\mathbf{\Sigma} - \mathbf{Z}\|_F\end{aligned}$$

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F$.

Claim 2: Should choose $\mathbf{Z}$ to be the best rank k approximation to $\mathbf{\Sigma}$. (We will then show this equals $\mathbf{\Sigma}_k$.)

Proof:

$$\begin{aligned}\min_{\text{rank}(Z)=k} \|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F &= \min_{\text{rank}(Z)=k} \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F \\ &= \min_{\text{rank}(Z)=k} \|\mathbf{U}(\mathbf{\Sigma} - \mathbf{Z})\mathbf{V}^T\|_F \\ &= \min_{\text{rank}(Z)=k} \|\mathbf{\Sigma} - \mathbf{Z}\|_F\end{aligned}$$

Optimal Low-Rank Approximation

Goal: Minimize $\|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F$.

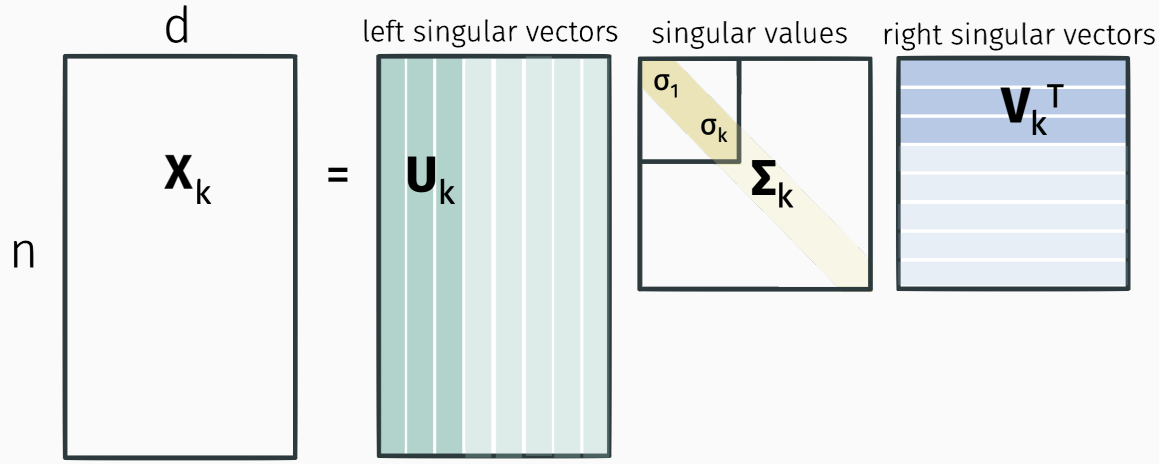
Claim 2: Should choose $\mathbf{Z}$ to be the best rank k approximation to $\mathbf{\Sigma}$. (We will then show this equals $\mathbf{\Sigma}_k$.)

Proof:

$$\begin{aligned}\min_{\text{rank}(Z)=k} \|\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F &= \min_{\text{rank}(Z)=k} \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F \\ &= \min_{\text{rank}(Z)=k} \|\mathbf{U}(\mathbf{\Sigma} - \mathbf{Z})\mathbf{V}^T\|_F \\ &= \min_{\text{rank}(Z)=k} \|\mathbf{\Sigma} - \mathbf{Z}\|_F\end{aligned}$$

Intuitively, the best rank k approximation to a diagonal matrix $\mathbf{\Sigma}$ should be to just take the largest k entries and set the rest to zero.

