Web Mining and Recommender Systems

Supervised learning – Regression

Learning Goals

- Introduce the concept of **Supervised Learning**
- Understand the components (inputs and outputs) of supervised learning problems
- Introduce **linear regression**, one of the simplest forms of supervised learning

What is supervised learning?

Supervised learning is the process of trying to infer from **labeled data** the underlying function that produced the labels associated with the data

What is supervised learning?

Given **labeled training data** of the form

$$\{(\text{data}_1, \text{label}_1), \ldots, (\text{data}_n, \text{label}_n)\}$$

Infer the function

$$f(\text{data}) \overset{?}{\to} \text{labels}$$

Example

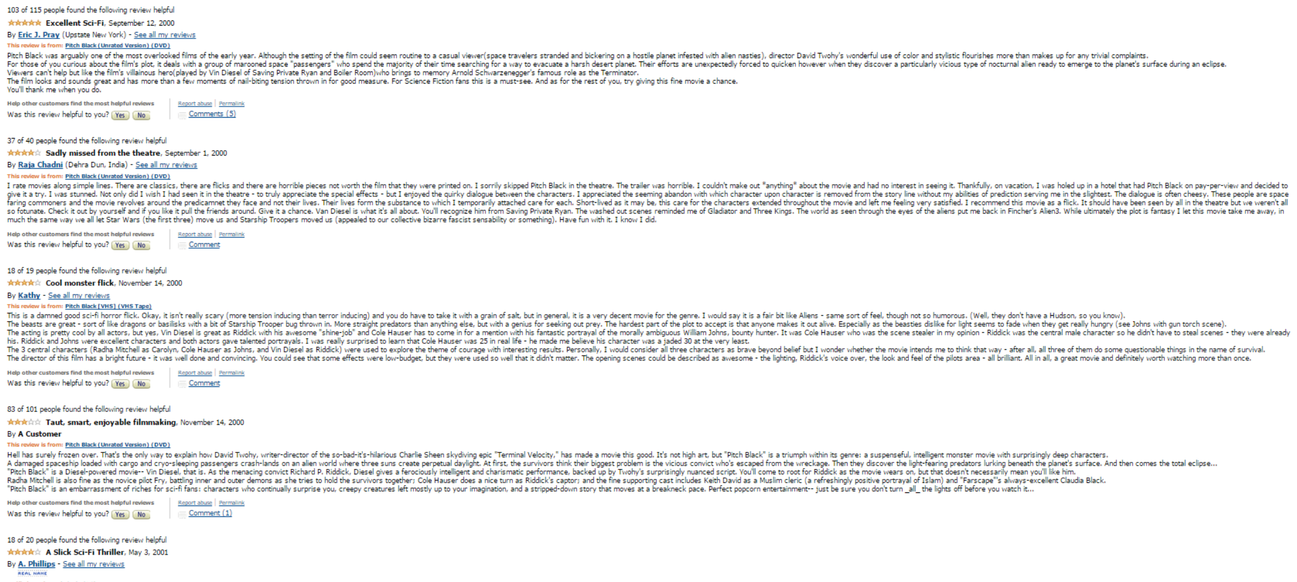
Suppose we want to build a movie recommender

e.g. which of these films will I rate highest?

Example

Q: What are the **labels?**

A: ratings that others have given to each movie, and that I have given to other movies



103 of 115 people found the following review helpful

Excellent Sci-Fi, September 12, 2000

By Eric J. Pray (Upstate New York) - See all my reviews

This review is from: Pitch Black (Unrated Version) (DVD)

Pitch Black was arguably one of the most overlooked films of the early year. Although the setting of the film could seem routine to a casual viewer(space travelers stranded and bickering on a hostile planet infested with alien nasties), director David Twohy's wonderful use of color and stylistic flourishes more than makes up for any trivial complaints.
For those of you curious about the film's plot, it deals with a group of marooned space "passengers" who spend the majority of their time searching for a way to evacuate a harsh desert planet. Their efforts are unexpectedly forced to quicken however when they discover a particularly vicious type of nocturnal alien ready to emerge to the planet's surface during an eclipse.
Viewers can't help but like the film's villainous hero(played by Vin Diesel of Saving Private Ryan and Boiler Room)who brings to memory Arnold Schwarzenegger's famous role as the Terminator.
The film looks and sounds great and has more than a few moments of nail-biting tension thrown in for good measure. For Science Fiction fans this is a must-see. And as for the rest of you, try giving this fine movie a chance.
You'll thank me when you do.

Help other customers find the most helpful reviews | Report abuse | Permalink

Was this review helpful to you? Yes No | Comments (5)

37 of 40 people found the following review helpful

Sadly missed from the theatre, September 1, 2000

By Raja Chadni (Dehra Dun, India) - See all my reviews

This review is from: Pitch Black (Unrated Version) (DVD)

I rate movies along simple lines. There are classics, there are flicks and there are horrible pieces not worth the film that they were printed on. I sorrily skipped Pitch Black in the theatre. The trailer was horrible. I couldn't make out *anything* about the movie and had no interest in seeing it. Thankfully, on vacation, I was holed up in a hotel that had Pitch Black on pay-per-view and decided to give it a try. I was stunned. Not only did I wish I had seen it in the theatre - to truly appreciate the special effects - but I enjoyed the quirky dialogue between the characters. I appreciated the seeming abandon with which character upon character is removed from the story line without my abilities of prediction serving me in the slightest. The dialogue is often cheesy. These people are space faring commoners and the movie revolves around the predicamnet they face and not their lives. Their lives form the substance to which I temporarily attached care for each. Short-lived as it may be, this care for the characters extended throughout the movie and left me feeling very satisfied. I recommend this movie as a flick. It should have been seen by all in the theatre but we weren't all so fotunate. Check it out by yourself and if you like it pull the friends around. Give it a chance. Van Diesel is what it's all about. You'll recognize him from Saving Private Ryan. The washed out scenes reminded me of Gladiator and Three Kings. The world as seen through the eyes of the aliens put me back in Fincher's Alien3. While ultimately the plot is fantasy I let this movie take me away, in much the same way we all let Star Wars (the first three) move us and Starship Troopers moved us (appealed to our collective bizarre fascist sensability or something). Have fun with it. I know I did.

