Perceptron and Linear Regresssion

Professor Ameet Talwalkar

Outline

Homeworks

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

Outline

Generative vs Discriminative

Discriminative

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

Generative

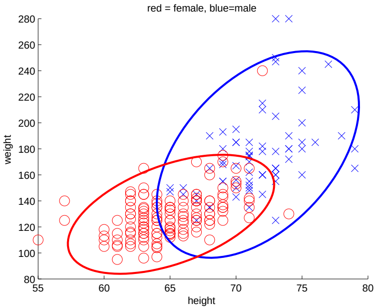
- Aims to model the joint probability $p(x, y)$ and thus maximize the *joint* likelihood $\sum_n \log p(\boldsymbol{x}_n, y_n)$.
- The generative models we cover do so by modeling $p(x|y)$ and $p(y)$

Generative approach

Model joint distribution of ($x =$ (height, weight), $y =$sex)

our data

Sex	Height	Weight
1	$6'$	175
0	$5'2"$	120
1	$5'6"$	140
1	$6'2"$	240
0	$5.7"$	130
...	...	...

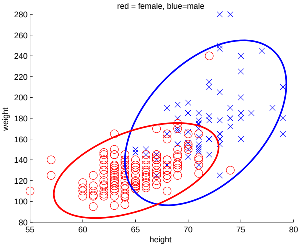


Intuition: we will model how heights vary (according to a Gaussian) in each sub-population (male and female).

Model of the joint distribution (1D)

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

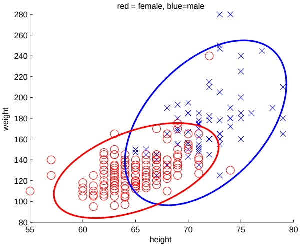
$p_0 + p_1 = 1$ are *prior* probabilities, and $p(x|y)$ is a *class conditional distribution*



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 $y_n \in \{0, 1\}$

$$\begin{aligned}\log P(\mathcal{D}) &= \sum_n \log p(x_n, y_n) \\ &= \sum_{n: y_n = 0} \log \left(p_0 \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x_n - \mu_0)^2}{2\sigma_0^2}} \right) \\ &+ \sum_{n: y_n = 1} \log \left(p_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_n - \mu_1)^2}{2\sigma_1^2}} \right)\end{aligned}$$

QDA Parameter estimation

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

$$
\begin{aligned}
\log P(\mathcal{D}) &= \sum_n \log p(x_n, y_n) \\
&= \sum_{n:y_n=0} \log \left(p_0 \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x_n-\mu_0)^2}{2\sigma_0^2}} \right) \\
&+ \sum_{n:y_n=1} \log \left(p_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_n-\mu_1)^2}{2\sigma_1^2}} \right)
\end{aligned}
$$

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

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 $(p_0^*, p_1^*, \mu_0^*, \mu_1^*, \sigma_0^*, \sigma_1^*) = \arg\max \log P(\mathcal{D})$

Max likelihood ($D = 2$) $(p_0^*, p_1^*, \boldsymbol{\mu}_0^*, \boldsymbol{\mu}_1^*, \boldsymbol{\Sigma}_0^*, \boldsymbol{\Sigma}_1^*) = \arg\max \log P(\mathcal{D})$

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,

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

$\Rightarrow ax^2 + bx + c \geq 0 \qquad \leftarrow$ the QDA decision boundary not *linear*!

QDA vs LDA vs NB

Max likelihood ($D = 2$) $(p_0^*, p_1^*, \boldsymbol{\mu}_0^*, \boldsymbol{\mu}_1^*, \boldsymbol{\Sigma}_0^*, \boldsymbol{\Sigma}_1^*) = \arg\max \log P(\mathcal{D})$

QDA vs LDA vs NB

Max likelihood ($D = 2$) $(p_0^*, p_1^*, \boldsymbol{\mu}_0^*, \boldsymbol{\mu}_1^*, \boldsymbol{\Sigma}_0^*, \boldsymbol{\Sigma}_1^*) = \arg\max \log P(\mathcal{D})$

- 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

Generative versus discriminative: which one to use?

