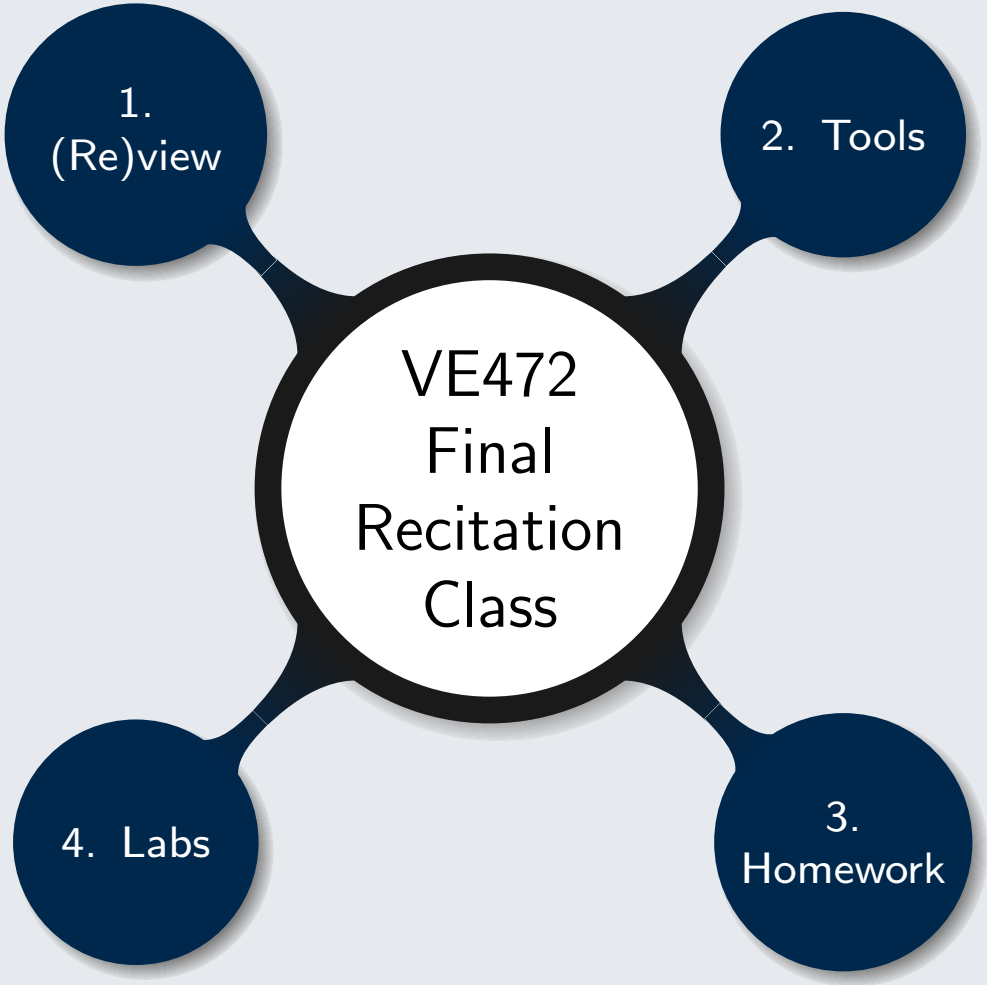
JOINT INSTITUTE

交大密西根学院

VE472 Final Recitation Class

July 22, 2021

Gu Zhenhao, Xie Jinglei


1.
(Re)view
2. Tools
VE472
Final
Recitation
Class
4. Labs
3.
Homework

1. (Re)view

Statistical Reminders

Some key terms:

- *Mean*: average position

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

- *Standard deviation*: average distance from mean

$$\sigma = \sqrt{\frac{(\sum_{i=1}^{n} X_i - \bar{X})^2}{n-1}}$$

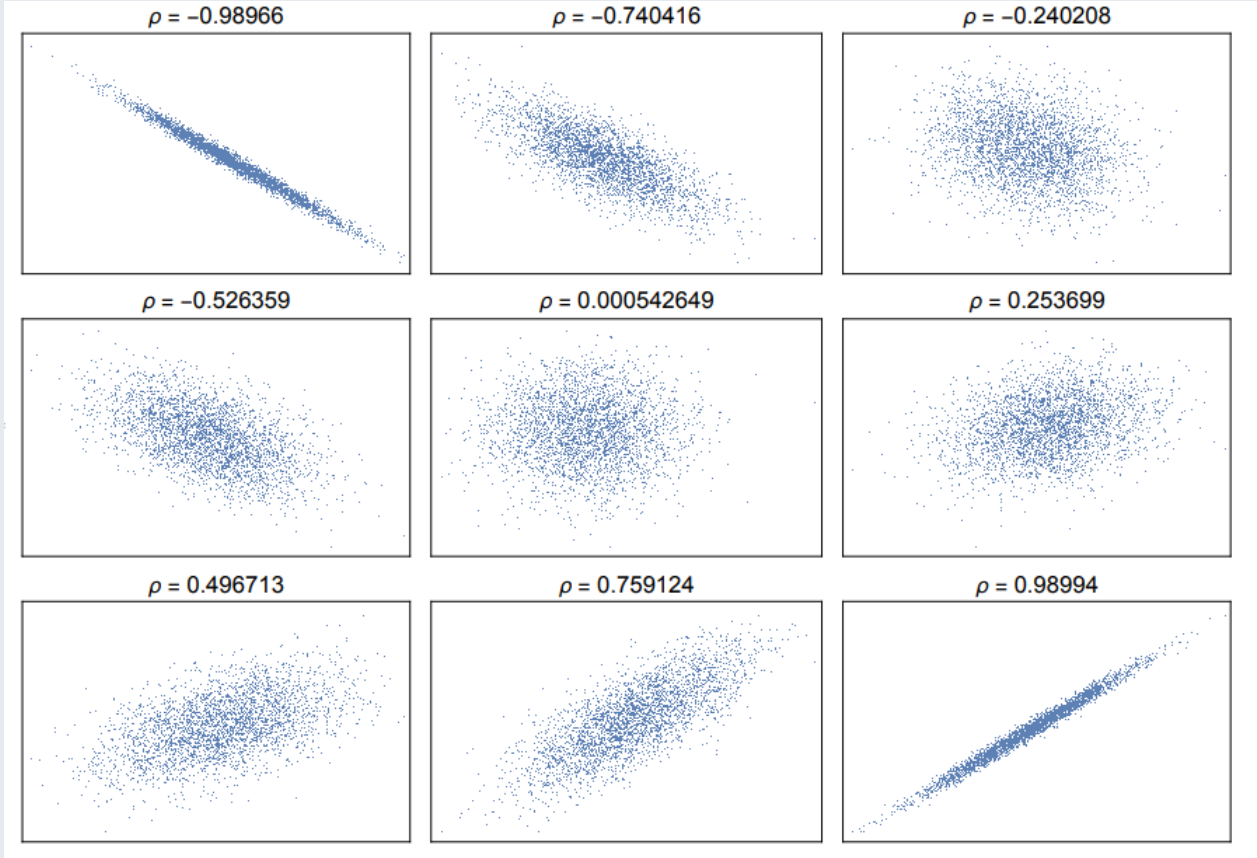
Generally, the first step of PCA and Gradient Descent should be data standardization:

$$Z_{i,j} = \frac{X_{i,j} - \bar{X}_i}{\sigma_{X_i}}$$

Statistical Reminders

- *Covariance*: Tendency of linear relationship between two columns X and Y.

$$\sigma_{X,Y} = \frac{\sum_{i=1}^{m}(X_i - \bar{X})(Y_i - \bar{Y})}{m-1}$$



Statistical Reminders

- *Covariance Matrix*:

$$\text{Cov}[\mathbf{X}] = \begin{bmatrix} \sigma_{X_1,X_1} & \cdots & \sigma_{X_1,X_n} \\ \vdots & \ddots & \vdots \\ \sigma_{X_n,X_1} & \cdots & \sigma_{X_n,X_n} \end{bmatrix}$$

Or in compact form[1],

$$\text{Cov}[\mathbf{X}] = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X}_i)(X_i - \bar{X}_i)^T$$

[1] Reference: en.wikipedia.org/wiki/Estimation_of_covariance_matrices

Linear Algebra Reminders

- *Rank*: The **rank** of a matrix A is the number of linearly independent columns in A.