Help other customers find the most helpful reviews | Report abuse | Permalink

Was this review helpful to you? Yes No | Comment

18 of 19 people found the following review helpful

Cool monster flick, November 14, 2000

By Kathy - See all my reviews

This review is from: Pitch Black [VHS] (VHS Tape)

This is a damned good sci-fi horror flick. Okay, it isn't really scary (more tension inducing than terror inducing) and you do have to take it with a grain of salt, but in general, it is a very decent movie for the genre. I would say it is a fair bit like Aliens - same sort of feel, though not so humorous. (Well, they don't have a Hudson, so you know).
The beasts are great - sort of like dragons or basilisks with a bit of Starship Trooper bug thrown in. More straight predators than anything else, but with a genius for seeking out prey. The hardest part of the plot to accept is that anyone makes it out alive. Especially as the beasties dislike for light seems to fade when they get really hungry (see Johns with gun torch scene).
The acting is pretty cool by all actors, but yes, Vin Diesel is great as Riddick with his awesome "shine-job" and Cole Hauser has to come in for a mention with his fantastic portrayal of the morally ambiguous William Johns, bounty hunter. It was Cole Hauser who was the scene stealer in my opinion - Riddick was the central male character so he didn't have to steal scenes - they were already his. Riddick and Johns were excellent characters and both actors gave talented portrayals. I was really surprised to learn that Cole Hauser was 25 in real life - he made me believe his character was a jaded 30 at the very least.
The 3 central characters (Radha Mitchell as Carolyn, Cole Hauser as Johns, and Vin Diesel as Riddick) were used to explore the theme of courage with interesting results. Personally, I would consider all three characters as brave beyond belief but I wonder whether the movie intends me to think that way - after all, all three of them do some questionable things in the name of survival.
The director of this film has a bright future - it was well done and convincing. You could see that some effects were low-budget, but they were used so well that it didn't matter. The opening scenes could be described as awesome - the lighting, Riddick's voice over, the look and feel of the pilots area - all brilliant. All in all, a great movie and definitely worth watching more than once.

Help other customers find the most helpful reviews | Report abuse | Permalink

Was this review helpful to you? Yes No | Comment

83 of 101 people found the following review helpful

Taut, smart, enjoyable filmmaking, November 14, 2000

By **A Customer**

This review is from: Pitch Black (Unrated Version) (DVD)

Hell has surely frozen over. That's the only way to explain how David Twohy, writer-director of the so-bad-it's-hilarious Charlie Sheen skydiving epic "Terminal Velocity," has made a movie this good. It's not high art, but "Pitch Black" is a triumph within its genre: a suspenseful, intelligent monster movie with surprisingly deep characters.
A damaged spaceship loaded with cargo and cryo-sleeping passengers crash-lands on an alien world where three suns create perpetual daylight. At first, the survivors think their biggest problem is the vicious convict who's escaped from the wreckage. Then they discover the light-fearing predators lurking beneath the planet's surface. And then comes the total eclipse...
"Pitch Black" is a Diesel-powered movie-- Vin Diesel, that is. As the menacing convict Richard P. Riddick, Diesel gives a ferociously intelligent and charismatic performance, backed up by Twohy's surprisingly nuanced script. You'll come to root for Riddick as the movie wears on, but that doesn't necessarily mean you'll like him.
Radha Mitchell is also fine as the novice pilot Fry, battling inner and outer demons as she tries to hold the survivors together; Cole Hauser does a nice turn as Riddick's captor; and the fine supporting cast includes Keith David as a Muslim cleric (a refreshingly positive portrayal of Islam) and "Farscape"'s always-excellent Claudia Black.
"Pitch Black" is an embarrassment of riches for sci-fi fans: characters who continually surprise you, creepy creatures left mostly up to your imagination, and a stripped-down story that moves at a breakneck pace. Perfect popcorn entertainment-- just be sure you don't turn _all_ the lights off before you watch it...

Help other customers find the most helpful reviews | Report abuse | Permalink

Was this review helpful to you? Yes No | Comment (1)

18 of 20 people found the following review helpful

A Slick Sci-Fi Thriller, May 3, 2001

By A. Phillips - See all my reviews

REAL NAME

Example

Q: What is the **data?**

A: features about the movie and the users who evaluated it

Movie features: genre, actors, rating, length, etc.

Product Details

Genres	Science Fiction, Action, Horror
Director	David Twohy
Starring	Vin Diesel, Radha Mitchell
Supporting actors	Cole Hauser, Keith David, Lewis Fitz-Gerald, Claudia Black, Rhiana Gr Angela Moore, Peter Chiang, Ken Twohy
Studio	NBC Universal
MPAA rating	R (Restricted)
Captions and subtitles	English Details
Rental rights	24 hour viewing period. Details
Purchase rights	Stream instantly and download to 2 locations Details
Format	Amazon Instant Video (streaming online video and digital download)

User features: age, gender, location, etc.