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

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 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

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?

$$p(y = C_k|\boldsymbol{x}) = \sigma[\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{x}]$$

First try

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

$$p(y = C_k|\boldsymbol{x}) = \sigma[\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{x}]$$

This would *not* work because:

$$\sum_k p(y = C_k|\boldsymbol{x}) = \sum_k \sigma[\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{x}] \neq 1$$

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!

Definition of multinomial logistic regression

Model

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

$$p(C_k|\boldsymbol{x}) = \frac{e^{\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{x}}}{\sum_{k'} e^{\boldsymbol{w}_{k'}^{\mathrm{T}}\boldsymbol{x}}} \qquad \leftarrow \qquad \text{This is called } \textit{softmax} \text{ function}$$

Definition of multinomial logistic regression

Model

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

$$p(C_k|\boldsymbol{x}) = \frac{e^{\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{x}}}{\sum_{k'} e^{\boldsymbol{w}_{k'}^{\mathrm{T}}\boldsymbol{x}}} \qquad \leftarrow \qquad \text{This is called } \textit{softmax} \text{ function}$$

Decision boundary: assign $\boldsymbol{x}$ with the label that is the maximum of posterior

$$\arg\max_k P(C_k|\boldsymbol{x}) \rightarrow \arg\max_k \boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{x}$$

Definition of multinomial logistic regression

Model

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

$$p(C_k|\boldsymbol{x}) = \frac{e^{\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{x}}}{\sum_{k'} e^{\boldsymbol{w}_{k'}^{\mathrm{T}}\boldsymbol{x}}} \qquad \leftarrow \qquad \text{This is called } \textit{softmax} \text{ function}$$

Decision boundary: assign $\boldsymbol{x}$ with the label that is the maximum of posterior

$$\arg\max_k P(C_k|\boldsymbol{x}) \rightarrow \arg\max_k \boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{x}$$

Properties:

- Preserves relative ordering of 'scores' $\boldsymbol{w}_k^{\top}\boldsymbol{x}$ 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

$$\log P(\mathcal{D}) = \sum_n \log P(y_n | \boldsymbol{x}_n)$$

Parameter estimation

Discriminative approach: maximize conditional likelihood

$$\log P(\mathcal{D}) = \sum_n \log P(y_n|\boldsymbol{x}_n)$$

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

Parameter estimation

Discriminative approach: maximize conditional likelihood

$$\log P(\mathcal{D}) = \sum_n \log P(y_n | \boldsymbol{x}_n)$$

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

$$\Rightarrow \sum_n \log P(y_n | \boldsymbol{x}_n) = \sum_n \log \prod_{k=1}^{K} P(C_k | \boldsymbol{x}_n)^{y_{nk}} = \sum_n \sum_k y_{nk} \log P(C_k | \boldsymbol{x}_n)$$

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

Outline

Main idea

Consider a linear model for binary classification

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$$

We use this model to distinguish between two classes $\{-1, +1\}$.

One goal

$$\varepsilon = \sum_n \mathbb{I}[y_n \neq \mathsf{sign}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n)]$$

i.e., to minimize errors on the training dataset.

Hard, but easy if we have only one training example

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

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

Two cases

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

$$\boldsymbol{w}^{\text{NEW}} \leftarrow \boldsymbol{w}^{\text{OLD}} + y_n\boldsymbol{x}_n$$

Why would it work?

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

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n) < 0$$

Why would it work?

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

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n) < 0$$

What would happen if we change to new $\boldsymbol{w}^{\text{NEW}} = \boldsymbol{w} + y_n\boldsymbol{x}_n$?

$$y_n[(\boldsymbol{w} + y_n\boldsymbol{x}_n)^{\mathrm{T}}\boldsymbol{x}_n] = y_n\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n + y_n^2\boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{x}_n$$

Why would it work?

