Perceptron and Linear Regression

Professor Ameet Talwalkar
Outline

1 Administration

2 Review – Generative vs Discriminative

3 Review – Multiclass classification

4 Perceptron

5 Linear regression
Homeworks

- Homework 2: due now
- Homework 3 available online
  - Due on Monday, 2/13 (two days before the midterm)
Outline

1. Administration
2. Review – Generative vs Discriminative
3. Review – Multiclass classification
4. Perceptron
5. Linear regression
Generative vs Discriminative

**Discriminative**

- Requires only specifying a model for the conditional distribution \( p(y|x) \), and thus, maximizes the *conditional* likelihood
  \[
  \sum_n \log p(y_n|x_n).
  \]
- Models that try to learn mappings directly from feature space to the labels are also discriminative, e.g., perceptron, SVMs (covered later)
Generative vs Discriminative

**Discriminative**

- Requires only specifying a model for the conditional distribution $p(y|x)$, and thus, maximizes the *conditional* likelihood $\sum_n \log p(y_n|x_n)$.
- Models that try to learn mappings directly from feature space to the labels are also discriminative, e.g., perceptron, SVMs (covered later).

**Generative**

- Aims to model the joint probability $p(x, y)$ and thus maximize the *joint* likelihood $\sum_n \log p(x_n, y_n)$.
- The generative models we cover do so by modeling $p(x|y)$ and $p(y)$.
Generative approach

Model joint distribution of \( x = (\text{height, weight}), y = \text{sex} \)

our data

<table>
<thead>
<tr>
<th>Sex</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6’</td>
<td>175</td>
</tr>
<tr>
<td>0</td>
<td>5’2”</td>
<td>120</td>
</tr>
<tr>
<td>1</td>
<td>5’6”</td>
<td>140</td>
</tr>
<tr>
<td>1</td>
<td>6’2”</td>
<td>240</td>
</tr>
<tr>
<td>0</td>
<td>5.7”</td>
<td>130</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Intuition: we will model how heights vary (according to a Gaussian) in each sub-population (male and female).
Model of the joint distribution (1D)

\[ p(x, y) = p(y)p(x|y) \]

\[
= \begin{cases} 
  p_0 \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}} & \text{if } y = 0 \\
  p_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} & \text{if } y = 1
\end{cases}
\]

\[ p_0 + p_1 = 1 \text{ are prior probabilities, and} \]
\[ p(x|y) \text{ is a class conditional distribution} \]
Model of the joint distribution (1D)

\[ p(x, y) = p(y)p(x|y) \]

\[ = \begin{cases} 
  p_0 \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}} & \text{if } y = 0 \\
  p_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} & \text{if } y = 1 
\end{cases} \]

\[ p_0 + p_1 = 1 \] are prior probabilities, and \( p(x|y) \) is a class conditional distribution.

What are the parameters to learn?
QDA Parameter estimation

**Log Likelihood of training data** \( \mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N} \) with \( y_n \in \{0, 1\} \)

\[
\log P(\mathcal{D}) = \sum_{n} \log p(x_n, y_n)
\]

\[
= \sum_{n : y_n = 0} \log \left( p_0 \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x_n - \mu_0)^2}{2\sigma_0^2}} \right)
\]

\[
+ \sum_{n : y_n = 1} \log \left( p_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_n - \mu_1)^2}{2\sigma_1^2}} \right)
\]
QDA Parameter estimation

Log Likelihood of training data \( \mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N} \) with \( y_n \in \{0, 1\} \)

\[
\log P(\mathcal{D}) = \sum_{n} \log p(x_n, y_n)
\]

\[
= \sum_{n: y_n = 0} \log \left( p_0 \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x_n-\mu_0)^2}{2\sigma_0^2}} \right) + \sum_{n: y_n = 1} \log \left( p_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_n-\mu_1)^2}{2\sigma_1^2}} \right)
\]

Max log likelihood \((p_0^*, p_1^*, \mu_0^*, \mu_1^*, \sigma_0^*, \sigma_1^*) = \arg \max \log P(\mathcal{D})\)
Log Likelihood of training data $D = \{(x_n, y_n)\}_{n=1}^N$ with $y_n \in \{0, 1\}$

$$\log P(D) = \sum_n \log p(x_n, y_n)$$

$$= \sum_{n: y_n = 0} \log \left( p_0 \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x_n - \mu_0)^2}{2\sigma_0^2}} \right)$$

$$+ \sum_{n: y_n = 1} \log \left( p_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_n - \mu_1)^2}{2\sigma_1^2}} \right)$$

Max log likelihood $(p_0^*, p_1^*, \mu_0^*, \mu_1^*, \sigma_0^*, \sigma_1^*) = \arg \max \log P(D)$

Max likelihood ($D = 2$) $(p_0^*, p_1^*, \mu_0^*, \mu_1^*, \Sigma_0^*, \Sigma_1^*) = \arg \max \log P(D)$
Decision boundary

Decision based on comparing conditional probabilities

\[ p(y = 1|x) \geq p(y = 0|x) \]

which is equivalent to

\[ p(x|y = 1)p(y = 1) \geq p(x|y = 0)p(y = 0) \]
Decision boundary

Decision based on comparing conditional probabilities

\[ p(y = 1 | x) \geq p(y = 0 | x) \]

which is equivalent to

\[ p(x | y = 1)p(y = 1) \geq p(x | y = 0)p(y = 0) \]

Namely,

\[ -\frac{(x - \mu_1)^2}{2\sigma_1^2} - \log \sqrt{2\pi\sigma_1} + \log p_1 \geq -\frac{(x - \mu_0)^2}{2\sigma_0^2} - \log \sqrt{2\pi\sigma_0} + \log p_0 \]
Decision boundary