For example, the matrix

$$A = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 1 & 0 & 2 \end{bmatrix}$$

has rank 2 since the first two columns are linearly independent, while the third column can be expressed as

$$a_3 = 2a_1 + a_2$$

which is a linear combination of the first two columns.

Linear Algebra Reminders

- *Inner product*: A way to multiply vectors together.

$$\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$$

Properties (in real vector space):

1. $\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$
2. $\langle \alpha v, w \rangle = \alpha \langle v, w \rangle$
3. $\langle v, w \rangle = \langle w, v \rangle$
4. $\langle v, v \rangle \geq 0$ and equal if and only if $v = 0$

- *Inner product in Euclidean space*: which is also called dot product,

$$\langle x, y \rangle = \left\langle \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \right\rangle = x^T y = \sum_{i=1}^{n} x_i y_i$$

Linear Algebra Reminders

- *Orthogonal Matrix*: Matrix with orthonormal columns. For each columns in an orthogonal matrix A, we have

$$\langle a_i, a_j \rangle = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases}$$

Or simply, $A^T A = I$, where I is the identity matrix,

$$I = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

Linear Algebra Reminders

- *Eigenvalues*: A number λ is an **eigenvalue** of an $n \times n$ matrix A if and only if there exists a non-zero $n \times 1$ vector x such that

$$\lambda x = Ax$$

The vector x is called the **eigenvector** corresponding to λ.

To calculate this, we need to find the λ such that $\det(A - \lambda I) = 0$.

Linear Algebra Reminders

Example from Vv417. calculate the eigenvalues and eigenvectors of

$$A = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -1 \end{bmatrix}$$

Linear Algebra Reminders

Example from Vv417. calculate the eigenvalues and eigenvectors of

$$A = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -1 \end{bmatrix}$$

Solution.

$$\begin{aligned} \det(A - \lambda I) &= \det \begin{bmatrix} 2-\lambda & 0 & 1 \\ 1 & 1-\lambda & 1 \\ -2 & 0 & -1-\lambda \end{bmatrix} \\ &= (2-\lambda)(1-\lambda)(-1-\lambda) + 2(1-\lambda) \\ &= -\lambda(\lambda-1)^2 = 0 \end{aligned}$$

Solving this equation gives $\lambda_1 = 0$, $\lambda_2 = \lambda_3 = 1$, with algebraic multiplicity 1 and 2 respectively.

Linear Algebra Reminders

Next we calculate the corresponding eigenvectors. We need to find a vector $x = (a, b, c)$ such that $Ax = \lambda x$.

1. For $\lambda_1 = 0$, we find

$$\begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow 2a = 2b = -c$$

   So we can say that $x_1 = (1, 1, -2)$ is an eigenvector of λ_1.

2. For $\lambda_2 = \lambda_3 = 1$,

$$\begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \Leftrightarrow a + c = 0$$

   So $x_2 = (0, 1, 0)$, $x_3 = (1, 0, -1)$ are both eigenvectors of $\lambda = 1$.

Linear Algebra Reminders

Remarks on eigenvalues:

1. If A has rank n, it has at most n eigenvalues. Rank is the number of linearly independent columns.
2. The number of linearly independent eigenvectors is equal to the algebraic multiplicity of the corresponding eigenvalue when A is diagonalizable.

2. Tools

Principal Component Analysis

Purpose: Dimension reduction for more efficient analysis. **Idea**: Find a vector subspace that

- We can project the dataset onto the subspace, and
- the variance of data is maximized.