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

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n) < 0$$

What would happen if we change to new $\boldsymbol{w}^{\text{NEW}} = \boldsymbol{w} + y_n\boldsymbol{x}_n$?

$$y_n[(\boldsymbol{w} + y_n\boldsymbol{x}_n)^{\mathrm{T}}\boldsymbol{x}_n] = y_n\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n + y_n^2\boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{x}_n$$

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

$$y_n(\boldsymbol{w}^{\text{NEW}\,\mathrm{T}}\boldsymbol{x}_n) > 0$$

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

Perceptron

Iteratively solving one case at a time

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

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + y_n\boldsymbol{x}_n$$

- UNTIL converged.

Perceptron

Iteratively solving one case at a time

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

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + y_n \boldsymbol{x}_n$$

- 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 $\boldsymbol{w}_0 = 0$)

Convergence under linear separability

- Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T \in \mathbb{R}^D$ be a sequence of T points processed until convergence

Convergence under linear separability

- Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T \in \mathbb{R}^D$ be a sequence of T points processed until convergence
- Assume $\|\boldsymbol{x}_t\| \leq r$ for all $t \in [1, T]$, for some $r > 0$

Convergence under linear separability

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

$$\rho \leq \frac{y_t(\boldsymbol{v} \cdot \boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$$

Convergence under linear separability

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

$$\rho \leq \frac{y_t(\boldsymbol{v} \cdot \boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$$

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

$$M \leq r^2/\rho^2$$

- Recall that $\rho \leq \frac{y_t(\boldsymbol{v} \cdot \boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t \boldsymbol{x}_t$, and $\boldsymbol{w}_0 = 0$
- Let I be the subset of the T rounds with an update, i.e., $|I| = M$

$$M\rho \leq \frac{\boldsymbol{v} \cdot \sum_{t \in I} y_t \boldsymbol{x}_t}{\|\boldsymbol{v}\|}$$

- Recall that $\rho \leq \frac{y_t(\boldsymbol{v} \cdot \boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t \boldsymbol{x}_t$, and $\boldsymbol{w}_0 = 0$
- Let I be the subset of the T rounds with an update, i.e., $|I| = M$

$$M\rho \leq \frac{\boldsymbol{v} \cdot \sum_{t \in I} y_t \boldsymbol{x}_t}{\|\boldsymbol{v}\|} \leq \Big\| \sum_{t \in I} y_t \boldsymbol{x}_t \Big\| \qquad \text{(Cauchy-Schwarz inequality)}$$

- Recall that $\rho \leq \frac{y_t(\boldsymbol{v} \cdot \boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t \boldsymbol{x}_t$, and $\boldsymbol{w}_0 = 0$
- Let I be the subset of the T rounds with an update, i.e., $|I| = M$

$$M\rho \leq \frac{\boldsymbol{v} \cdot \sum_{t \in I} y_t \boldsymbol{x}_t}{\|\boldsymbol{v}\|} \leq \Big\| \sum_{t \in I} y_t \boldsymbol{x}_t \Big\| \qquad \text{(Cauchy-Schwarz inequality)}$$

$$= \Big\| \sum_{t \in I} (\boldsymbol{w}_{t+1} - \boldsymbol{w}_t) \Big\| \qquad \text{(definition of updates)}$$

- Recall that $\rho \leq \frac{y_t(\boldsymbol{v} \cdot \boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t \boldsymbol{x}_t$, and $\boldsymbol{w}_0 = 0$
- Let I be the subset of the T rounds with an update, i.e., $|I| = M$

$$
\begin{aligned}
M\rho &\leq \frac{\boldsymbol{v} \cdot \sum_{t \in I} y_t \boldsymbol{x}_t}{\|\boldsymbol{v}\|} \leq \Big\| \sum_{t \in I} y_t \boldsymbol{x}_t \Big\| && \text{(Cauchy-Schwarz inequality)} \\
&= \Big\| \sum_{t \in I} (\boldsymbol{w}_{t+1} - \boldsymbol{w}_t) \Big\| && \text{(definition of updates)} \\
&= \|\boldsymbol{w}_{T+1}\| && \text{(telescoping sum, } \boldsymbol{w}_0 = 0\text{)}
\end{aligned}
$$