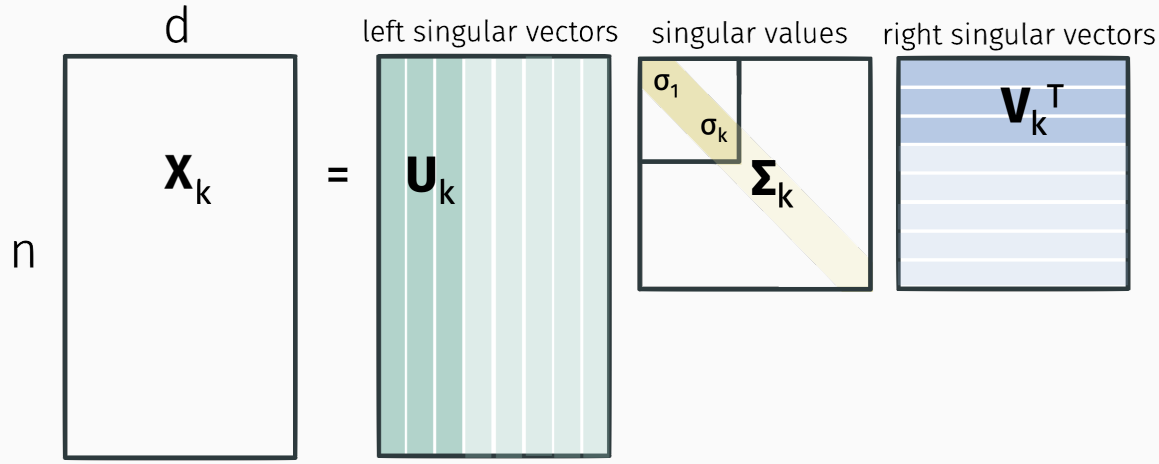
Optimal Low-Rank Approximation



Claim 3: Switching between minimization and maximization:

$$\arg\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2 = \arg\max_{\mathbf{W}\in\mathbb{R}^{d\times k}} \|\mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2$$

Optimal Low-Rank Approximation



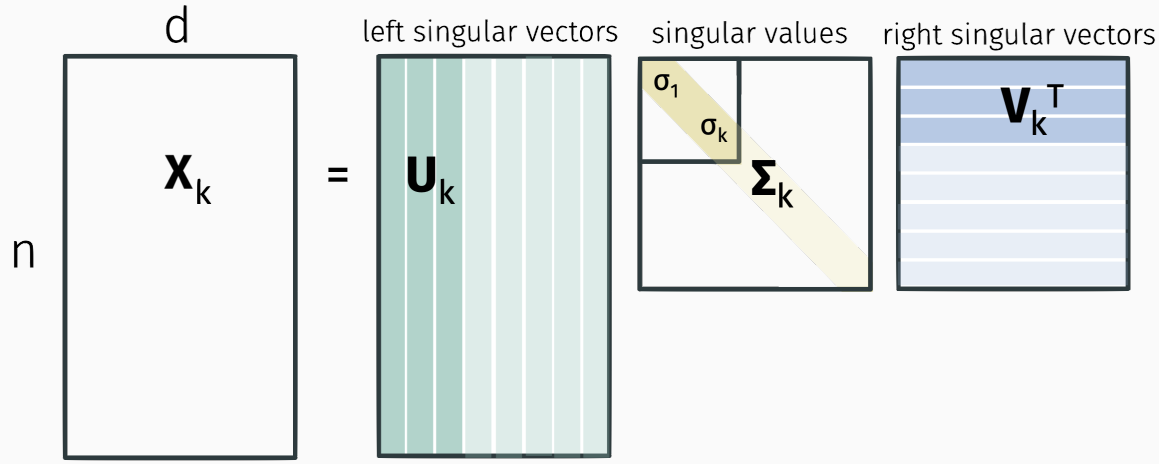
Claim 3: Switching between minimization and maximization:

$$\arg\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2 = \arg\max_{\mathbf{W}\in\mathbb{R}^{d\times k}} \|\mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2$$

Proof: Follows from fact that for all orthogonal $\mathbf{W}$:

$$\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2$$

Optimal Low-Rank Approximation



Claim 3: Switching between minimization and maximization:

$$\arg\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2 = \arg\max_{\mathbf{W}\in\mathbb{R}^{d\times k}} \|\mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2$$

Proof: Follows from fact that for all orthogonal $\mathbf{W}$:

$$\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2$$

$$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2 = \min_{\mathbf{W}} \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2 = \min_{\mathbf{W}} -\|\mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2$$