Principal Component Analysis

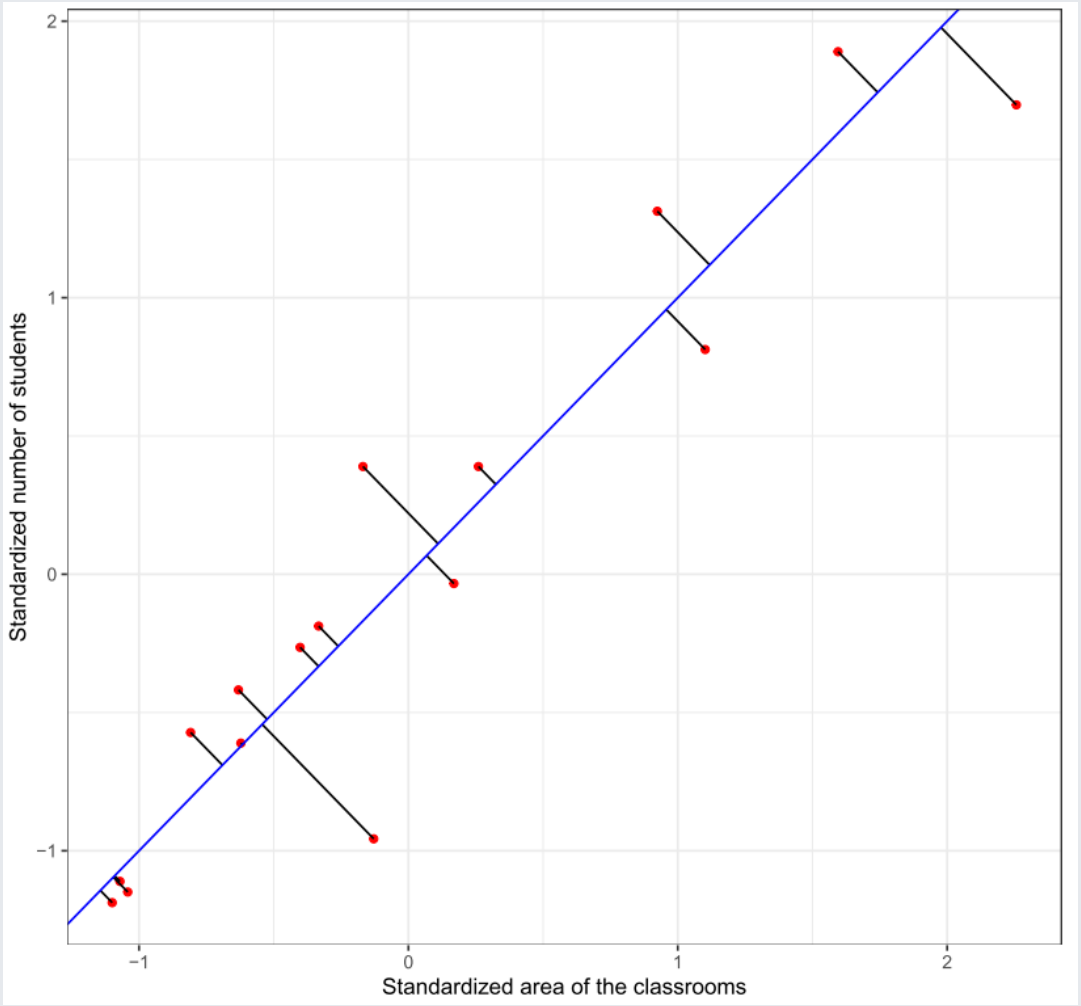


Figure: In this case, we need to find a line such that the distance from data points to the line is minimized (black segments).