A. Phillips

Reviewer ranking: #17,230,554

90% helpful

votes received on reviews

(151 of 167)

ABOUT ME

Enjoy the reviews...

ACTIVITIES

Reviews (16)

Public Wish List (2)

Listmania Lists (2)

Tagged Items (1)

Example

Movie recommendation:

$$f(\text{data}) \stackrel{?}{\to} \text{labels}$$

=

$$f(\text{user features}, \text{movie features}) \stackrel{?}{\to} \text{star rating}$$

Solution 1

Design a system based on **prior knowledge**, e.g.

```
def prediction(user, movie):
    if (user['age'] <= 14):
        if (movie['mpaa_rating']) == "G"):
            return 5.0
        else:
            return 1.0
    else if (user['age'] <= 18):
        if (movie['mpaa_rating']) == "PG"):
            return 5.0
….. Etc.
```

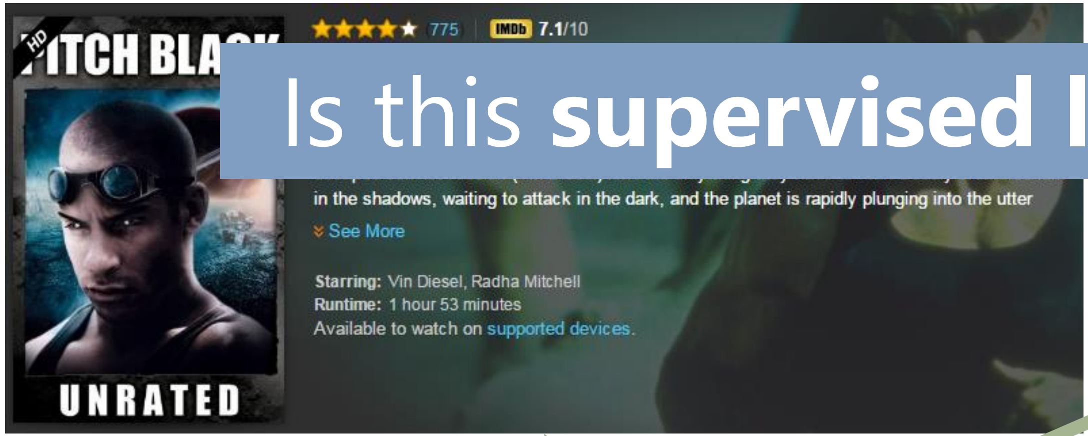
Is this **supervised learning?**

Solution 2

Identify words that I frequently mention in my social media posts, and recommend movies whose plot synopses use **similar** types of language

Social media posts

Plot synopsis



Is this supervised learning?

argmax similarity(synopsis, post)

Solution 3

Identify which attributes (e.g. actors, genres) are associated with positive ratings. Recommend movies that exhibit those attributes.

Is this **supervised learning?**

Solution 1

(design a system based on prior knowledge)

Disadvantages:

- Depends on possibly false **assumptions** about how users relate to items
- Cannot adapt to new data/information

Advantages:

- Requires no data!

Solution 2

(identify similarity between wall posts and synopses)

Disadvantages:

- Depends on possibly false **assumptions** about how users relate to items
- May not be adaptable to new settings

Advantages:

- Requires data, but does not require **labeled** data

Solution 3

(identify attributes that are associated with positive ratings)

Disadvantages:

- Requires a (possibly large) dataset of movies with labeled ratings

Advantages:

- Directly optimizes a measure we care about (predicting ratings)
- Easy to adapt to new settings and data

Supervised versus unsupervised learning

Learning approaches attempt to **model data** in order to solve a problem

Unsupervised learning approaches find patterns/relationships/structure in data, but **are not** optimized to solve a particular predictive task

Supervised learning aims to directly model the relationship between input and output variables, so that the output variables can be predicted accurately given the input

Regression

Regression is one of the simplest supervised learning approaches to learn relationships between input variables (features) and output variables (predictions)

Linear regression

Linear regression assumes a predictor of the form

$$X\theta = y$$

matrix of features
(data)

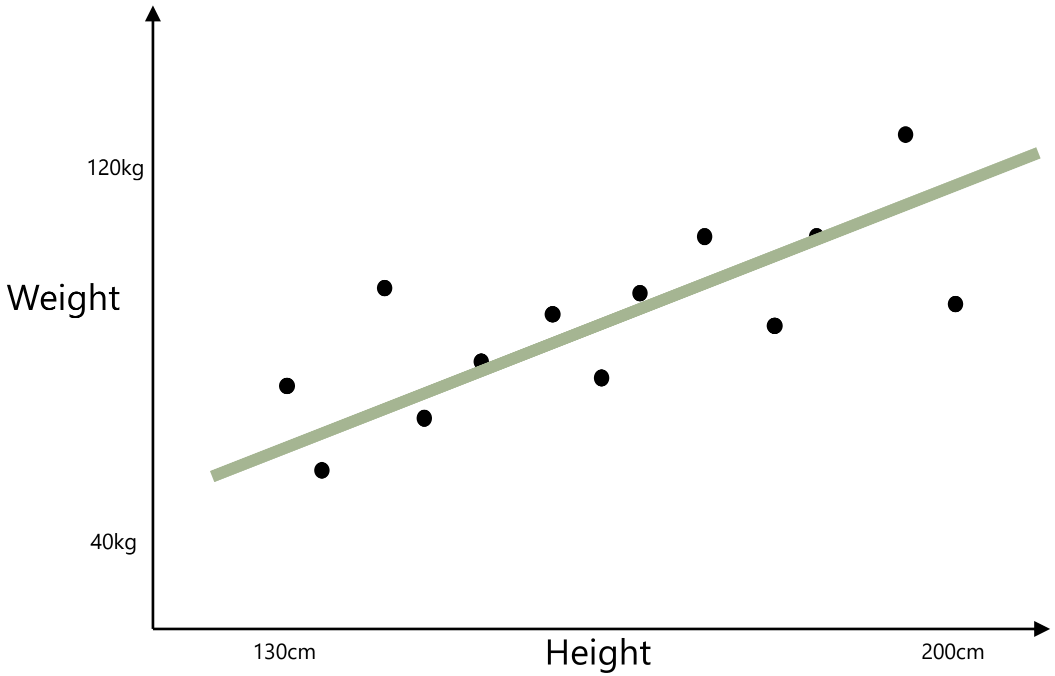
unknowns
(which features are relevant)