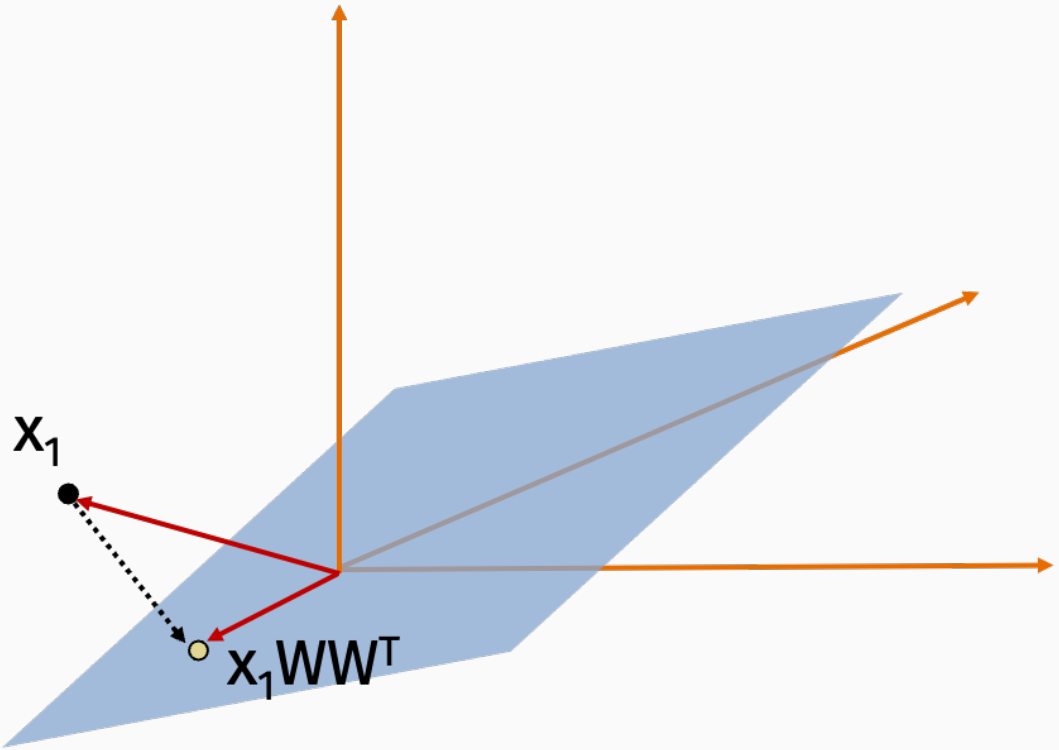
Optimal Low-Rank Approximation

Remains to show $\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2$

Optimal Low-Rank Approximation

Remains to show $\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2$

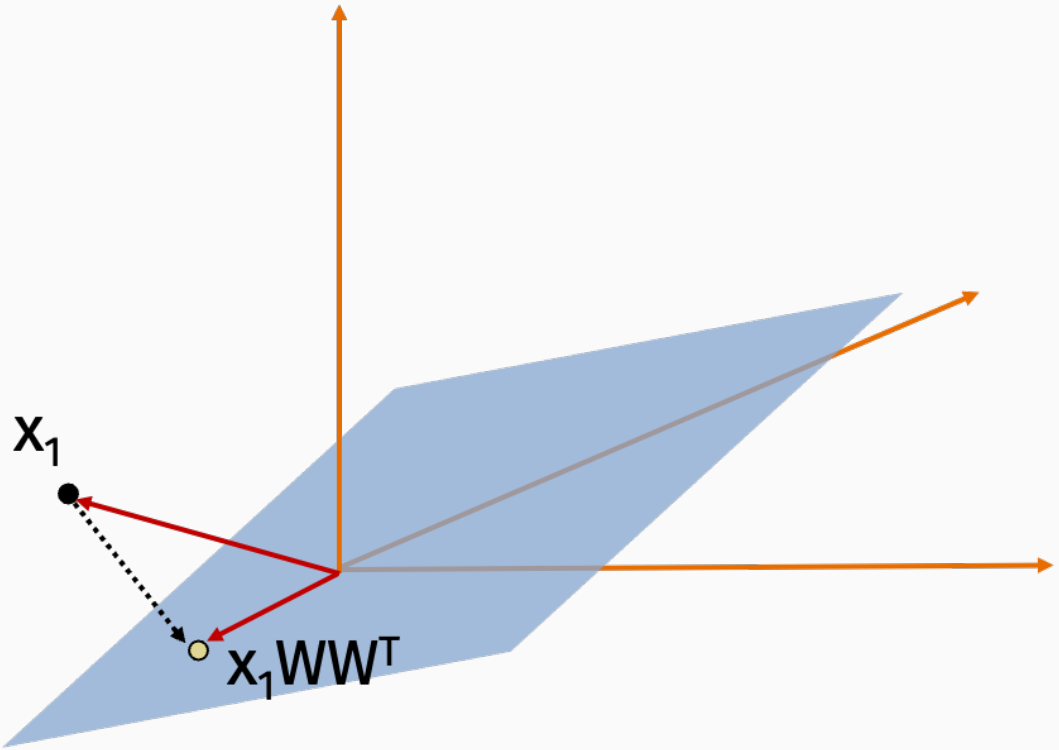
Let $\mathbf{X}_1$ be the first row of X. Then,



Optimal Low-Rank Approximation

Remains to show $\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{W}\mathbf{W}^T\|_F^2$

Let $\mathbf{X}_1$ be the first row of X. Then,



Pythagorean theorem: $\|\mathbf{X}_1\|_2^2 = \|\mathbf{X}_1\mathbf{W}\mathbf{W}^T\|_2^2 + \|\mathbf{X}_1 - \mathbf{X}_1\mathbf{W}\mathbf{W}^T\|_2^2$.

Optimal Low-Rank Approximation

Final Step: Let $\mathbf{W}^* = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]$ be the first k standard basis vectors. Then we claim that $\mathbf{W}^* = \arg\max_{\mathbf{W}} \|\mathbf{\Sigma W W}^T\|_F^2$.

Optimal Low-Rank Approximation