Decision based on comparing conditional probabilities

\[ p(y = 1 | x) \geq p(y = 0 | x) \]

which is equivalent to

\[ p(x | y = 1) p(y = 1) \geq p(x | y = 0) p(y = 0) \]

Namely,

\[- \left( \frac{x - \mu_1}{2\sigma_1} \right)^2 - \log \sqrt{2\pi \sigma_1} + \log p_1 \geq - \left( \frac{x - \mu_0}{2\sigma_0} \right)^2 - \log \sqrt{2\pi \sigma_0} + \log p_0 \]

\[ \Rightarrow ax^2 + bx + c \geq 0 \quad \leftarrow \text{the QDA decision boundary not } \textit{linear!} \]
QDA vs LDA vs NB

Max likelihood ($D = 2$) $(p_0^*, p_1^*, \mu_0^*, \mu_1^*, \Sigma_0^*, \Sigma_1^*) = \arg \max \log P(D)$
QDA vs LDA vs NB

Max likelihood \((D = 2)\) \( (p^*_0, p^*_1, \mu^*_0, \mu^*_1, \Sigma^*_0, \Sigma^*_1) = \arg \max \log P(D) \)

- QDA: Allows distinct, arbitrary covariance matrices for each class
- LDA: Requires the same arbitrary covariance matrix across classes
- GNB: Allows for distinct covariance matrices across each class, but these covariance matrices must be diagonal
- GNB in HW2 Problem 1: Requires the same diagonal covariance matrix across classes
There is no fixed rule

- It depends on how well your modeling assumption fits the data
- When data follows the generative assumption, generative models will likely yield a model that better fits the data
- But, discriminative models are less sensitive to incorrect modelling assumptions (and often require less parameters to train)
Outline

1. Administration

2. Review – Generative vs Discriminative

3. Review – Multiclass classification
   - Use binary classifiers as building blocks
   - Multinomial logistic regression

4. Perceptron

5. Linear regression
Setup

**Predict multiple classes/outcomes:** $C_1, C_2, \ldots, C_K$

- Weather prediction: sunny, cloudy, raining, etc
- Optical character recognition: 10 digits + 26 characters (lower and upper cases) + special characters, etc

**Studied methods**

- Nearest neighbor classifier
- Naive Bayes
- Gaussian discriminant analysis
- Logistic regression
From multiclass to binary classification

“one versus the rest”

- Train a binary classifier for each class $C_k$:
  1. Relabel training data with label $C_k$, into POSITIVE (or ‘1’)
  2. Relabel all the rest data into NEGATIVE (or ‘0’)

- Train $K$ total binary classifiers

- Aggregate predictions at test time
From multiclass to binary classification

“one versus the rest”

- Train a binary classifier or each class $C_k$:
  1. Relabel training data with label $C_k$, into **positive** (or ‘1’)
  2. Relabel all the rest data into **negative** (or ‘0’)
- Train $K$ total binary classifiers
- Aggregate predictions at test time

“one versus one”

- Train a binary classifier for each pair of classes $C_k$ and $C_k'$
  1. Relabel training data with label $C_k$, into **positive** (or ‘1’)
  2. Relabel training data with label $C_k'$ into **negative** (or ‘0’)
  3. **Disregard** all other data
- Train $K(K - 1)/2$ total binary classifiers
- Tally ‘votes’ from each classifier at test time
Contrast these two approaches

Pros of each approach

- **one versus the rest**: only needs to train $K$ classifiers.
  - Makes a *big* difference if you have a lot of *classes* to go through.

- **one versus one**: only needs to train a smaller subset of data (only those labeled with those two classes would be involved).
  - Makes a *big* difference if you have a lot of *data* to go through.
Contrast these two approaches

Pros of each approach

- **one versus the rest**: only needs to train $K$ classifiers.
  - Makes a *big* difference if you have a lot of *classes* to go through.
- **one versus one**: only needs to train a smaller subset of data (only those labeled with those two classes would be involved).
  - Makes a *big* difference if you have a lot of *data* to go through.

Bad about both of them

*Combining classifiers’ outputs seem to be a bit tricky.*

Is there a more natural approach to generalize logistic regression?
First try

Can we just define the following conditional model for each class?

\[ p(y = C_k | x) = \sigma [w_k^T x] \]

This would not work because:

\[ \sum_k p(y = C_k | x) = \sum_k \sigma [w_k^T x] \neq 1 \]

as each summand can be any number (independently) between 0 and 1.

But we are close! We can learn the \( K \) linear models jointly to ensure this property holds!
First try

Can we just define the following conditional model for each class?

\[ p(y = C_k | x) = \sigma[w_k^T x] \]

This would \textit{not} work because:

\[ \sum_k p(y = C_k | x) = \sum_k \sigma[w_k^T x] \neq 1 \]

as each summand can be any number (independently) between 0 and 1.

\textit{But we are close!} We can learn the \( K \) linear models jointly to ensure this property holds!
Definition of multinomial logistic regression

Model

For each class $C_k$, we have a parameter vector $w_k$ and model the posterior probability as

$$p(C_k | x) = \frac{e^{w_k^T x}}{\sum_{k'} e^{w_{k'}^T x}} \quad \leftarrow \text{This is called softmax function}$$
Definition of multinomial logistic regression

Model

For each class $C_k$, we have a parameter vector $w_k$ and model the posterior probability as

$$p(C_k|x) = \frac{e^{w_k^T x}}{\sum_{k'} e^{w_{k'}^T x}}$$

This is called softmax function

Decision boundary: assign $x$ with the label that is the maximum of posterior

$$\arg \max_k P(C_k|x) \rightarrow \arg \max_k w_k^T x$$
Definition of multinomial logistic regression