vector of outputs
(labels)

(or $Ax = b$ if you prefer)

Motivation: height vs. weight

Q: Can we find a line that (approximately) fits the data?



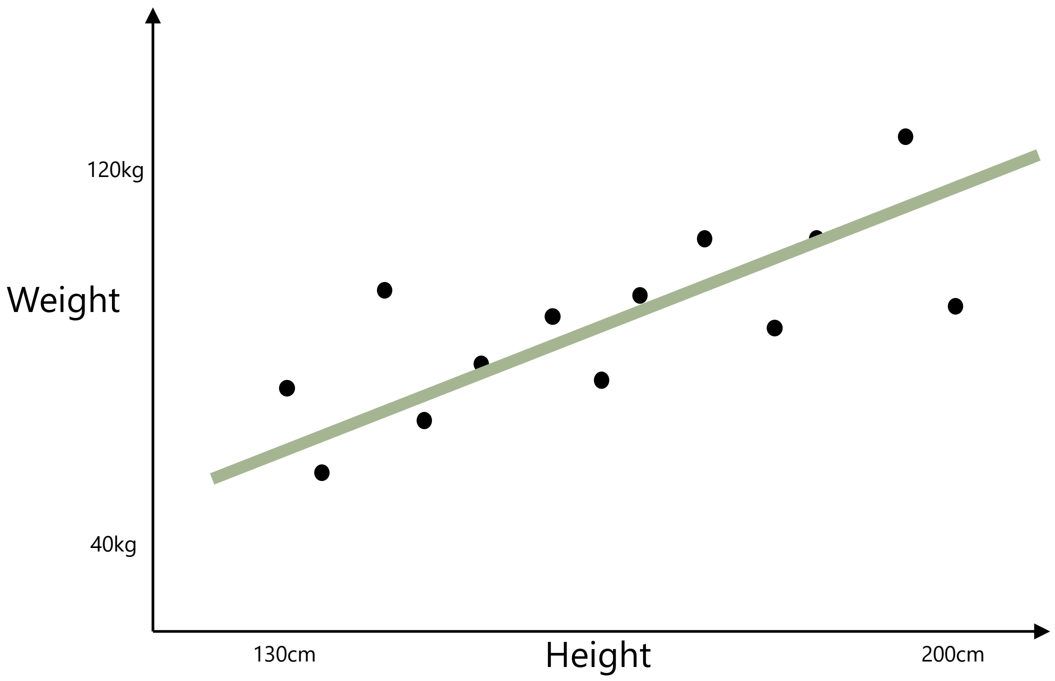
Motivation: height vs. weight

Q: Can we find a line that (approximately) fits the data?

- If we can find such a line, we can use it to make **predictions** (i.e., estimate a person's weight given their height)
- How do we **formulate** the problem of finding a line?
- If no line will fit the data exactly, how to **approximate?**
- What is the "best" line?

Recap: equation for a line

What is the formula describing the line?

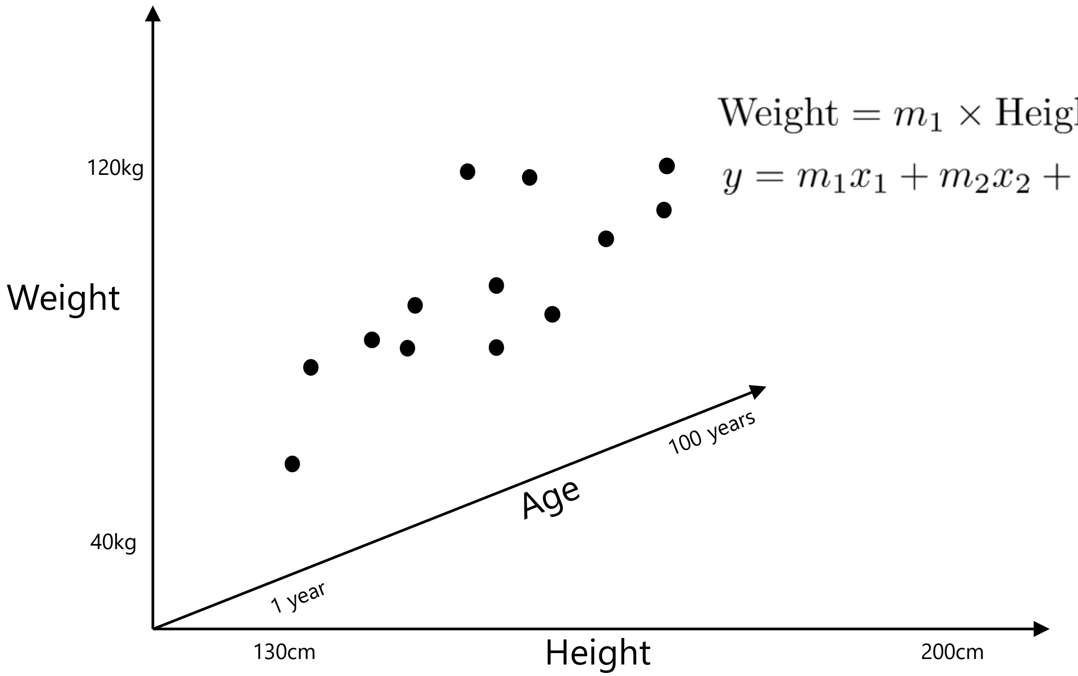


$$y = mx + b$$

$$\text{Weight} = m \times \text{Height} + b$$

Recap: equation for a line (plane)

What about in more dimensions?