- Recall that $\rho \leq \frac{y_t(\boldsymbol{v} \cdot \boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t \boldsymbol{x}_t$, and $\boldsymbol{w}_0 = 0$
- Let I be the subset of the T rounds with an update, i.e., $|I| = M$

$$M\rho \leq \frac{\boldsymbol{v} \cdot \sum_{t \in I} y_t \boldsymbol{x}_t}{\|\boldsymbol{v}\|} \leq \Big\| \sum_{t \in I} y_t \boldsymbol{x}_t \Big\| \qquad \text{(Cauchy-Schwarz inequality)}$$

$$= \Big\| \sum_{t \in I} (\boldsymbol{w}_{t+1} - \boldsymbol{w}_t) \Big\| \qquad \text{(definition of updates)}$$

$$= \|\boldsymbol{w}_{T+1}\| \qquad \text{(telescoping sum, } \boldsymbol{w}_0 = 0\text{)}$$

$$= \sqrt{\sum_{t \in I} \|\boldsymbol{w}_{t+1}\|^2 - \|\boldsymbol{w}_t\|^2} \qquad \text{(telescoping sum, } \boldsymbol{w}_0 = 0\text{)}$$

- Recall that $\rho \leq \frac{y_t(\boldsymbol{v} \cdot \boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t \boldsymbol{x}_t$, and $\boldsymbol{w}_0 = 0$
- Let I be the subset of the T rounds with an update, i.e., $|I| = M$

$$
\begin{aligned}
M\rho &\leq \frac{\boldsymbol{v} \cdot \sum_{t \in I} y_t \boldsymbol{x}_t}{\|\boldsymbol{v}\|} \leq \Big\| \sum_{t \in I} y_t \boldsymbol{x}_t \Big\| && \text{(Cauchy-Schwarz inequality)} \\
&= \Big\| \sum_{t \in I} (\boldsymbol{w}_{t+1} - \boldsymbol{w}_t) \Big\| && \text{(definition of updates)} \\
&= \|\boldsymbol{w}_{T+1}\| && \text{(telescoping sum, } \boldsymbol{w}_0 = 0) \\
&= \sqrt{\sum_{t \in I} \|\boldsymbol{w}_{t+1}\|^2 - \|\boldsymbol{w}_t\|^2} && \text{(telescoping sum, } \boldsymbol{w}_0 = 0) \\
&= \sqrt{\sum_{t \in I} \|\boldsymbol{w}_t + y_t \boldsymbol{x}_t\|^2 - \|\boldsymbol{w}_t\|^2} && \text{(definition of updates)}
\end{aligned}
$$

- Recall that $\rho \leq \frac{y_t(\boldsymbol{v} \cdot \boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t \boldsymbol{x}_t$, and $\boldsymbol{w}_0 = 0$
- Let I be the subset of the T rounds with an update, i.e., $|I| = M$

$$
\begin{aligned}
M\rho &\leq \frac{\boldsymbol{v} \cdot \sum_{t \in I} y_t \boldsymbol{x}_t}{\|\boldsymbol{v}\|} \leq \Big\| \sum_{t \in I} y_t \boldsymbol{x}_t \Big\| && \text{(Cauchy-Schwarz inequality)} \\
&= \Big\| \sum_{t \in I} (\boldsymbol{w}_{t+1} - \boldsymbol{w}_t) \Big\| && \text{(definition of updates)} \\
&= \|\boldsymbol{w}_{T+1}\| && \text{(telescoping sum, } \boldsymbol{w}_0 = 0) \\
&= \sqrt{\sum_{t \in I} \|\boldsymbol{w}_{t+1}\|^2 - \|\boldsymbol{w}_t\|^2} && \text{(telescoping sum, } \boldsymbol{w}_0 = 0) \\
&= \sqrt{\sum_{t \in I} \|\boldsymbol{w}_t + y_t \boldsymbol{x}_t\|^2 - \|\boldsymbol{w}_t\|^2} && \text{(definition of updates)} \\
&= \sqrt{\sum_{t \in I} 2 \underbrace{y_t \boldsymbol{w}_t \cdot \boldsymbol{x}_t}_{\leq 0} + \|\boldsymbol{x}_t\|^2}
\end{aligned}
$$

