Today

- SVD + applications
- least squares
- maybe - perceptron alg.

Projects - google form

- presenting at Microsoft?

Credit for Figures:
- Roughgarden & Valiant
- Leskovec, Rajaraman, Ullman slides
- Emily Fox (6446)
- Kilian Weinberger (CS4780 Cornell)
PCA & SVD

have data set \( x_1, \ldots, x_m \) \( x \in \mathbb{R}^n \)

\[
X = \begin{pmatrix}
    x_1 \\
    \vdots \\
    x_m
\end{pmatrix}
\]

PCA: Fix \( k \). find orthonormal vectors \( v_1^*, \ldots, v_k^* \)

\[
\text{s.t. } x_i = \sum_{j=1}^k (x_i, v_j^*) v_j^*
\]

Find \( v_i \) to min

\[
\frac{1}{m} \sum_{i=1}^m \left( \sum_{j \neq i} (x_i, v_j^*) \right)^2
\]

\( \equiv \) max \( \frac{1}{m^2} \sum_{i=1}^m \sum_{j \neq i} (x_i, v_j^*)^2 \) variance.

\( v_i \), principal eigenvector of matrix \( X^T X \)

\[
X^T X = Q D Q^T
\]

\( \text{orthogonal.} \)

\( Q = \begin{pmatrix}
    q_1 \\
    \vdots \\
    q_k
\end{pmatrix} \)

\( \text{symmetric matrix} \)

\( D = \begin{pmatrix}
    \lambda_1 \\
    \vdots \\
    \lambda_k
\end{pmatrix} \)

\( \text{pos semi-definite. \ all eigenvalues non-negative} \)

\[
X^T X (q_i) = \lambda_i q_i
\]

Singular value decomposition (SVD)

\( \Rightarrow \) tells us best way to approximate

our matrix with a "low rank" matrix
Can we reconstruct missing entries?

Suppose assume that this matrix is rank 1, every row is a multiple of every other row.

\[
\begin{bmatrix}
7 & ? & ? \\
? & 8 & ? \\
? & 12 & 6 \\
? & ? & 2 \\
21 & 6 & ? \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
7 & 2 & 1 \\
8 & 12 & 6 \\
14 & 2 & 3 \\
21 & 6 & 3 \\
\end{bmatrix}
\]
Rank 0 matrix all 0.

Rank 1

\[ A = uu^T = \begin{bmatrix} u_1 u_2 \cdots u_m \end{bmatrix} \begin{bmatrix} u_1^T & u_2^T & \cdots & u_m^T \end{bmatrix} = \begin{bmatrix} v_1 u & v_2 u & \cdots & v_n u \end{bmatrix} \]

Rank 2

\[ A = uu^T + wz^T = \begin{bmatrix} u_1 u_2 \cdots u_m \end{bmatrix} \begin{bmatrix} u_1^T + w_1 z_1^T & u_2^T + w_2 z_1^T & \cdots & u_m^T + w_m z_1^T \\ \vdots \\ u_1^T + w_1 z_m^T & u_2^T + w_2 z_m^T & \cdots & u_m^T + w_m z_m^T \end{bmatrix} = \begin{bmatrix} u w \end{bmatrix} \begin{bmatrix} v^T \\ z^T \end{bmatrix} \]

Figure 1: Any matrix \( A \) of rank \( k \) can be decomposed into a long and skinny matrix times a short and long one.
Suppose we want the "best" rank $k$ approximation to $A$. Each singular value in $S$ has an associated left singular vector in $U$ and right singular vector in $V$. The singular values are called the singular values $S_1, S_2, \ldots, S_{\min(m,n)} \geq 0$ and can be computed in $\text{min}(O(m^2 n), O(n^2 m))$ time.
This low-rank approx is optimal in the sense that any matrix $A$ $(m \times n)$ and rank target $k \geq 1$ and any other rank $k$ matrix $B$ $(m \times n)$

\[
\|A - A_k\|_F^2 \leq \|A - B\|_F^2
\]

\[
\Sigma_{ij} (a_{ij} - b_{ij})^2 \leq \Sigma_{ij} (a_{ij} - b_{ij})^2
\]

Frobenius
**Application 1**: Denoising.