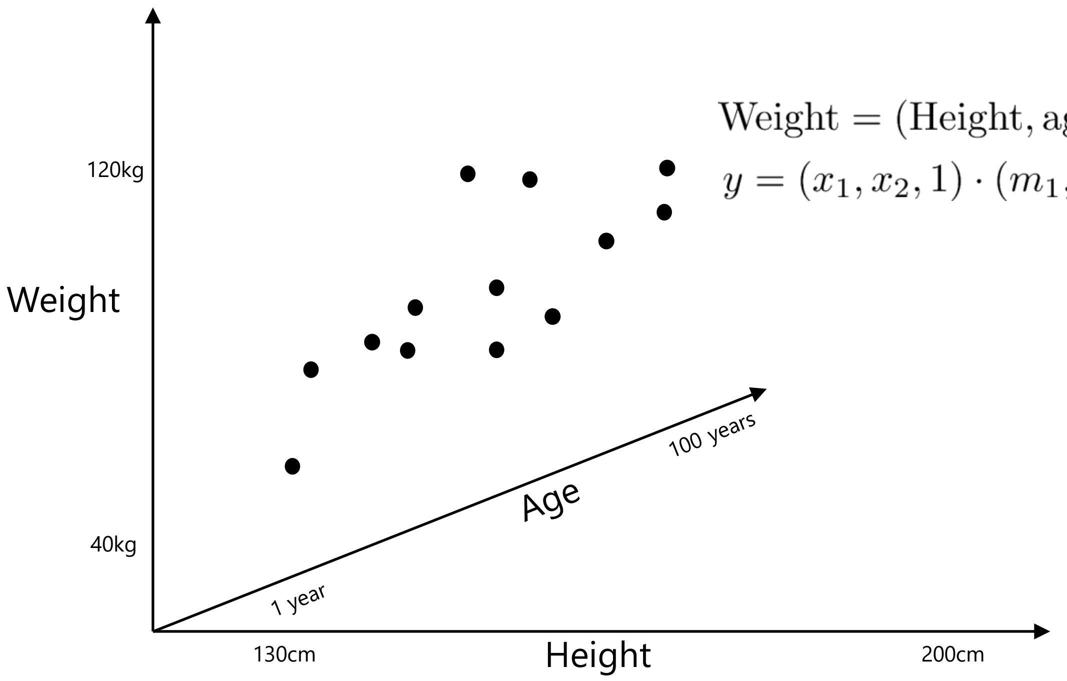


$$\text{Weight} = m_1 \times \text{Height} + m_2 \times \text{age} + b$$

$$y = m_1 x_1 + m_2 x_2 + b$$

Recap: equation for a line as an inner product

What about in more dimensions?



$$\text{Weight} = (\text{Height}, \text{age}, 1) \cdot (m_1, m_2, b)$$
$$y = (x_1, x_2, 1) \cdot (m_1, m_2, b)$$

Linear regression

Linear regression assumes a predictor of the form

$$X\theta = y$$

Q: Solve for theta
A:

Linear regression

Linear regression assumes a predictor of the form

$$X\theta = y$$

Q: Solve for theta

A: $\theta = (X^T X)^{-1} X^T y$

Learning Outcomes

- Explained **Supervised Learning** problems in terms of data, labels, and features
- Explained how regression can be setup in terms of lines (or hyperplanes) of best fit

Web Mining and Recommender Systems

Worked Example – Regression

Learning Goals

- Work through an example of a regression problem
- Introduce some simple **feature engineering** strategies

Linear regression

Linear regression assumes a predictor of the form

$$X\theta = y$$

Q: Solve for theta
A:

Linear regression

Linear regression assumes a predictor of the form

$$X\theta = y$$

Q: Solve for theta
A: $\theta = (X^T X)^{-1} X^T y$

Example 1

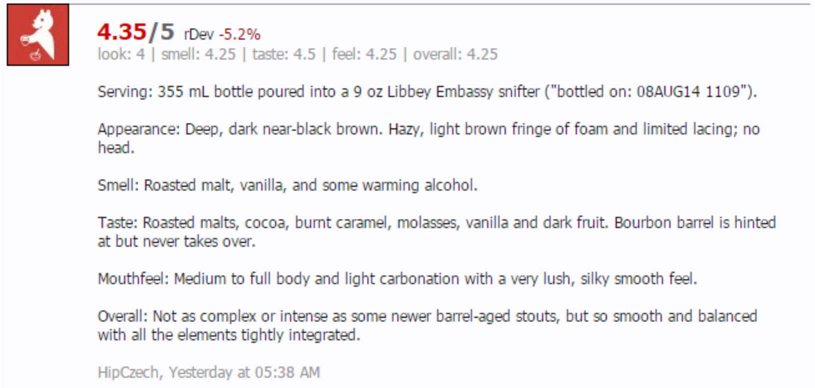
How do preferences toward certain beers vary with age?