Final Step: Let $\mathbf{W}^* = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]$ be the first k standard basis vectors. Then we claim that $\mathbf{W}^* = \arg\max_{\mathbf{W}} \|\mathbf{\Sigma W W}^T\|_F^2$.

Proof: Observe, $\max_{\mathbf{W}} \|\mathbf{\Sigma W W}^T\|_F^2 = \max_{\mathbf{W}} \|\mathbf{\Sigma W}\|_F^2$ and thus

$$\|\mathbf{\Sigma W}\|_F^2 = \sum_{i=1}^{d} \|\sigma_i \mathbf{W}_i\|_2^2 = \sum_{i=1}^{d} \sigma_i^2 \|\mathbf{W}_i\|_2^2$$

where $\mathbf{W}_i \leq 1$ and $\sum_{i=1}^{d} \|\mathbf{W}_i\|_2^2 = k$.

Optimal Low-Rank Approximation

Final Step: Let $\mathbf{W}^* = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]$ be the first k standard basis vectors. Then we claim that $\mathbf{W}^* = \arg\max_{\mathbf{W}} \|\mathbf{\Sigma W W}^T\|_F^2$.

Proof: Observe, $\max_{\mathbf{W}} \|\mathbf{\Sigma W W}^T\|_F^2 = \max_{\mathbf{W}} \|\mathbf{\Sigma W}\|_F^2$ and thus

$$\|\mathbf{\Sigma W}\|_F^2 = \sum_{i=1}^{d} \|\sigma_i \mathbf{W}_i\|_2^2 = \sum_{i=1}^{d} \sigma_i^2 \|\mathbf{W}_i\|_2^2$$

where $\mathbf{W}_i \leq 1$ and $\sum_{i=1}^{d} \|\mathbf{W}_i\|_2^2 = k$.