- Recall that $\rho \leq \frac{y_t(\boldsymbol{v} \cdot \boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t \boldsymbol{x}_t$, and $\boldsymbol{w}_0 = 0$
- Let I be the subset of the T rounds with an update, i.e., $|I| = M$

$$
\begin{aligned}
M\rho &\leq \frac{\boldsymbol{v} \cdot \sum_{t \in I} y_t \boldsymbol{x}_t}{\|\boldsymbol{v}\|} \leq \Big\| \sum_{t \in I} y_t \boldsymbol{x}_t \Big\| && \text{(Cauchy-Schwarz inequality)} \\
&= \Big\| \sum_{t \in I} (\boldsymbol{w}_{t+1} - \boldsymbol{w}_t) \Big\| && \text{(definition of updates)} \\
&= \|\boldsymbol{w}_{T+1}\| && \text{(telescoping sum, } \boldsymbol{w}_0 = 0) \\
&= \sqrt{\sum_{t \in I} \|\boldsymbol{w}_{t+1}\|^2 - \|\boldsymbol{w}_t\|^2} && \text{(telescoping sum, } \boldsymbol{w}_0 = 0) \\
&= \sqrt{\sum_{t \in I} \|\boldsymbol{w}_t + y_t \boldsymbol{x}_t\|^2 - \|\boldsymbol{w}_t\|^2} && \text{(definition of updates)} \\
&= \sqrt{\sum_{t \in I} 2 \underbrace{y_t \boldsymbol{w}_t \cdot \boldsymbol{x}_t}_{\leq 0} + \|\boldsymbol{x}_t\|^2} \\
&\leq \sqrt{\sum_{t \in I} \|\boldsymbol{x}_t\|^2}
\end{aligned}
$$

- Recall that $\rho \leq \frac{y_t(\boldsymbol{v} \cdot \boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t \boldsymbol{x}_t$, and $\boldsymbol{w}_0 = 0$
- Let I be the subset of the T rounds with an update, i.e., $|I| = M$

$$
\begin{aligned}
M\rho &\leq \frac{\boldsymbol{v} \cdot \sum_{t \in I} y_t \boldsymbol{x}_t}{\|\boldsymbol{v}\|} \leq \Big\| \sum_{t \in I} y_t \boldsymbol{x}_t \Big\| && \text{(Cauchy-Schwarz inequality)} \\
&= \Big\| \sum_{t \in I} (\boldsymbol{w}_{t+1} - \boldsymbol{w}_t) \Big\| && \text{(definition of updates)} \\
&= \|\boldsymbol{w}_{T+1}\| && \text{(telescoping sum, } \boldsymbol{w}_0 = 0\text{)} \\
&= \sqrt{\sum_{t \in I} \|\boldsymbol{w}_{t+1}\|^2 - \|\boldsymbol{w}_t\|^2} && \text{(telescoping sum, } \boldsymbol{w}_0 = 0\text{)} \\
&= \sqrt{\sum_{t \in I} \|\boldsymbol{w}_t + y_t \boldsymbol{x}_t\|^2 - \|\boldsymbol{w}_t\|^2} && \text{(definition of updates)} \\
&= \sqrt{\sum_{t \in I} 2 \underbrace{y_t \boldsymbol{w}_t \cdot \boldsymbol{x}_t}_{\leq 0} + \|\boldsymbol{x}_t\|^2} \\
&\leq \sqrt{\sum_{t \in I} \|\boldsymbol{x}_t\|^2} \leq \sqrt{M r^2}
\end{aligned}
$$