Example 1

Beeradvocate

Beers:

Ratings/reviews:



User profiles:

Example 1

50,000 reviews are available on
http://cseweb.ucsd.edu/classes/fa19/cse258-a/data/beer_50000.json
(see course webpage)

Example 1

Real-valued features

How do preferences toward certain beers vary with age?
How about **ABV**?

(code for all examples is on the course webpage)

Example 1

Real-valued features

What is the interpretation of:

$$\theta = (3.4, 10e^{-7})$$

(code for all examples is on the course webpage)

Example 2

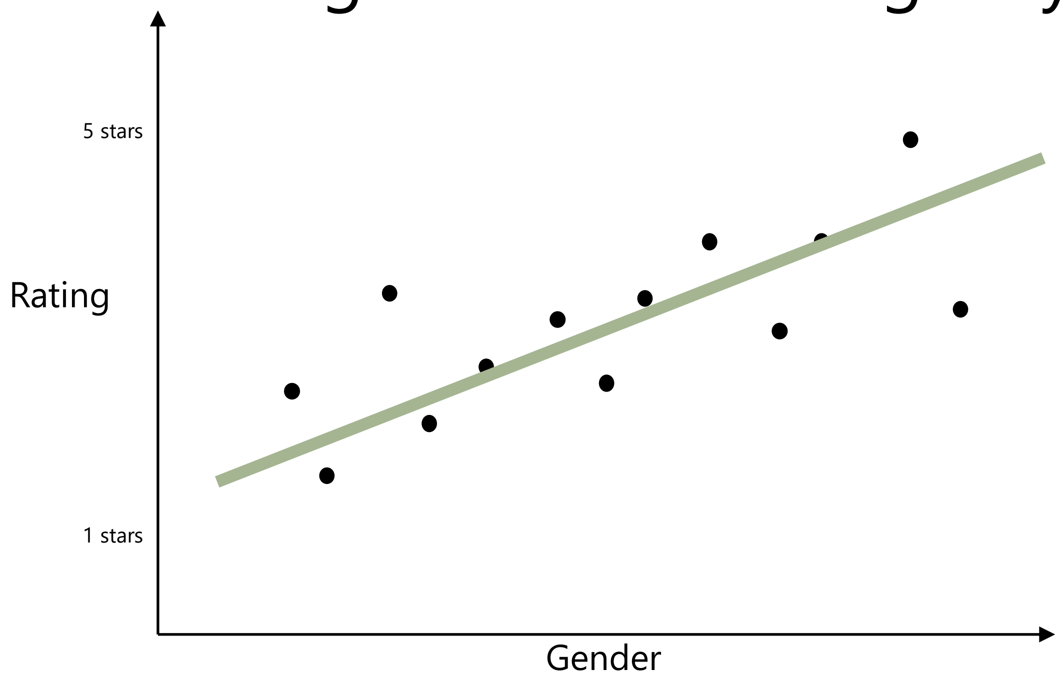
Categorical features

How do beer preferences vary as a function of **gender**?

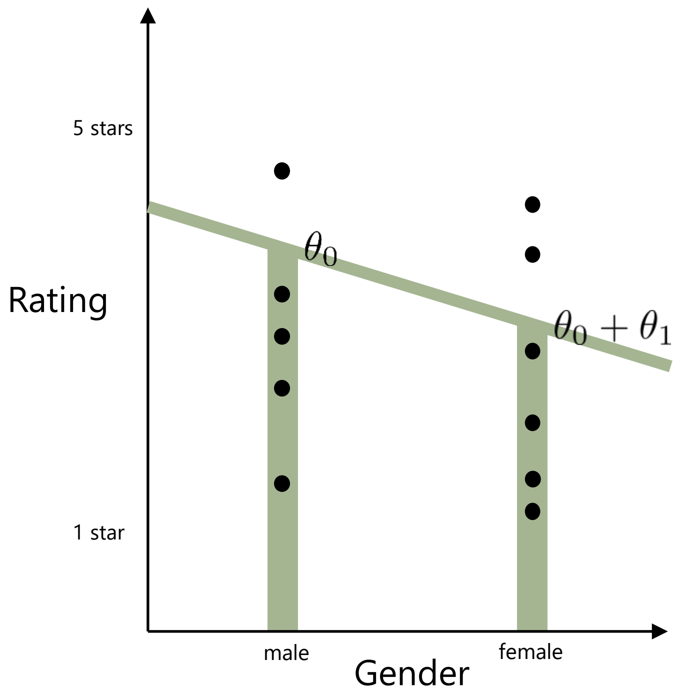
(code for all examples is on the course webpage)

Example 2

E.g. How does rating vary with **gender?**



Example 2



θ_0 is the (predicted/average) rating for males

θ_1 is the **how much higher** females rate than males (in this case a negative number)

We're really still fitting a line though!

Exercise

How would you build a feature to represent the **month**, and the impact it has on people's rating behavior?