Model

For each class \( C_k \), we have a parameter vector \( \mathbf{w}_k \) and model the posterior probability as

\[
p(C_k|\mathbf{x}) = \frac{e^{\mathbf{w}_k^T \mathbf{x}}}{\sum_{k'} e^{\mathbf{w}_{k'}^T \mathbf{x}}} \quad \leftarrow \quad \text{This is called softmax function}
\]

Decision boundary: assign \( \mathbf{x} \) with the label that is the maximum of posterior

\[
\arg \max_k P(C_k|\mathbf{x}) \rightarrow \arg \max_k \mathbf{w}_k^T \mathbf{x}
\]

Properties:
- Preserves relative ordering of ‘scores’ \( \mathbf{w}_k^T \mathbf{x} \) for each class
- Maps scores to values between 0 and 1 that also sum to 1
- Reduces to binary logistic regression when \( K = 2 \)
Parameter estimation

**Discriminative approach:** maximize conditional likelihood

\[
\log P(D) = \sum_n \log P(y_n|x_n)
\]
Parameter estimation

**Discriminative approach:** maximize conditional likelihood

\[
\log P(D) = \sum_n \log P(y_n|\mathbf{x}_n)
\]

We will change \(y_n\) to \(y_n = [y_{n1} \ y_{n2} \ \cdots \ y_{nK}]^T\), a \(K\)-dimensional vector using 1-of-\(K\) encoding, e.g., if \(y_n = 2\), then, \(y_n = [0 \ 1 \ 0 \ 0 \ \cdots \ 0]^T\).
Parameter estimation

**Discriminative approach:** maximize conditional likelihood

\[
\log P(D) = \sum_{n} \log P(y_n|x_n)
\]

We will change \( y_n \) to \( y_n = [y_{n1} \ y_{n2} \ \cdots \ y_{nK}]^T \), a \( K \)-dimensional vector using 1-of-K encoding, e.g., if \( y_n = 2 \), then, \( y_n = [0 \ 1 \ 0 \ 0 \ \cdots \ 0]^T \).

\[
\Rightarrow \sum_n \log P(y_n|x_n) = \sum_n \log \left( \prod_{k=1}^{K} P(C_k|x_n)^{y_{nk}} \right) = \sum_n \sum_k y_{nk} \log P(C_k|x_n)
\]

Optimization requires numerical procedures, analogous to those used for binary logistic regression.
Outline

1 Administration

2 Review – Generative vs Discriminative

3 Review – Multiclass classification

4 Perceptron
   - Intuition
   - Algorithm

5 Linear regression
Consider a linear model for binary classification

\[ w^T x \]

We use this model to distinguish between two classes \([-1, +1]\).

One goal

\[ \varepsilon = \sum_n \mathbb{I}[y_n \neq \text{sign}(w^T x_n)] \]

i.e., to minimize errors on the training dataset.
Hard, but easy if we have only one training example

How can we change $\mathbf{w}$ such that

$$y_n = \text{sign}(\mathbf{w}^T \mathbf{x}_n)$$

Two cases

- If $y_n = \text{sign}(\mathbf{w}^T \mathbf{x}_n)$, do nothing.
- If $y_n \neq \text{sign}(\mathbf{w}^T \mathbf{x}_n)$,

$$\mathbf{w}^{\text{NEW}} \leftarrow \mathbf{w}^{\text{OLD}} + y_n \mathbf{x}_n$$
Why would it work?

If \( y_n \neq \text{sign}(\mathbf{w}^T \mathbf{x}_n) \), then

\[
y_n (\mathbf{w}^T \mathbf{x}_n) < 0
\]
Why would it work?

If \( y_n \neq \text{sign}(w^T x_n) \), then

\[
y_n(w^T x_n) < 0
\]

What would happen if we change to new \( w^{\text{NEW}} = w + y_n x_n \)?

\[
y_n[(w + y_n x_n)^T x_n] = y_n w^T x_n + y_n^2 x_n^T x_n
\]
Why would it work?

If \( y_n \neq \text{sign}(w^T x_n) \), then

\[
y_n(w^T x_n) < 0
\]

What would happen if we change to new \( w^\text{NEW} = w + y_n x_n \)?

\[
y_n[(w + y_n x_n)^T x_n] = y_n w^T x_n + y_n^2 x_n^T x_n
\]

We are adding a positive number, so it is possible that

\[
y_n(w^\text{NEW}^T x_n) > 0
\]

i.e., we are more likely to classify correctly
Perceptron

Iteratively solving one case at a time

- **REPEAT**
- Pick a data point \( x_n \) (can be a fixed order of the training instances)
- Make a prediction \( y = \text{sign}(w^T x_n) \) using the *current* \( w \)
- If \( y = y_n \), do nothing. Else,

  \[
  w \leftarrow w + y_n x_n
  \]

- **UNTIL** converged.
Perceptron

Iteratively solving one case at a time

- REPEAT
- Pick a data point \( x_n \) (can be a fixed order of the training instances)
- Make a prediction \( y = \text{sign}(w^T x_n) \) using the current \( w \)
- If \( y = y_n \), do nothing. Else,

\[
w \leftarrow w + y_n x_n
\]

- UNTIL converged.