Suppose $A$ is a rank $k$ matrix.

$$ C = A + \mathcal{N} $$

noise matrix, each entry of $\mathcal{N}$ is independent $\mathcal{N}(0, \sigma^2)$

Then claim is if variance of noise sufficiently small, then

$$ \| C_k - A \|_F^2 \stackrel{\text{small w.h.p.}}{\sim} \sum_{i=k+1}^{n} s_i^2 $$

where $A = \sum_{i=1}^{k} s_i u_i v_i^T$ in $C$, $s_j \ll s_1, \ldots, s_k$, $j > k$.
**Theorem:** A \( m \times n \) matrix \( \mathbf{A} \) of independent random variables whose variances are bounded by \( \sigma^2 \).

If \( \mathbf{A} = E(\hat{\mathbf{A}}) \) is rank \( k \), then w.h.p.

\[
\| \mathbf{A} - \hat{\mathbf{A}} \|_F^2 = O(k \sigma^2 (m+n))
\]

\[
\hat{\mathbf{A}} = E(A) + \hat{\mathbf{A}} - E(\hat{\mathbf{A}})
\]

 deviation from mean 0 noise

\[
E(\hat{\mathbf{A}}) = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}
\]

\[
E(x_i) = \begin{pmatrix} 1 & 3 \\ 0 & 3 \end{pmatrix}
\]

\[
= o(1)
\]

---

**Collaborative Filtering**

**Recommendations**

movies

\[
\begin{bmatrix}
3 & 2 & 4 & 5 \\
1 & 3 & 4
\end{bmatrix}
\rightarrow
\begin{bmatrix}
3 & ? & 3 \\
2 & ? & 4 \\
? & 3 & 4
\end{bmatrix}
\]

\( \mathbf{R} \) ground truth

assumption: \( \mathbf{R} \) is rank \( k \)

---

**Movie:**

Honey, I Shrunk

**Reviewer:**

1 1 1

---

**Ground Truth:**

Movies

\[
\begin{bmatrix}
3 & 2 & 4 & 5 \\
1 & 3 & 4
\end{bmatrix}
\]
Define 
\[ \hat{R} = \begin{cases} \frac{R_{ij}}{P_{ij}} & \text{if entry } (i,j) \text{ is present} \\ 0 & \text{o.w.} \end{cases} \]

\[ E(\hat{R}_{ij}) = R_{ij} \frac{R_{ij}}{P_{ij}} + (1-P_{ij})0 \]

\[ = R_{ij} \]

\[ \Rightarrow \hat{R}_k \text{ is very close to } R. \]
Assume $P$ itself is low rank.

Given $R$?

Construct matrix $\hat{P}$ whose $(i,j)$ entry is 1

see rarg there and 0

o.w.

$$E(\hat{P}) = P$$

$$\Rightarrow \hat{P}_k$$ very close to $P$
Linear regression.

How much is my house worth?

I want to list my house for sale.

Data

- \((x_1 = \text{sq.ft.}, y_1 = 5)\)
- \((x_2 = \text{sq.ft.}, y_2 = 5)\)
- \((x_3 = \text{sq.ft.}, y_3 = 5)\)
- \((x_4 = \text{sq.ft.}, y_4 = 5)\)
- \((x_5 = \text{sq.ft.}, y_5 = 5)\)
- \(\vdots\)

\(f(\text{square footage}) \rightarrow \text{price}\)

Model –
How we assume the world works

Essentially, all models are wrong, but some are useful.”
George Box, 1987.

\((x_i; y_i)\) pairs.

Find best choice for \(w_0, w_1\).
Many possible inputs

- Square feet
- # bathrooms
- # bedrooms
- Lot size
- Year built
- ...