Optimal Low-Rank Approximation

Final Step: Let $\mathbf{W}^* = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]$ be the first k standard basis vectors. Then we claim that $\mathbf{W}^* = \arg\max_{\mathbf{W}} \|\mathbf{\Sigma W W}^T\|_F^2$.

Proof: Observe, $\max_{\mathbf{W}} \|\mathbf{\Sigma W W}^T\|_F^2 = \max_{\mathbf{W}} \|\mathbf{\Sigma W}\|_F^2$ and thus

$$\|\mathbf{\Sigma W}\|_F^2 = \sum_{i=1}^{d} \|\sigma_i \mathbf{W}_i\|_2^2 = \sum_{i=1}^{d} \sigma_i^2 \|\mathbf{W}_i\|_2^2$$

where $\mathbf{W}_i \leq 1$ and $\sum_{i=1}^{d} \|\mathbf{W}_i\|_2^2 = k$.

Claim: The objective is maximized when $\|\mathbf{W}_i\|_2^2 = 1$ for the k largest σ_i's and 0 otherwise.

Optimal Low-Rank Approximation

Final Step: Let $\mathbf{W}^* = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]$ be the first k standard basis vectors. Then we claim that $\mathbf{W}^* = \arg\max_{\mathbf{W}} \|\mathbf{\Sigma W W}^T\|_F^2$.

Proof: Observe, $\max_{\mathbf{W}} \|\mathbf{\Sigma W W}^T\|_F^2 = \max_{\mathbf{W}} \|\mathbf{\Sigma W}\|_F^2$ and thus

$$\|\mathbf{\Sigma W}\|_F^2 = \sum_{i=1}^{d} \|\sigma_i \mathbf{W}_i\|_2^2 = \sum_{i=1}^{d} \sigma_i^2 \|\mathbf{W}_i\|_2^2$$

where $\mathbf{W}_i \leq 1$ and $\sum_{i=1}^{d} \|\mathbf{W}_i\|_2^2 = k$.

Claim: The objective is maximized when $\|\mathbf{W}_i\|_2^2 = 1$ for the k largest σ_i's and 0 otherwise.

Suffices to show that $\sum_{i=1}^{d} \sigma_i^2 \|\mathbf{W}_i\|_2^2 \leq \sum_{i=1}^{k} \sigma_i^2$.

Optimal Low-Rank Approximation

Claim: Let $0 \leq w_i \leq 1$ and $\sum_{i=1}^{d} w_i = k$. Then,

$$\sum_{i=1}^{d} \sigma_i^2 w_i \leq \sum_{i=1}^{k} \sigma_i^2$$

Optimal Low-Rank Approximation

Claim: Let $0 \leq w_i \leq 1$ and $\sum_{i=1}^{d} w_i = k$. Then,

$$\sum_{i=1}^{d} \sigma_i^2 w_i \leq \sum_{i=1}^{k} \sigma_i^2$$

Proof:

$$\sum_{i=1}^{d} \sigma_i^2 w_i = \sum_{i=1}^{k} \sigma_i^2 w_i + \sum_{i=k+1}^{d} \sigma_i^2 w_i \leq \sum_{i=1}^{k} \sigma_i^2 w_i + \sigma_k^2 \sum_{i=k+1}^{d} w_i$$

Optimal Low-Rank Approximation

Claim: Let $0 \leq w_i \leq 1$ and $\sum_{i=1}^{d} w_i = k$. Then,

$$\sum_{i=1}^{d} \sigma_i^2 w_i \leq \sum_{i=1}^{k} \sigma_i^2$$

Proof:

$$\begin{aligned}\sum_{i=1}^{d} \sigma_i^2 w_i &= \sum_{i=1}^{k} \sigma_i^2 w_i + \sum_{i=k+1}^{d} \sigma_i^2 w_i \leq \sum_{i=1}^{k} \sigma_i^2 w_i + \sigma_k^2 \sum_{i=k+1}^{d} w_i \\ &= \sum_{i=1}^{k} \sigma_i^2 w_i + \sigma_k^2 \left(\sum_{i=1}^{d} w_i - \sum_{i=1}^{k} w_i \right)\end{aligned}$$