Properties

- This is an online algorithm.
- If the training data is linearly separable, the algorithm stops in a finite number of steps.
- The parameter vector is always a linear combination of training instances (requires initialization of \( w_0 = 0 \))
Convergence under linear separability

Let $x_1, \ldots, x_T \in \mathbb{R}^D$ be a sequence of $T$ points processed until convergence.
Convergence under linear separability

- Let \( x_1, \ldots, x_T \in \mathbb{R}^D \) be a sequence of \( T \) points processed until convergence.
- Assume \( \|x_t\| \leq r \) for all \( t \in [1, T] \), for some \( r > 0 \).
Convergence under linear separability

- Let $x_1, \ldots, x_T \in \mathbb{R}^D$ be a sequence of $T$ points processed until convergence.
- Assume $\|x_t\| \leq r$ for all $t \in [1, T]$, for some $r > 0$.
- Assume that there exist $\rho > 0$ and $v \in \mathbb{R}^D$ s.t. for all $t \in [1, T]$,
  \[ \rho \leq \frac{y_t (v \cdot x_t)}{\|v\|} \]
**Convergence under linear separability**

- Let \( \mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^D \) be a sequence of \( T \) points processed until convergence
- Assume \( \|\mathbf{x}_t\| \leq r \) for all \( t \in [1, T] \), for some \( r > 0 \)
- Assume that there exist \( \rho > 0 \) and \( \mathbf{v} \in \mathbb{R}^D \) s.t. for all \( t \in [1, T] \),
  \[
  \rho \leq \frac{y_t (\mathbf{v} \cdot \mathbf{x}_t)}{\|\mathbf{v}\|}
  \]

Then, the number of updates \( M \) made by the Perceptron algorithm when processing \( \mathbf{x}_1, \ldots, \mathbf{x}_T \) is bounded by

\[
M \leq \frac{r^2}{\rho^2}
\]
Recall that $\rho \leq \frac{y_t(v \cdot x_t)}{\|v\|}$, $w_{t+1} = w_t + y_t x_t$, and $w_0 = 0$

Let $I$ be the subset of the $T$ rounds with an update, i.e., $|I| = M$

$$M \rho \leq \frac{v \cdot \sum_{t \in I} y_t x_t}{\|v\|}$$
Recall that $\rho \leq \frac{y_t (v \cdot x_t)}{||v||}$, $w_{t+1} = w_t + y_t x_t$, and $w_0 = 0$

Let $I$ be the subset of the $T$ rounds with an update, i.e., $|I| = M$

$$M \rho \leq \frac{v \cdot \sum_{t \in I} y_t x_t}{||v||} \leq \left\| \sum_{t \in I} y_t x_t \right\|$$

(Cauchy-Schwarz inequality)
Recall that $\rho \leq \frac{y_t(v \cdot x_t)}{\|v\|}$, $w_{t+1} = w_t + y_t x_t$, and $w_0 = 0$

Let $I$ be the subset of the $T$ rounds with an update, i.e., $|I| = M$

\[
M \rho \leq \frac{v \cdot \sum_{t \in I} y_t x_t}{\|v\|} \leq \left\| \sum_{t \in I} y_t x_t \right\| \quad \text{(Cauchy-Schwarz inequality)}
\]

\[
= \left\| \sum_{t \in I} (w_{t+1} - w_t) \right\| \quad \text{(definition of updates)}
\]
- Recall that $\rho \leq \frac{y_t (v \cdot x_t)}{\|v\|}$, $w_{t+1} = w_t + y_t x_t$, and $w_0 = 0$
- Let $I$ be the subset of the $T$ rounds with an update, i.e., $|I| = M$

$$M \rho \leq \frac{v \cdot \sum_{t \in I} y_t x_t}{\|v\|} \leq \left\| \sum_{t \in I} y_t x_t \right\|$$

(Cauchy-Schwarz inequality)

$$= \left\| \sum_{t \in I} (w_{t+1} - w_t) \right\|$$

(definition of updates)

$$= \left\| w_{T+1} \right\|$$

(telescoping sum, $w_0 = 0$)
Recall that $\rho \leq \frac{y_t (v \cdot x_t)}{\|v\|}$, $w_{t+1} = w_t + y_t x_t$, and $w_0 = 0$.

Let $I$ be the subset of the $T$ rounds with an update, i.e., $|I| = M$.

$$M \rho \leq \frac{v \cdot \sum_{t \in I} y_t x_t}{\|v\|} \leq \left\| \sum_{t \in I} y_t x_t \right\|$$

(Cauchy-Schwarz inequality)

$$= \left\| \sum_{t \in I} (w_{t+1} - w_t) \right\|$$

(definition of updates)

$$= \|w_{T+1}\|$$

(telescoping sum, $w_0 = 0$)

$$= \sqrt{\sum_{t \in I} \|w_{t+1}\|^2 - \|w_t\|^2}$$

(telescoping sum, $w_0 = 0$)
Recall that $\rho \leq \frac{y_t(v \cdot x_t)}{\|v\|}$, $w_{t+1} = w_t + y_t x_t$, and $w_0 = 0$

Let $I$ be the subset of the $T$ rounds with an update, i.e., $|I| = M$

$$M \rho \leq \frac{v \cdot \sum_{t \in I} y_t x_t}{\|v\|} \leq \left\| \sum_{t \in I} y_t x_t \right\|$$

(Cauchy-Schwarz inequality)

$$= \left\| \sum_{t \in I} (w_{t+1} - w_t) \right\|$$

(definition of updates)

$$= \left\| w_{T+1} \right\|$$

(telescoping sum, $w_0 = 0$)

$$= \sqrt{\sum_{t \in I} \|w_{t+1}\|^2 - \|w_t\|^2}$$

(telescoping sum, $w_0 = 0$)

$$= \sqrt{\sum_{t \in I} \|w_t + y_t x_t\|^2 - \|w_t\|^2}$$

(definition of updates)
Recall that \( \rho \leq \frac{y_t(v \cdot x_t)}{\|v\|} \), \( w_{t+1} = w_t + y_t x_t \), and \( w_0 = 0 \).