- Recall that $\rho \leq \frac{y_t(\boldsymbol{v}\cdot\boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t\boldsymbol{x}_t$, and $\boldsymbol{w}_0 = 0$
- Let I be the subset of the T rounds with an update, i.e., $|I| = M$

$$
\begin{aligned}
M\rho &\leq \frac{\boldsymbol{v}\cdot\sum_{t\in I} y_t\boldsymbol{x}_t}{\|\boldsymbol{v}\|} \leq \Big\|\sum_{t\in I} y_t\boldsymbol{x}_t\Big\| && \text{(Cauchy-Schwarz inequality)}\\
&= \Big\|\sum_{t\in I}(\boldsymbol{w}_{t+1} - \boldsymbol{w}_t)\Big\| && \text{(definition of updates)}\\
&= \|\boldsymbol{w}_{T+1}\| && \text{(telescoping sum, } \boldsymbol{w}_0 = 0)\\
&= \sqrt{\sum_{t\in I}\|\boldsymbol{w}_{t+1}\|^2 - \|\boldsymbol{w}_t\|^2} && \text{(telescoping sum, } \boldsymbol{w}_0 = 0)\\
&= \sqrt{\sum_{t\in I}\|\boldsymbol{w}_t + y_t\boldsymbol{x}_t\|^2 - \|\boldsymbol{w}_t\|^2} && \text{(definition of updates)}\\
&= \sqrt{\sum_{t\in I} 2\underbrace{y_t\boldsymbol{w}_t\cdot\boldsymbol{x}_t}_{\leq 0} + \|\boldsymbol{x}_t\|^2}\\
&\leq \sqrt{\sum_{t\in I}\|\boldsymbol{x}_t\|^2} \leq \sqrt{Mr^2} && \text{(Therefore, } M\rho \leq \sqrt{Mr^2} \to M \leq \frac{r^2}{\rho^2})
\end{aligned}
$$

Outline

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 flaura / 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 flaura / 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 flaura / fauna
- Predicting song year from audio features

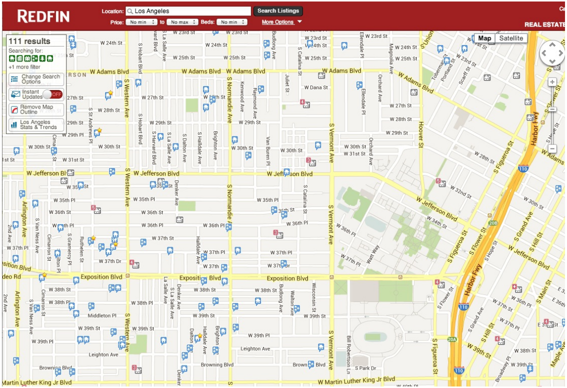
Key difference from classification

- 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

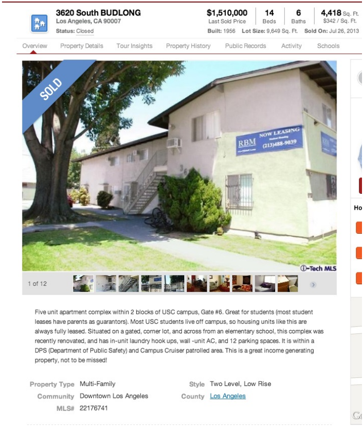
Ex: predicting the sale price of a house

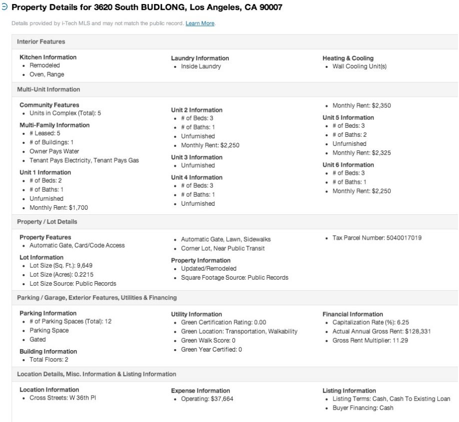
Retrieve historical sales records

(This will be our training data)



Features used to predict





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 Unit(s)

Multi-Unit Information

Community Features
- Units in Complex (Total): 5

Multi-Family Information
- # Leased: 5
- # of Buildings: 1
- Owner Pays Water
- Tenant Pays Electricity, 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
- Unfurnished

Unit 4 Information
- # of Beds: 3
- # of Baths: 1
- Unfurnished
- Monthly Rent: $2,350

Unit 5 Information
- # of Beds: 3
- # of Baths: 2
- Unfurnished
- Monthly Rent: $2,325