Learning Outcomes

- Worked through a simple regression problem
- Began some simple **feature engineering** with binary features

Web Mining and Recommender Systems

Regression – Feature Transforms & Worked Example

Learning Goals

- Work through a real example of a regression problem
- Discuss the topic of **feature engineering** in more depth

Regression

Regression is one of the simplest supervised learning approaches to learn relationships between input variables (features) and output variables (predictions)

Linear regression

Linear regression assumes a predictor of the form

$$X\theta = y$$

matrix of features (data)

unknowns (which features are relevant)

vector of outputs (labels)

(or $Ax = b$ if you prefer)

Linear regression

Linear regression assumes a predictor of the form

$$X\theta = y$$

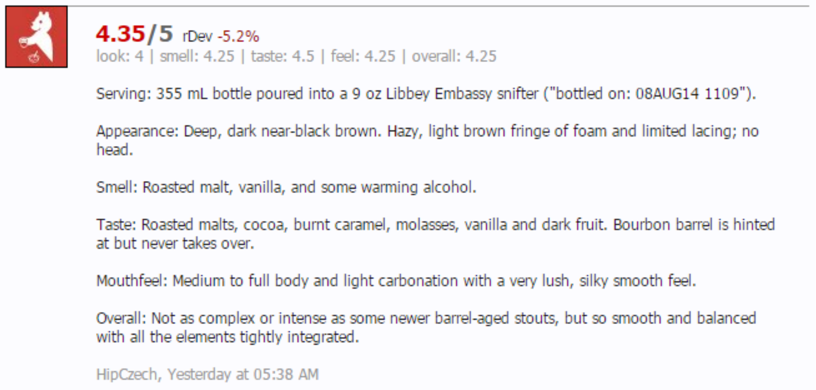
Q: Solve for theta
A: $\theta = (X^T X)^{-1} X^T y$

Example

Beeradvocate

Beers:

Ratings/reviews:



User profiles:

Example

Real-valued features

How do preferences toward certain beers vary with age?
How about **ABV**?

(code for all examples on course webpage)

Example: Polynomial functions

What about something like ABV^2?

$$\text{rating} = \theta_0 + \theta_1 \times \text{ABV} + \theta_2 \times \text{ABV}^2 + \theta_3 \times \text{ABV}^3$$

- Note that this is perfectly straightforward: the model still takes the form

$$\text{weight} = \theta \cdot x$$

- We just need to use the feature vector

```
x = [1, ABV, ABV^2, ABV^3]
```

Fitting complex functions

Note that we can use the same approach to fit arbitrary functions of the features! E.g.:

$$\text{Rating} = \theta_0 + \theta_1 \times \text{ABV} + \theta_2 \times \text{ABV}^2 + \theta_3 \exp(\text{ABV}) + \theta_4 \sin(\text{ABV})$$

- We can perform arbitrary combinations of the **features** and the model will still be linear in the **parameters** (theta):

$$\text{Rating} = \theta \cdot x$$

Fitting complex functions

The same approach would **not** work if we wanted to transform the parameters:

$$\text{Rating} = \theta_0 + \theta_1 \times \text{ABV} + \theta_2^2 \times \text{ABV} + \sigma(\theta_3) \times \text{ABV}$$

- The **linear** models we've seen so far do not support these types of transformations (i.e., they need to be linear in their parameters)
- There *are* alternative models that support non-linear transformations of parameters, e.g. neural networks

Learning Outcomes

- Worked through a real regression example
- Explained how to use more complex feature transforms to fit (e.g.) polynomials with regression algorithms

Web Mining and Recommender Systems

Regression – Categorical Features

Learning Goals

- Explain how to use categorical features within regression algorithms

Example

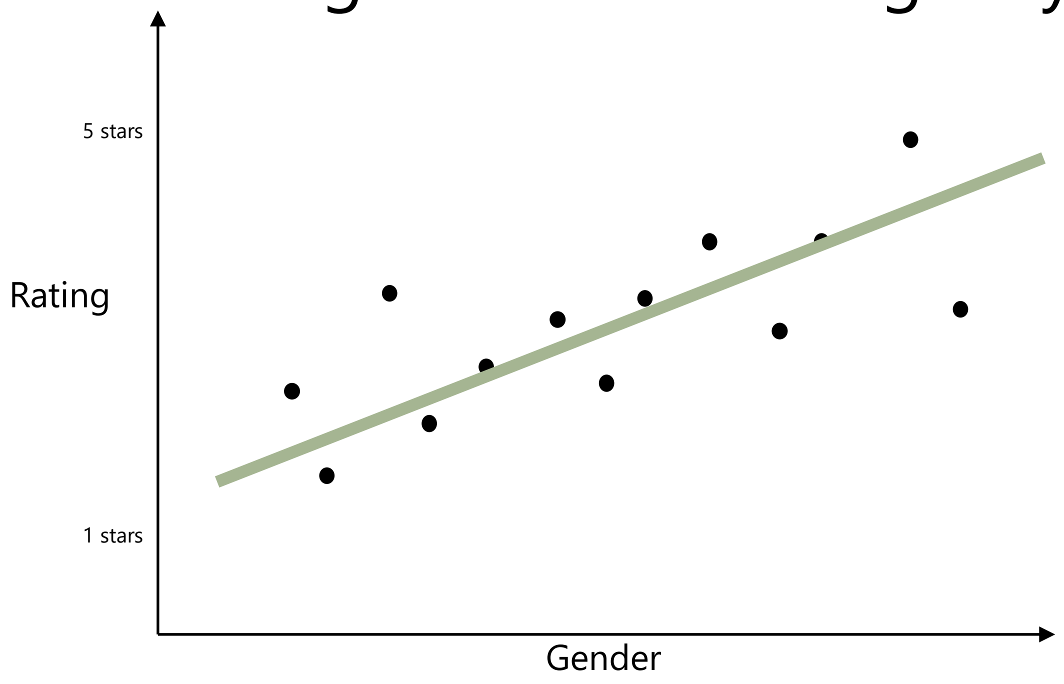
Categorical features

How do beer preferences vary as a function of **gender**?