Let \( I \) be the subset of the \( T \) rounds with an update, i.e., \( |I| = M \).

\[
M \rho \leq \frac{v \cdot \sum_{t \in I} y_t x_t}{\|v\|} \leq \left\| \sum_{t \in I} y_t x_t \right\| \quad \text{(Cauchy-Schwarz inequality)}
\]

\[
= \left\| \sum_{t \in I} (w_{t+1} - w_t) \right\| \quad \text{(definition of updates)}
\]

\[
= \|w_{T+1}\| \quad \text{(telescoping sum, \( w_0 = 0 \))}
\]

\[
= \sqrt{\sum_{t \in I} \|w_{t+1}\|^2 - \|w_t\|^2} \quad \text{(telescoping sum, \( w_0 = 0 \))}
\]

\[
= \sqrt{\sum_{t \in I} \|w_t + y_t x_t\|^2 - \|w_t\|^2} \quad \text{(definition of updates)}
\]

\[
= \sqrt{\sum_{t \in I} 2y_t w_t \cdot x_t + \|x_t\|^2} \leq 0
\]

\( \leq 0 \)
Recall that $\rho \leq \frac{y_t (v \cdot x_t)}{\|v\|}$, $w_{t+1} = w_t + y_t x_t$, and $w_0 = 0$

Let $I$ be the subset of the $T$ rounds with an update, i.e., $|I| = M$

$$M \rho \leq \frac{v \cdot \sum_{t \in I} y_t x_t}{\|v\|} \leq \left\| \sum_{t \in I} y_t x_t \right\|$$

(Cauchy-Schwarz inequality)

$$= \left\| \sum_{t \in I} (w_{t+1} - w_t) \right\|$$

(definition of updates)

$$= \left\| w_{T+1} \right\|$$

(telescoping sum, $w_0 = 0$)

$$= \sqrt{\sum_{t \in I} \left( \left\| w_{t+1} \right\|^2 - \left\| w_t \right\|^2 \right)}$$

(telescoping sum, $w_0 = 0$)

$$= \sqrt{\sum_{t \in I} \left( \left\| w_t + y_t x_t \right\|^2 - \left\| w_t \right\|^2 \right)}$$

(definition of updates)

$$= \sqrt{\sum_{t \in I} \left( \begin{array}{c} 2 y_t w_t \cdot x_t + \|x_t\|^2 \\ \leq 0 \end{array} \right)}$$

$$\leq \sqrt{\sum_{t \in I} \|x_t\|^2}$$
Recall that $\rho \leq \frac{y_t (v \cdot x_t)}{\|v\|}$, $w_{t+1} = w_t + y_t x_t$, and $w_0 = 0$

Let $I$ be the subset of the $T$ rounds with an update, i.e., $|I| = M$

$$M \rho \leq \frac{v \cdot \sum_{t \in I} y_t x_t}{\|v\|} \leq \left\| \sum_{t \in I} y_t x_t \right\|$$

(Cauchy-Schwarz inequality)

$$= \left\| \sum_{t \in I} (w_{t+1} - w_t) \right\|$$

(definition of updates)

$$= \left\| w_{T+1} \right\|$$

(telescoping sum, $w_0 = 0$)

$$= \sqrt{\sum_{t \in I} \|w_{t+1}\|^2 - \|w_t\|^2}$$

(telescoping sum, $w_0 = 0$)

$$= \sqrt{\sum_{t \in I} \|w_t + y_t x_t\|^2 - \|w_t\|^2}$$

(definition of updates)

$$\leq \sqrt{\sum_{t \in I} 2 y_t w_t \cdot x_t + \|x_t\|^2} \leq 0$$

$$\leq \sqrt{\sum_{t \in I} \|x_t\|^2} \leq \sqrt{Mr^2}$$
Recall that $\rho \leq \frac{y_t(v \cdot x_t)}{\|v\|}$, $w_{t+1} = w_t + y_t x_t$, and $w_0 = 0$

Let $I$ be the subset of the $T$ rounds with an update, i.e., $|I| = M$

$$M \rho \leq \frac{v \cdot \sum_{t \in I} y_t x_t}{\|v\|} \leq \left\| \sum_{t \in I} y_t x_t \right\|$$  \hspace{1cm} (Cauchy-Schwarz inequality)

$$= \left\| \sum_{t \in I} (w_{t+1} - w_t) \right\|$$  \hspace{1cm} (definition of updates)

$$= \left\| w_{T+1} \right\|$$  \hspace{1cm} (telescoping sum, $w_0 = 0$)

$$= \sqrt{\sum_{t \in I} \|w_{t+1}\|^2 - \|w_t\|^2}$$  \hspace{1cm} (telescoping sum, $w_0 = 0$)

$$= \sqrt{\sum_{t \in I} \|w_t + y_t x_t\|^2 - \|w_t\|^2}$$  \hspace{1cm} (definition of updates)

$$= \sqrt{\sum_{t \in I} \left[ 2y_t w_t \cdot x_t + \|x_t\|^2 \right]} \leq 0$$

$$\leq \sqrt{\sum_{t \in I} \|x_t\|^2} \leq \sqrt{Mr^2}$$  \hspace{1cm} (Therefore, $M \rho \leq \sqrt{Mr^2} \rightarrow M \leq \frac{r^2}{\rho^2}$)
Outline

1. Administration
2. Review – Generative vs Discriminative
3. Review – Multiclass classification
4. Perceptron
5. Linear regression
   - Motivation
   - Algorithm
   - Univariante solution
   - Probabilistic interpretation
Regression

Predicting a continuous outcome variable

- Predicting shoe size from height, weight and gender
- Predicting a company’s future stock price using its profit and other financial info
- Predicting annual rainfall based on local flora/fauna
- Predicting song year from audio features
Regression

Predicting a continuous outcome variable
- Predicting shoe size from height, weight and gender
- Predicting a company’s future stock price using its profit and other financial info
- Predicting annual rainfall based on local flora / fauna
- Predicting song year from audio features