Unit 6 Information
- # of Beds: 3
- # of Baths: 1
- Monthly Rent: $2,250

Property / Lot Details

Property Features
- Automatic Gate, Card/Code Access
- Automatic Gate, Lawn, Sidewalks
- Corner Lot, Near Public Transit

Lot Information
- Lot Size (Sq. Ft.): 9,649
- Lot Size (Acres): 0.2215
- Lot Size Source: Public Records

Property Information
- Updated/Remodeled
- Square Footage Source: Public Records
- Tax Parcel Number: 5040017019

Parking / Garage, Exterior Features, Utilities & Financing

Parking Information
- # of Parking Spaces (Total): 12
- Parking Space
- Gated

Building Information
- Total Floors: 2

Utility Information
- Green Certification Rating: 0.00
- Green Location: Transportation, Walkability
- Green Walk Score: 0
- Green Year Certified: 0

Financial Information
- Capitalization Rate (%): 6.25
- Actual Annual Gross Rent: $128,331
- Gross Rent Multiplier: 11.29

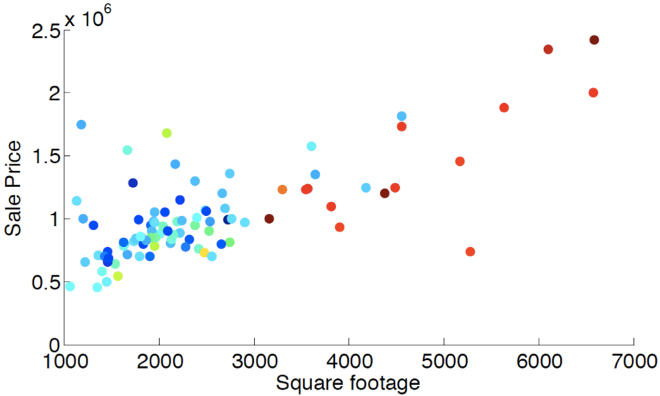
Location Details, Misc. Information & Listing Information

Location Information
- Cross Streets: W 36th Pl

Expense Information
- Operating: $37,664

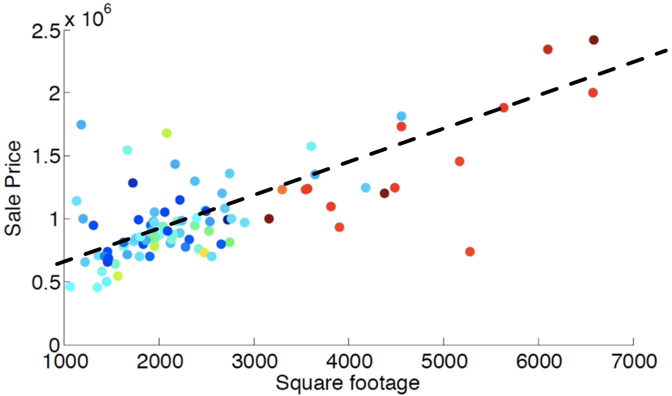
Listing Information
- Listing Terms: Cash, Cash To Existing Loan
- Buyer Financing: Cash

Correlation between square footage and sale price

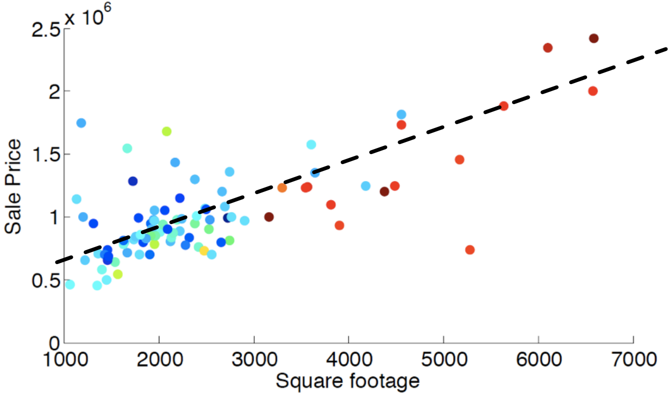


Note: colors here do NOT represent different labels as in classification

Roughly linear relationship



Roughly linear relationship



Sale price ≈ price_per_sqft × square_footage + fixed_expense

How to learn the unknown parameters?