Principal Component Analysis

Steps:

1. Standardize the data so that the data is centered at 0 and each feature contribute equally.
2. Find the covariance matrix of the data Σ,
3. Calculate $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$ be the eigenvalues of Σ and $p_1, p_2 \ldots, p_k$ be the corresponding orthonormal eigenvectors.
4. We can express a data point x_i in the subspace spanned by the first a eigenvectors as

$$x \approx \sum_{j=1}^{a} \underbrace{\langle x_i, p_j \rangle p_j}_{\text{projection of } x \text{ onto } p_j}$$

Singular Value Decomposition

Let X be rank r matrix of shape $m \times n$ with $m \geq n$, then we have $\sigma_1 \geq \cdots \geq \sigma_r > 0$,

$$X = U\Sigma V^T = \begin{bmatrix} u_1 & \dots & u_r \end{bmatrix} \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_r \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \end{bmatrix}$$

where U and V are orthogonal matrices of size $m \times r$ and $r \times n$, respectively.

Singular Value Decomposition

Purpose: Provide stable solution to linear systems of equations and regression problems.
With an invertible matrix $X = U\Sigma V^T$, we can easily calculate that $X^{-1} = V\Sigma^{-1}U^T$.
For example we want to find

$$\arg\min_{b\in\mathbb{R}^m} \|y - Xb\|^2$$

We simply need to calculate

$$b = V\Sigma^{-1}U^T y$$

Properties:

- More numerically stable than LU, Cholesky, ...
- A bit slower than LU and Cholesky,
- Exist for any matrix, invertible or not.

Singular Value Decomposition

Steps:

- calculate the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_r$ of $A^T A$.
  We can prove that $\lambda_i \geq 0$:

Proof.

$$\begin{aligned}
\underbrace{\|Xv_i\|^2}_{\geq 0} &= (Xv_i)^T Xv_i \\
&= v_i^T X^T X v_i \\
&= v_i^T \lambda_i v_i \\
&= \lambda_i v_i^T v_i = \lambda \underbrace{\|v_i\|^2}_{\geq 0}
\end{aligned}$$

□

- Calculate v_i, the corresponding orthonormal eigenvector of λ_i.

Singular Value Decomposition

Steps (Continued):

- Find singular value of X, $\sigma_i = \sqrt{\lambda_i}$.
- Determine $u_j \in \mathbb{R}^m$, $j = 1, \ldots, k$ by finding

$$u_j = \frac{Xv_j}{\|Xv_j\|}$$

Practice doing SVD by hand using the matrix

$$X = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -1 \end{bmatrix}$$

Singular Value Decomposition

We first find the eigenvalues of

$$X^T X = \begin{bmatrix} 9 & 1 & 5 \\ 1 & 1 & 1 \\ 5 & 1 & 3 \end{bmatrix}$$

by calculating

$$\begin{aligned} \det(X^T X - \lambda I) = & \det \begin{bmatrix} 9-\lambda & 1 & 5 \\ 1 & 1-\lambda & 1 \\ 5 & 1 & 3-\lambda \end{bmatrix} \\ = & (9-\lambda)(1-\lambda)(3-\lambda) + 5 + 5 \\ & -25(1-\lambda) - (9-\lambda) - (3-\lambda) \\ = & -\lambda^3 + 13\lambda^2 - 12\lambda \\ = & -\lambda(\lambda - 12)(\lambda - 1) \end{aligned}$$

So we have two positive eigenvalues $\lambda_1 = 12$ and $\lambda_2 = 1$.

Singular Value Decomposition

This gives singular values $\sigma_1 = \sqrt{12}$ and $\sigma_2 = 1$, which gives

$$\Sigma = \begin{bmatrix} \sqrt{12} & 0 \\ 0 & 1 \end{bmatrix}$$

- For $\lambda_1 = 12$, we solve $x = (a, b, c)$ such that

$$\begin{bmatrix} 9 & 1 & 5 \\ 1 & 1 & 1 \\ 5 & 1 & 3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 12 \begin{bmatrix} a \\ b \\ c \end{bmatrix} \Leftrightarrow a = 7b, c = 4b$$

  So the eigenvector is multiple of $(7, 1, 4)$. We want an orthonormal vector, so we have

$$v_1 = \frac{1}{\sqrt{66}}(7, 1, 4) = \left(\frac{7}{\sqrt{66}}, \frac{1}{\sqrt{66}}, \frac{4}{\sqrt{66}} \right)$$