Key difference from classification
Regression

Predicting a continuous outcome variable

- Predicting shoe size from height, weight and gender
- Predicting a company’s future stock price using its profit and other financial info
- Predicting annual rainfall based on local flora / fauna
- Predicting song year from audio features

Key difference from classification

- We can measure ‘closeness’ of prediction and labels, leading to different ways to evaluate prediction errors.
  - Predicting shoe size: better to be off by one size than by 5 sizes
  - Predicting song year: better to be off by one year than by 20 years
- This will lead to different learning models and algorithms
Ex: predicting the sale price of a house

Retrieve historical sales records
(This will be our training data)
Features used to predict

Five unit apartment complex within 2 blocks of USC campus, Gate #6. Great for students (most student units have parents as guarantor). Most USC students live off campus, so housing units like this are always fully leased. Situated on a gated, corner lot, and across from an elementary school, this complex was recently renovated, and has in-unit laundry hook ups, wall -unit AC, and 12 parking spaces. It is within a CPS (Department of Public Safety) and Campus Cruiser policed area. This is a great income-generating property, not to be missed.

Property Details for 3620 South BUDLONG, Los Angeles, CA 90007

Details provided by i-Tech MLS and may not match the public record. Learn More

Interior Features

- Kitchen Information:
  - Remodeled
  - Oven, Range

- Laundry Information:
  - Inside Laundry

- Heating & Cooling:
  - Wall Cooling Units

Multi-Unit Information

- Community Features:
  - Units in Complex (Total): 5
  - Multi-Family Information:
    - # of Leased: 5
    - # of Bedrooms: 1
    - Owner Pays Water
    - Tenant Pays Electric, Tenant Pays Gas

- Unit 1 Information:
  - # of Beds: 2
  - # of Baths: 1
  - Unfurnished
  - Monthly Rent: $1,700

- Unit 2 Information:
  - # of Beds: 3
  - # of Baths: 1
  - Unfurnished
  - Monthly Rent: $2,250

- Unit 3 Information:
  - # of Beds: 3
  - Unfurnished

- Unit 4 Information:
  - # of Beds: 3
  - # of Baths: 1
  - Unfurnished

Property / Lot Details

- Property Features:
  - Automatic Gate, Card/Code Access

- Lot Information:
  - Lot Size (Bd. Fl.): 9,649
  - Lot Size (Area): 9,649

- Lot Information Source:
  - Public Records

- Property Information:
  - Updated: Remodeled
  - Square Footage Source: Public Records

- Building Information:
  - Total Floors: 2

Parking / Garage, Exterior Features, Utilities & Financing

- Parking Information:
  - # of Parking Spaces (Total): 12
  - Parking Spots

- Garage

Location Details, Misc. Information & Listing Information

- Location Information:
  - Cross Streets: W. 36th PI

- Expense Information:
  - Operating: $37,884

- Financial Information:
  - Capitalization Rate (%): 6.25

- Actual Annual Gross Rent: $126,331

- Gross Rent Multiplier: 11.29

- Listing Information:
  - Listing Terms: Cash, Cash To Existing Loan, Buyer Financing Cash
Correlation between square footage and sale price

Note: colors here do NOT represent different labels as in classification
Roughly linear relationship

Sale price ≈ price per sqft × square footage + fixed expense
Roughly linear relationship

Sale price $\approx price\_per\_sqft \times square\_footage + fixed\_expense$
How to learn the unknown parameters?

**training data** (past sales record)

<table>
<thead>
<tr>
<th>sqft</th>
<th>sale price</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>800K</td>
</tr>
<tr>
<td>2100</td>
<td>907K</td>
</tr>
<tr>
<td>1100</td>
<td>312K</td>
</tr>
<tr>
<td>5500</td>
<td>2,600K</td>
</tr>
<tr>
<td>(\cdot)</td>
<td>(\cdot)</td>
</tr>
</tbody>
</table>
Reduce prediction error

How to measure errors?

- The classification error (hit or miss) is not appropriate for continuous outcomes.
- How should we evaluate quality of a prediction?
Reduce prediction error

**How to measure errors?**

- The classification error (*hit* or *miss*) is not appropriate for continuous outcomes.
- How should we evaluate quality of a prediction?
  - *absolute* difference: $|\text{prediction} - \text{sale price}|$
  - *squared* difference: $(\text{prediction} - \text{sale price})^2$ [differentiable]

<table>
<thead>
<tr>
<th>sqft</th>
<th>sale price</th>
<th>prediction</th>
<th>error</th>
<th>squared error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>810K</td>
<td>720K</td>
<td>90K</td>
<td>8100</td>
</tr>
<tr>
<td>2100</td>
<td>907K</td>
<td>800K</td>
<td>107K</td>
<td>107^2</td>
</tr>
<tr>
<td>1100</td>
<td>312K</td>
<td>350K</td>
<td>38K</td>
<td>38^2</td>
</tr>
<tr>
<td>5500</td>
<td>2,600K</td>
<td>2,600K</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Minimize squared errors

**Our model**

Sale price = price\_per\_sqft × square\_footage + fixed\_expense + unexplainable\_stuff

**Training data**

<table>
<thead>
<tr>
<th>sqft</th>
<th>sale price</th>
<th>prediction</th>
<th>error</th>
<th>squared error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>810K</td>
<td>720K</td>
<td>90K</td>
<td>8100</td>
</tr>
<tr>
<td>2100</td>
<td>907K</td>
<td>800K</td>
<td>107K</td>
<td>107^2</td>
</tr>
<tr>
<td>1100</td>
<td>312K</td>
<td>350K</td>
<td>38K</td>
<td>38^2</td>
</tr>
<tr>
<td>5500</td>
<td>2,600K</td>
<td>2,600K</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>(8100 + 107^2 + 38^2 + 0 + \cdots)</td>
</tr>
</tbody>
</table>
Minimize squared errors