training data (past sales record)

sqft	sale price
2000	800K
2100	907K
1100	312K
5500	2,600K
$\cdots$	$\cdots$

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: | prediction - sale price|
  - *squared* difference: (prediction - sale price)^2 [differentiable]

sqft	sale price	prediction	error	squared error
2000	810K	720K	90K	8100
2100	907K	800K	107K	107^2
1100	312K	350K	38K	38^2
5500	2,600K	2,600K	0	0
$\cdots$	$\cdots$			

Minimize squared errors

Our model

Sale price = price_per_sqft × square_footage + fixed_expense + unexplainable_stuff

Training data

sqft	sale price	prediction	error	squared error
2000	810K	720K	90K	8100
2100	907K	800K	107K	107^2
1100	312K	350K	38K	38^2
5500	2,600K	2,600K	0	0
$\cdots$	$\cdots$			
Total				$8100 + 107^2 + 38^2 + 0 + \cdots$

Minimize squared errors

Our model

Sale price = price_per_sqft × square_footage + fixed_expense + unexplainable_stuff

Training data

sqft	sale price	prediction	error	squared error
2000	810K	720K	90K	8100
2100	907K	800K	107K	107^2
1100	312K	350K	38K	38^2
5500	2,600K	2,600K	0	0
$\cdots$	$\cdots$			
Total				$8100 + 107^2 + 38^2 + 0 + \cdots$

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

Linear regression

Setup

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

Linear regression

Setup

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

How do we learn parameters?

Minimize prediction error on training data

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

$$RSS(\tilde{\boldsymbol{w}}) = \sum_n [y_n - f(\boldsymbol{x}_n)]^2 = \sum_n [y_n - (w_0 + \sum_d w_d x_{nd})]^2$$

A simple case: x is just one-dimensional ($D=1$)

Residual sum of squares

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

A simple case: x is just one-dimensional ($D=1$)

Residual sum of squares

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

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

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

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

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

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

Simplify these expressions to get "Normal Equations"

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

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

Simplify these expressions to get "Normal Equations"

$$\sum y_n = N w_0 + w_1 \sum x_n$$

$$\sum x_n y_n = w_0 \sum x_n + w_1 \sum x_n^2$$

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

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

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:

$$w_1 = \frac{\sum (x_n - \bar{x})(y_n - \bar{y})}{\sum (x_i - \bar{x})^2} \qquad \text{and} \qquad w_0 = \bar{y} - w_1 \bar{x}$$

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)

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

Probabilistic interpretation (cont'd)

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

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

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

Probabilistic interpretation (cont'd)

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

$$\begin{aligned}
\log P(\mathcal{D}) &= \log \prod_{n=1}^{\mathsf{N}} p(y_n|x_n) = \sum_n \log p(y_n|x_n) \\
&= \sum_n \left\{ -\frac{[y_n - (w_0 + w_1 x_n)]^2}{2\sigma^2} - \log \sqrt{2\pi}\sigma \right\} \\
&= -\frac{1}{2\sigma^2} \sum_n [y_n - (w_0 + w_1 x_n)]^2 - \frac{\mathsf{N}}{2} \log \sigma^2 - \mathsf{N} \log \sqrt{2\pi}
\end{aligned}$$

Probabilistic interpretation (cont'd)

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

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

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

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

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

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

Maximum likelihood estimation

Estimating σ, w_0 and w_1 can be done in two steps

- Maximize over w_0 and w_1

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

Maximum likelihood estimation

Estimating σ, w_0 and w_1 can be done in two steps

- Maximize over w_0 and w_1

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

- Maximize over $s = \sigma^2$

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

Maximum likelihood estimation

Estimating σ, w_0 and w_1 can be done in two steps

- Maximize over w_0 and w_1

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

- Maximize over $s = \sigma^2$

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

$$\rightarrow \sigma^{*2} = s^* = \frac{1}{\mathsf{N}}\sum_n [y_n - (w_0 + w_1 x_n)]^2$$

How does this probabilistic interpretation help us?

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