Singular Value Decomposition

- For $\lambda_2 = 11$, we solve $x = (a, b, c)$ such that

$$\begin{bmatrix} 9 & 1 & 5 \\ 1 & 1 & 1 \\ 5 & 1 & 3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \Leftrightarrow b = -3a, c = -a$$

  So the eigenvector is multiple of $(-1, 3, 1)$. We want an orthonormal vector, so we have

$$v_2 = \frac{1}{\sqrt{11}}(-1, 3, 1) = \left(-\frac{1}{\sqrt{11}}, \frac{3}{\sqrt{11}}, \frac{1}{\sqrt{11}}\right)$$

So we have

$$V^T = \begin{bmatrix} 7/\sqrt{66} & 1/\sqrt{66} & 4/\sqrt{66} \\ -1/\sqrt{11} & 3/\sqrt{11} & 1/\sqrt{11} \end{bmatrix}$$

Singular Value Decomposition

We can see that

- For v_1,

$$Xv_1 = \frac{1}{\sqrt{66}} \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -1 \end{bmatrix} \begin{bmatrix} 7 \\ 1 \\ 4 \end{bmatrix} = \frac{1}{\sqrt{66}} \begin{bmatrix} 18 \\ 12 \\ -18 \end{bmatrix} = c \begin{bmatrix} 3 \\ 2 \\ -3 \end{bmatrix}$$

  for some constant c. Again we want to normalize this vector,

$$u_1 = \frac{1}{\sqrt{22}}(3, 2, -3)$$

- For v_2, similarly, we have

$$u_2 = Xv_2 = \frac{1}{\sqrt{11}} \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -1 \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{11}} \begin{bmatrix} -1 \\ 3 \\ 1 \end{bmatrix}$$

Singular Value Decomposition

So we have

$$U = \begin{bmatrix} 3/\sqrt{22} & -1/\sqrt{11} \\ 2/\sqrt{22} & 3/\sqrt{11} \\ -3/\sqrt{22} & 1/\sqrt{11} \end{bmatrix}$$

And we have the whole SVD $X = U\Sigma V^T$, or

$$X = \begin{bmatrix} 3/\sqrt{22} & -1/\sqrt{11} \\ 2/\sqrt{22} & 3/\sqrt{11} \\ -3/\sqrt{22} & 1/\sqrt{11} \end{bmatrix} \begin{bmatrix} \sqrt{12} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 7/\sqrt{66} & 1/\sqrt{66} & 4/\sqrt{66} \\ -1/\sqrt{11} & 3/\sqrt{11} & 1/\sqrt{11} \end{bmatrix}$$

QR Decomposition

For any $m \times n$ matrix X with linearly independent columns, we can write $X = QR$, where

- Q is an $m \times n$ orthogonal matrix,
- R is an $n \times n$ upper triangular matrix.

Properties:

- Slower than SVD,
- The most numerically stable method.

QR Decomposition

Idea: Find a new coordinate system so that we can store the information of X in an upper triangular matrix.