Our model
Sale price = price_per_sqft × square_footage + fixed_expense + unexplainable_stuff

Training data

<table>
<thead>
<tr>
<th>sqft</th>
<th>sale price</th>
<th>prediction</th>
<th>error</th>
<th>squared error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>810K</td>
<td>720K</td>
<td>90K</td>
<td>8100</td>
</tr>
<tr>
<td>2100</td>
<td>907K</td>
<td>800K</td>
<td>107K</td>
<td>107²</td>
</tr>
<tr>
<td>1100</td>
<td>312K</td>
<td>350K</td>
<td>38K</td>
<td>38²</td>
</tr>
<tr>
<td>5500</td>
<td>2,600K</td>
<td>2,600K</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>8100 + 107² + 38² + 0 + ...</td>
</tr>
</tbody>
</table>

Aim
Adjust price_per_sqft and fixed_expense such that the sum of the squared error is minimized — i.e., the residual/remaining unexplainable_stuff is minimized.
Linear regression

Setup

- **Input**: \( x \in \mathbb{R}^D \) (covariates, predictors, features, etc)
- **Output**: \( y \in \mathbb{R} \) (responses, targets, outcomes, outputs, etc)

Model:

\[
\begin{align*}
\text{f}: x & \to y \\
\text{f}(x) & = w_0 + \sum_{d} w_d x_d \\
& = w_0 + w^T x \\
& = \begin{bmatrix} w_1 & w_2 & \cdots & w_D \end{bmatrix}^T \\
\end{align*}
\]

▶ \( w_0 \) is called bias

We also sometimes call \( \tilde{w} = \begin{bmatrix} w_0 & w_1 & w_2 & \cdots & w_D \end{bmatrix}^T \) parameters too

Training data:

\( \mathcal{D} = \{(x_n, y_n), n = 1, 2, \ldots, N\} \)
Linear regression

Setup

- **Input**: \( x \in \mathbb{R}^D \) (covariates, predictors, features, etc)
- **Output**: \( y \in \mathbb{R} \) (responses, targets, outcomes, outputs, etc)
- **Model**: \( f : x \rightarrow y \), with \( f(x) = w_0 + \sum_d w_d x_d = w_0 + \mathbf{w}^T \mathbf{x} \)
  - \( \mathbf{w} = [w_1 \ w_2 \cdots \ w_D]^T \): weights, parameters, or parameter vector
  - \( w_0 \) is called **bias**
  - We also sometimes call \( \tilde{\mathbf{w}} = [w_0 \ w_1 \ w_2 \cdots \ w_D]^T \) parameters too
Linear regression

Setup

- **Input:** $x \in \mathbb{R}^D$ (covariates, predictors, features, etc)
- **Output:** $y \in \mathbb{R}$ (responses, targets, outcomes, outputs, etc)
- **Model:** $f : x \to y$, with $f(x) = w_0 + \sum_d w_d x_d = w_0 + w^T x$
  - $w = [w_1 \; w_2 \; \cdots \; w_D]^T$: weights, parameters, or parameter vector
  - $w_0$ is called bias
  - We also sometimes call $\tilde{w} = [w_0 \; w_1 \; w_2 \; \cdots \; w_D]^T$ parameters too
- **Training data:** $\mathcal{D} = \{(x_n, y_n), n = 1, 2, \ldots, N\}$
How do we learn parameters?

Minimize prediction error on training data

- Use squared difference to measure error
- Residual sum of squares

\[ RSS(\tilde{w}) = \sum_n [y_n - f(x_n)]^2 = \sum_n [y_n - (w_0 + \sum_d w_d x_{nd})]^2 \]
A simple case: \( x \) is just one-dimensional \((D=1)\)

**Residual sum of squares**

\[
RSS(\tilde{w}) = \sum_{n}[y_n - f(x_n)]^2 = \sum_{n}[y_n - (w_0 + w_1 x_n)]^2
\]
A simple case: \( x \) is just one-dimensional (\( D=1 \))

Residual sum of squares

\[
RSS(\tilde{w}) = \sum_n [y_n - f(x_n)]^2 = \sum_n [y_n - (w_0 + w_1 x_n)]^2
\]

Identify stationary points by taking derivative with respect to parameters and setting to zero

\[
\frac{\partial RSS(\tilde{w})}{\partial w_0} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] = 0
\]

\[
\frac{\partial RSS(\tilde{w})}{\partial w_1} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] x_n = 0
\]
\[
\frac{\partial \text{RSS}(\tilde{w})}{\partial w_0} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] = 0
\]

\[
\frac{\partial \text{RSS}(\tilde{w})}{\partial w_1} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] x_n = 0
\]

Simplify these expressions to get “Normal Equations”
\[
\frac{\partial \text{RSS}(\tilde{w})}{\partial w_0} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] = 0
\]

\[
\frac{\partial \text{RSS}(\tilde{w})}{\partial w_1} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] x_n = 0
\]

Simplify these expressions to get “Normal Equations”

\[
\sum y_n = N w_0 + w_1 \sum x_n
\]

\[
\sum x_n y_n = w_0 \sum x_n + w_1 \sum x_n^2
\]
\[
\frac{\partial RSS(\tilde{w})}{\partial w_0} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] = 0
\]

\[
\frac{\partial RSS(\tilde{w})}{\partial w_1} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] x_n = 0
\]

Simplify these expressions to get “Normal Equations”

\[
\sum y_n = N w_0 + w_1 \sum x_n
\]

\[
\sum x_n y_n = w_0 \sum x_n + w_1 \sum x_n^2
\]

We have two equations and two unknowns! Do some algebra to get:

\[
w_1 = \frac{\sum(x_n - \bar{x})(y_n - \bar{y})}{\sum(x_i - \bar{x})^2}
\]

and

\[
w_0 = \bar{y} - w_1 \bar{x}
\]

where \( \bar{x} = \frac{1}{n} \sum_n x_n \) and \( \bar{y} = \frac{1}{n} \sum_n y_n \).
Why is minimizing RSS sensible?