Optimal Low-Rank Approximation

Claim: Let $0 \leq w_i \leq 1$ and $\sum_{i=1}^{d} w_i = k$. Then,

$$\sum_{i=1}^{d} \sigma_i^2 w_i \leq \sum_{i=1}^{k} \sigma_i^2$$

Proof:

$$\begin{aligned}
\sum_{i=1}^{d} \sigma_i^2 w_i &= \sum_{i=1}^{k} \sigma_i^2 w_i + \sum_{i=k+1}^{d} \sigma_i^2 w_i \leq \sum_{i=1}^{k} \sigma_i^2 w_i + \sigma_k^2 \sum_{i=k+1}^{d} w_i \\
&= \sum_{i=1}^{k} \sigma_i^2 w_i + \sigma_k^2 \left(\sum_{i=1}^{d} w_i - \sum_{i=1}^{k} w_i \right) \\
&= \sum_{i=1}^{k} (\sigma_i^2 - \sigma_k^2) w_i + \sigma_k^2 k
\end{aligned}$$

Optimal Low-Rank Approximation

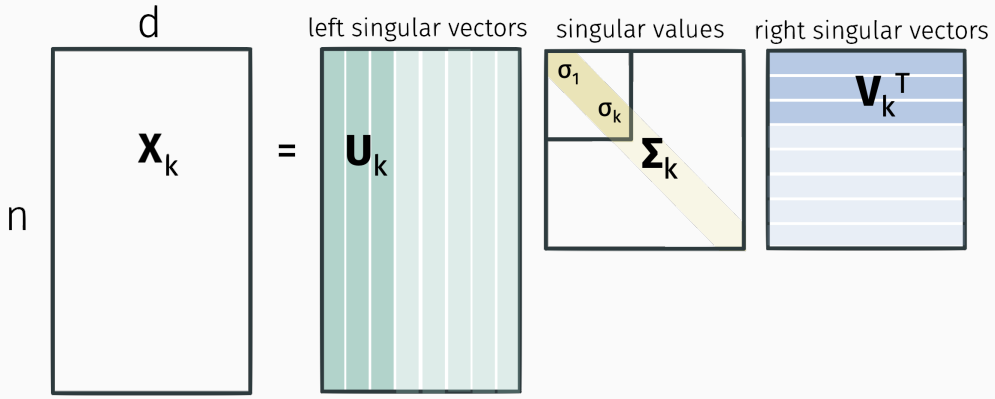
Claim: Let $0 \leq w_i \leq 1$ and $\sum_{i=1}^{d} w_i = k$. Then,

$$\sum_{i=1}^{d} \sigma_i^2 w_i \leq \sum_{i=1}^{k} \sigma_i^2$$

Proof:

$$\begin{aligned}
\sum_{i=1}^{d} \sigma_i^2 w_i &= \sum_{i=1}^{k} \sigma_i^2 w_i + \sum_{i=k+1}^{d} \sigma_i^2 w_i \leq \sum_{i=1}^{k} \sigma_i^2 w_i + \sigma_k^2 \sum_{i=k+1}^{d} w_i \\
&= \sum_{i=1}^{k} \sigma_i^2 w_i + \sigma_k^2 \left(\sum_{i=1}^{d} w_i - \sum_{i=1}^{k} w_i \right) \\
&= \sum_{i=1}^{k} (\sigma_i^2 - \sigma_k^2) w_i + \sigma_k^2 k \\
&\leq \sum_{i=1}^{k} (\sigma_i^2 - \sigma_k^2) + \sigma_k^2 k = \sum_{i=1}^{k} \sigma_i^2
\end{aligned}$$

Conclusion:



Theorem: The best rank k approximation to $\mathbf{X}$ is given by the partial SVD $\mathbf{X}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$.