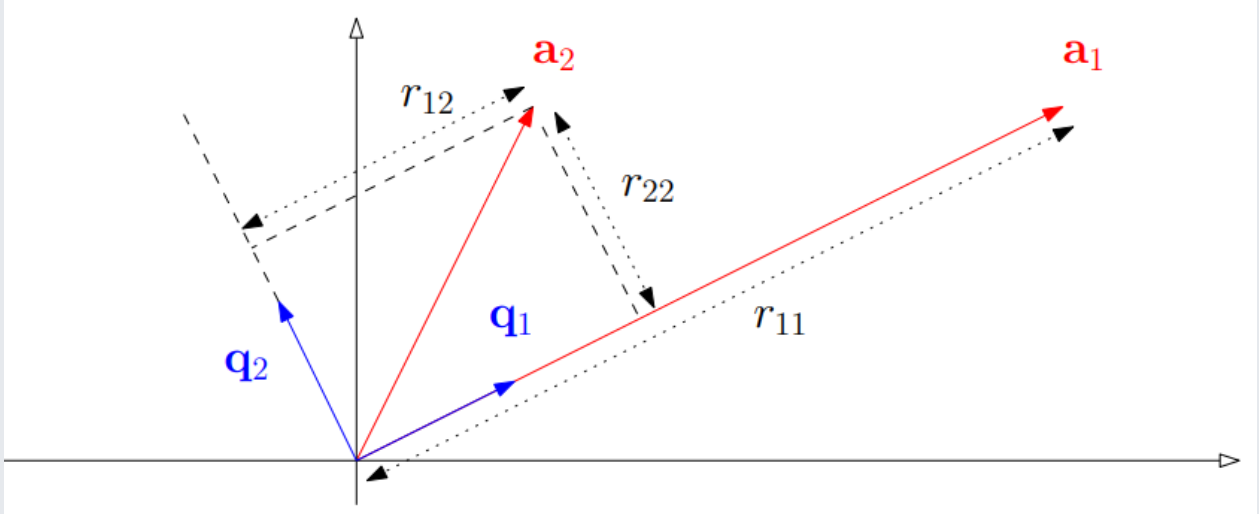


Figure: We find a new orthonormal basis (q_1, q_2), so that the columns in X, a_1 only is a multiple of q_1 and a_2 is only a linear combination of q_1 and q_2.

$$X = \begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix} = \begin{bmatrix} | & | \\ q_1 & q_2 \\ | & | \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix}$$

QR Decomposition

Several ways of finding the orthonormal basis Q:

- **Projection**: Gram-Schmidt, unstable
- **Reflection**: Householder, more stable and fast
- **Rotation**: Givens, most stable but slow

Steps:

1. Determine Q first with one of the above methods,
2. Find R by looking at the projections of the columns of X onto the new basis.

3. Homework

Gradient Descent

Purpose: An easy way to find the minimum/maximum of a function $f(w, x)$.
Idea:

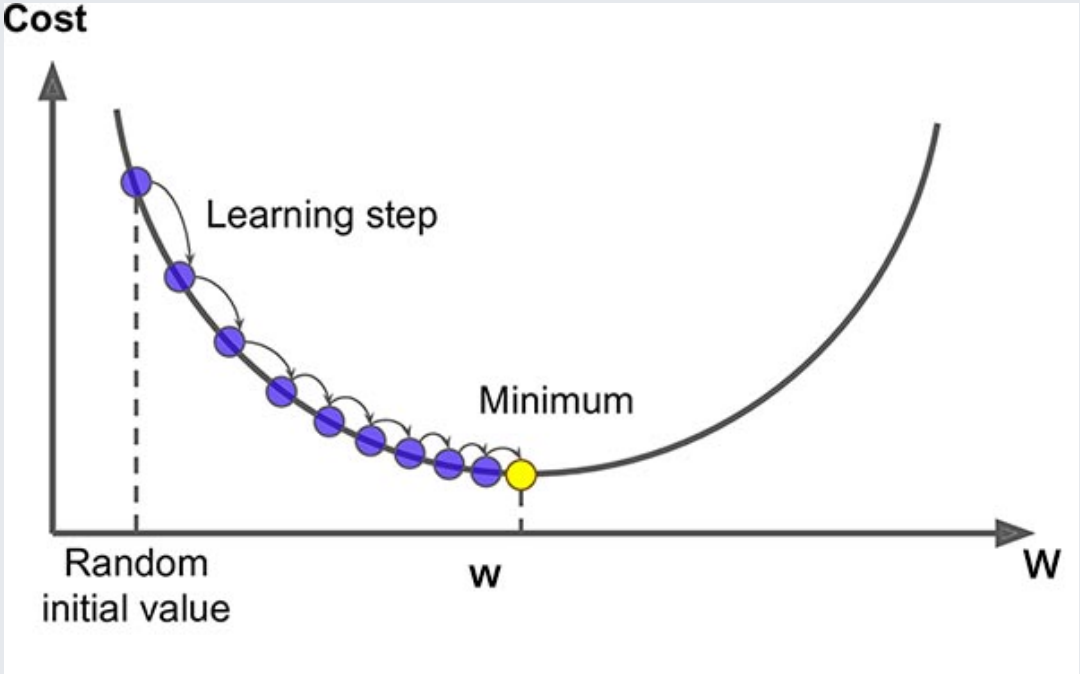


Figure: We start from a random point, and go down-hill by updating our parameter $w \leftarrow w - \alpha \nabla f(x)$ until convergence.

Gradient Descent

Several ways to compute $\nabla f(w, x)$:

- Take one data at a time and compute the gradient: SGD, faster but may be unstable.
- Take a lot of data at a time: (Batch) Gradient Descent, slower but more stable.

PRAM Model

Purpose: Analyze the complexity of an algorithm using distributed systems.
Idea: Draw a direct acylic graph, each node representing a calculation (such as multiplication/addition). We put

- all nodes that can be done in parallel in one level,
- node that depend on previous nodes as parent of these nodes

The number of all nodes is the **work** (how much computation is needed) and the depth of the DAG is the **depth** (how parallel can it be).

4. Labs

Lab 4: Spark and Drill

a. Common deploy methods for Spark?
   - Standalone Deploy Mode
   - Apache Mesos
   - Hadoop YARN
   - Kubernetes

Lab 4: Spark and Drill

b. Recall how you run your programs with spark.

- How to use `spark-submit`?
  Example:

```
./bin/spark-submit \
  --class <java_class> \ # if using java
  --master <?> \
  --deploy-mode <?> \
  --executor-memory <?> \
  --num-executors <?> \
  --files <related_files> \
  /path/to/the/executive/file \
  <args>
```

What do the options mean? How to use each option?

Lab 4: Spark and Drill

- When running on YARN, 2 modes exist:
  - client : runs the Driver on the client which submits the spark job. The driver runs in the client process, and the application master is only used for requesting resources from YARN.
  - cluster: runs the Driver on a slave node. The Spark driver runs inside an application master process which is managed by YARN on the cluster, and the client can go away after initiating the application.
- What is the difference of `map()` function and `flatmap()` function?
  - `map()` function produces one output for one input value, whereas `flatMap()` function produces an arbitrary no of values as output (i.e. zero or more than zero) for each input value.

  Why `flatmap()` is useful?

Lab 6: Big cluster

a. Docker:
Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called “containers”.
   - Image
   - Container

b. Hadoop node status: How to check node status? What happens when a node is not connected as expected?

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
3195	0	0	3195	0	0 B	72 GB	0 B	0	18	0	3	0	0	0	0

What do the metrics mean?

Lab 6: Big cluster

c. Drill:
Where to submit a query in a cluster? How to check the node status in drill?

```
apache drill> SELECT * FROM sys.drillbits;
+-----------------+-----------+--------------+-----------+-----------+---------+---------+--------+
|    hostname     | user_port | control_port | data_port | http_port | current | version | state  |
+-----------------+-----------+--------------+-----------+-----------+---------+---------+--------+
| hadoop-master   | 31010     | 31011        | 31012     | 8047      | false   | 1.18.0  | ONLINE |
| hadoop-worker-1 | 31010     | 31011        | 31012     | 8047      | true    | 1.18.0  | ONLINE |
+-----------------+-----------+--------------+-----------+-----------+---------+---------+--------+
2 rows selected (4.515 seconds)
```

Thank you and good luck with your final!

References

1. Jing Liu. "Ve472, Methods and Tools for Big Data" (lecture 3, 4, 7). In: *Canvas* (May. 2020).
2. Manuel Charlemagne. "Ve472, Methods and Tools for Big Data (c4)". In: *Canvas* (Jul. 2021).
3. Manuel Charlemagne. "Vv417, Linear Algebra". In: *Canvas* (Dec. 2020).