**Probabilistic interpretation**

- Noisy observation model

\[ Y = w_0 + w_1 X + \eta \]

where \( \eta \sim N(0, \sigma^2) \) is a Gaussian random variable
Why is minimizing RSS sensible?

Probabilistic interpretation

- Noisy observation model
  \[ Y = w_0 + w_1 X + \eta \]
  where \( \eta \sim N(0, \sigma^2) \) is a Gaussian random variable
- Conditional likelihood of one training sample:
  \[
  p(y_n | x_n) = N(w_0 + w_1 x_n, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(y_n - (w_0 + w_1 x_n))^2}{2\sigma^2}}
  \]
Probabilistic interpretation (cont’d)

Log-likelihood of the training data $\mathcal{D}$ (assuming i.i.d)

$$\log P(\mathcal{D}) = \log \prod_{n=1}^{N} p(y_n | x_n) = \sum_{n} \log p(y_n | x_n)$$
Probabilistic interpretation (cont’d)

Log-likelihood of the training data $\mathcal{D}$ (assuming i.i.d)

$$\log P(\mathcal{D}) = \log \prod_{n=1}^{N} p(y_n|x_n) = \sum_n \log p(y_n|x_n)$$

$$= \sum_n \left\{ -\frac{[y_n - (w_0 + w_1x_n)]^2}{2\sigma^2} - \log \sqrt{2\pi\sigma} \right\}$$
Probabilistic interpretation (cont’d)

Log-likelihood of the training data $\mathcal{D}$ (assuming i.i.d)

$$
\log P(\mathcal{D}) = \log \prod_{n=1}^{N} p(y_n | x_n) = \sum_{n} \log p(y_n | x_n)
$$

$$
= \sum_{n} \left\{ -\frac{(y_n - (w_0 + w_1 x_n))^2}{2\sigma^2} - \log \sqrt{2\pi\sigma} \right\}
$$

$$
= -\frac{1}{2\sigma^2} \sum_{n} [y_n - (w_0 + w_1 x_n)]^2 - \frac{N}{2} \log \sigma^2 - N \log \sqrt{2\pi}
$$
Probabilistic interpretation (cont’d)

Log-likelihood of the training data $\mathcal{D}$ (assuming i.i.d)

$$
\log P(\mathcal{D}) = \log \prod_{n=1}^{N} p(y_n|x_n) = \sum_{n} \log p(y_n|x_n)
$$

$$
= \sum_{n} \left\{ -\frac{[y_n - (w_0 + w_1 x_n)]^2}{2\sigma^2} - \log \sqrt{2\pi\sigma} \right\}
$$

$$
= -\frac{1}{2\sigma^2} \sum_{n} [y_n - (w_0 + w_1 x_n)]^2 - \frac{N}{2} \log \sigma^2 - N \log \sqrt{2\pi}
$$

$$
= -\frac{1}{2} \left\{ \frac{1}{\sigma^2} \sum_{n} [y_n - (w_0 + w_1 x_n)]^2 + N \log \sigma^2 \right\} + \text{const}
$$

What is the relationship between minimizing RSS and maximizing the log-likelihood?
Maximum likelihood estimation

**Estimating $\sigma, w_0$ and $w_1$ can be done in two steps**

- Maximize over $w_0$ and $w_1$

\[
\max \log P(D) \iff \min \sum_n [y_n - (w_0 + w_1 x_n)]^2 \leftarrow \text{That is } \text{RSS}(\tilde{w})!
\]
Maximum likelihood estimation

**Estimating $\sigma$, $w_0$ and $w_1$ can be done in two steps**

- Maximize over $w_0$ and $w_1$

  $$\max \log P(D) \Leftrightarrow \min \sum_n [y_n - (w_0 + w_1 x_n)]^2 \leftarrow \text{That is } \text{RSS}(\tilde{w})!$$

- Maximize over $s = \sigma^2$

  $$\frac{\partial \log P(D)}{\partial s} = -\frac{1}{2} \left\{ -\frac{1}{s^2} \sum_n [y_n - (w_0 + w_1 x_n)]^2 + N \frac{1}{s} \right\} = 0$$
Maximum likelihood estimation

Estimating $\sigma$, $w_0$ and $w_1$ can be done in two steps

1. Maximize over $w_0$ and $w_1$

   $$\max \log P(D) \iff \min \sum_n [y_n - (w_0 + w_1 x_n)]^2 \leftarrow \text{That is RSS}(\tilde{w})!$$

2. Maximize over $s = \sigma^2$

   $$\frac{\partial \log P(D)}{\partial s} = -\frac{1}{2} \left\{ -\frac{1}{s^2} \sum_n [y_n - (w_0 + w_1 x_n)]^2 + N \frac{1}{s} \right\} = 0$$

   $$\rightarrow \sigma^*^2 = s^* = \frac{1}{N} \sum_n [y_n - (w_0 + w_1 x_n)]^2$$
How does this probabilistic interpretation help us?

- It gives a solid footing to our intuition: minimizing $\text{RSS}(\tilde{w})$ is a sensible thing based on reasonable modeling assumptions.
- Estimating $\sigma^*$ tells us how much noise there could be in our predictions. For example, it allows us to place confidence intervals around our predictions.