"Cost" of using a given line

\[
\text{RSS}(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2
\]
RSS for multiple regression
Lofgren

Pairs.

Linear regression

(x_i, y_i) pairs.

PCA

\[ \mathbf{X} \in \mathbb{R}^{n \times m}, \quad k \in \mathbb{N} \]

\[ k_{(m \times n)} \in \mathbb{R}^{k \times \min(m, n)} \]
Supervised learning & Perceptron Algorithm

labelled data: use that to come up with a way to give answers on data that haven't seen. $(\tilde{x}_i, \tilde{y}_i)$

binary classification. label for each pt $\tilde{y}_i \in \{1, -1\}$

Assumption: data set is linearly separable.

$\exists \mathbf{w}, b \text{ s.t. }$

\[ \tilde{y}_i = 1 \implies \mathbf{w}^T \tilde{x}_i + b > 0 \]

\[ \tilde{y}_i = -1 \implies \mathbf{w}^T \tilde{x}_i + b < 0 \]

Objective: find one

Simplify life by getting rid of additive cost $b$

\[
\begin{pmatrix}
\vdots \\
\mathbf{x}_n \\
\vdots 
\end{pmatrix}
\rightarrow
\begin{pmatrix}
\mathbf{w} \\
\end{pmatrix}
\]
Initialize $\vec{w} = \vec{0}$
while TRUE do
  $m = 0$
  for $(x_i, y_i) \in D$ do
    if $y_i (\vec{w}^T \cdot x_i) < 0$ then
      $\vec{w} \leftarrow \vec{w} + y_i x_i$
      $m \leftarrow m + 1$
    end if
  end for
  if $m = 0$ then
    break
  end if
end while

// Initialize $\vec{w}$. $\vec{w} = \vec{0}$ misclassifies everything.
// Keep looping
// Count the number of misclassifications, $m$
// Loop over each (data, label) pair in the dataset, $D$
// If the pair $(\vec{x}_i, y_i)$ is misclassified
// Update the weight vector $\vec{w}$
// Counter the number of misclassification

// If the most recent $\vec{w}$ gave 0 misclassifications
// Break out of the while-loop
// Otherwise, keep looping!

Illustration of a Perceptron update. (Left) The hyperplane defined by $\vec{w}_1$ misclassifies one red (-1) and one blue (+1) point. (Middle) The red point $\vec{x}_1$ is chosen and used for an update. Because its label is -1 we need to subtract $\vec{x}_1$ from $\vec{w}_1$. (Right) The updated hyperplane $\vec{w}_{1+1} = \vec{w}_1 - \vec{x}_1$ separates the two classes and the Perceptron algorithm has converged.

$\exists \vec{w} \ni y_i \left( \text{sign}(\vec{x}_i^T \vec{w})) > 0 \right.$

$log ||\vec{w}||=1$
to simplify

$||\vec{x}_i|| \leq 1$
A unit circle is shown with a vector $w$ within the circle. The expression $\theta^*$ is called the margin.

$$\theta^* = \min_{(x, y)} |x^T w^*|$$

Theorem: $m \leq \frac{1}{\theta^*}$

Look at 2 quantities:

1. $w^* \cdot w$ can't be going up
2. $w^* \cdot w$ increasing

When we make a mistake:

1. $y(w^* \cdot x) \leq 0$
2. $y(w^* \cdot x) > 0$

Let mistake increase by at most 1:

$$y(w^* \cdot (w + yx)) = w^* \cdot w + y(w^* \cdot x) \leq 0$$

0 < $m \theta^* \leq \frac{T^* w^*}{w^* \cdot w}$

$$\leq \frac{||w|| \cdot ||w^*||}{||w^*||}$$

$$\leq \frac{||w||}{||w||} = 1$$

$m \theta^* \leq \sqrt{m}$

$m^2 \theta^* \leq m$

$m \leq \frac{1}{\theta^*}$
Bad example for perceptron alg.