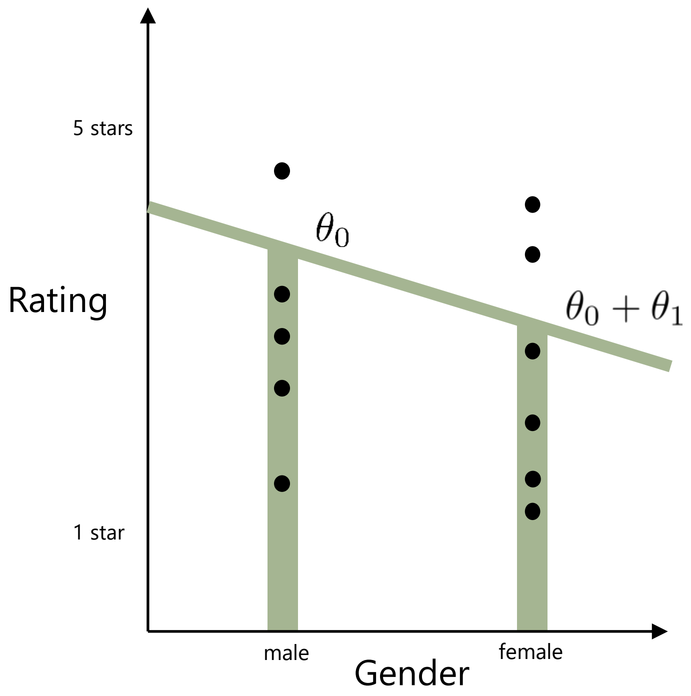
(code for all examples is the course webpage)

Example

E.g. How does rating vary with **gender?**



Example



θ_0 is the (predicted/average) rating for males

θ_1 is the **how much higher** females rate than males (in this case a negative number)

We're really still fitting a line though!

Motivating examples

What if we had more than two values?
(e.g {"male", "female", "other", "not specified"})

Could we apply the same approach?

$$\text{Rating} = \theta_0 + \theta_1 \times \text{gender}$$

gender = **0 if "male", 1 if "female", 2 if "other", 3 if "not specified"**

$\text{Rating} = \theta_0$ **if male**

$\text{Rating} = \theta_0 + \theta_1$ **if female**

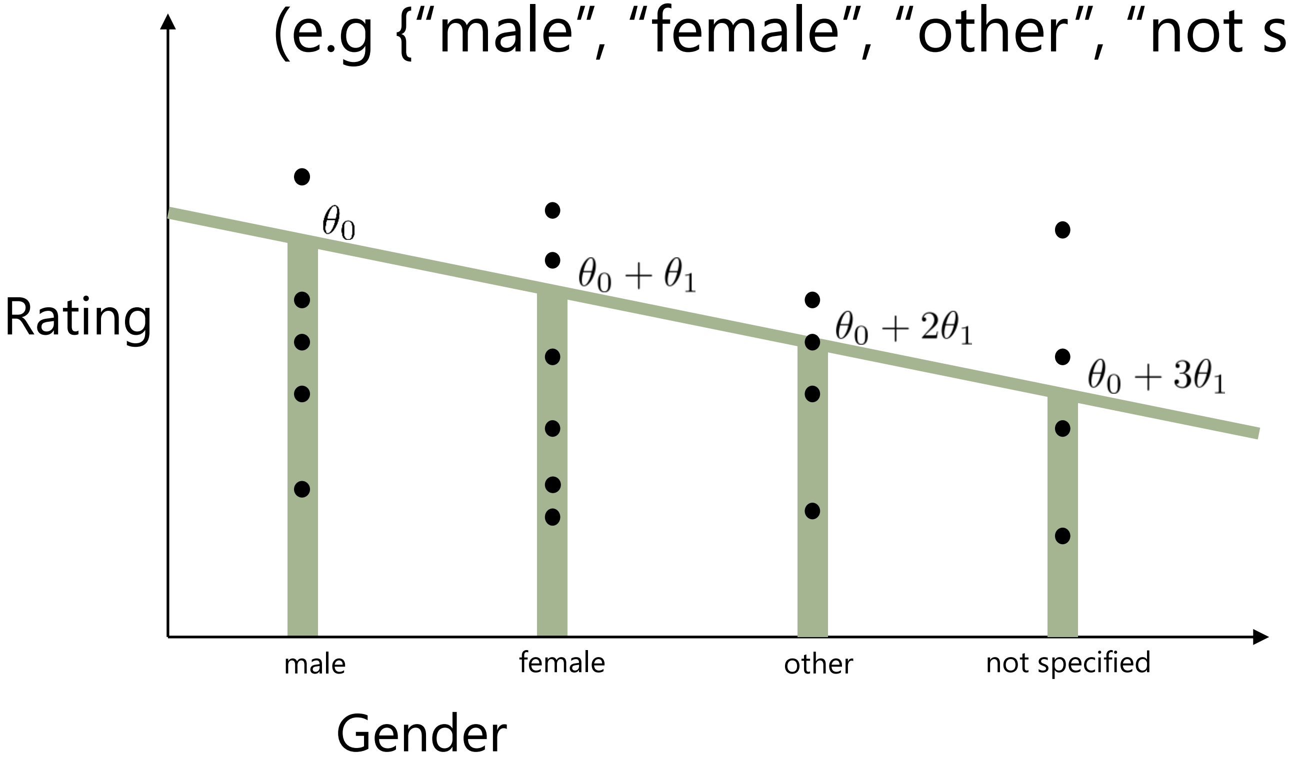
$\text{Rating} = \theta_0 + 2\theta_1$ **if other**

$\text{Rating} = \theta_0 + 3\theta_1$ **if not specified**

Motivating examples

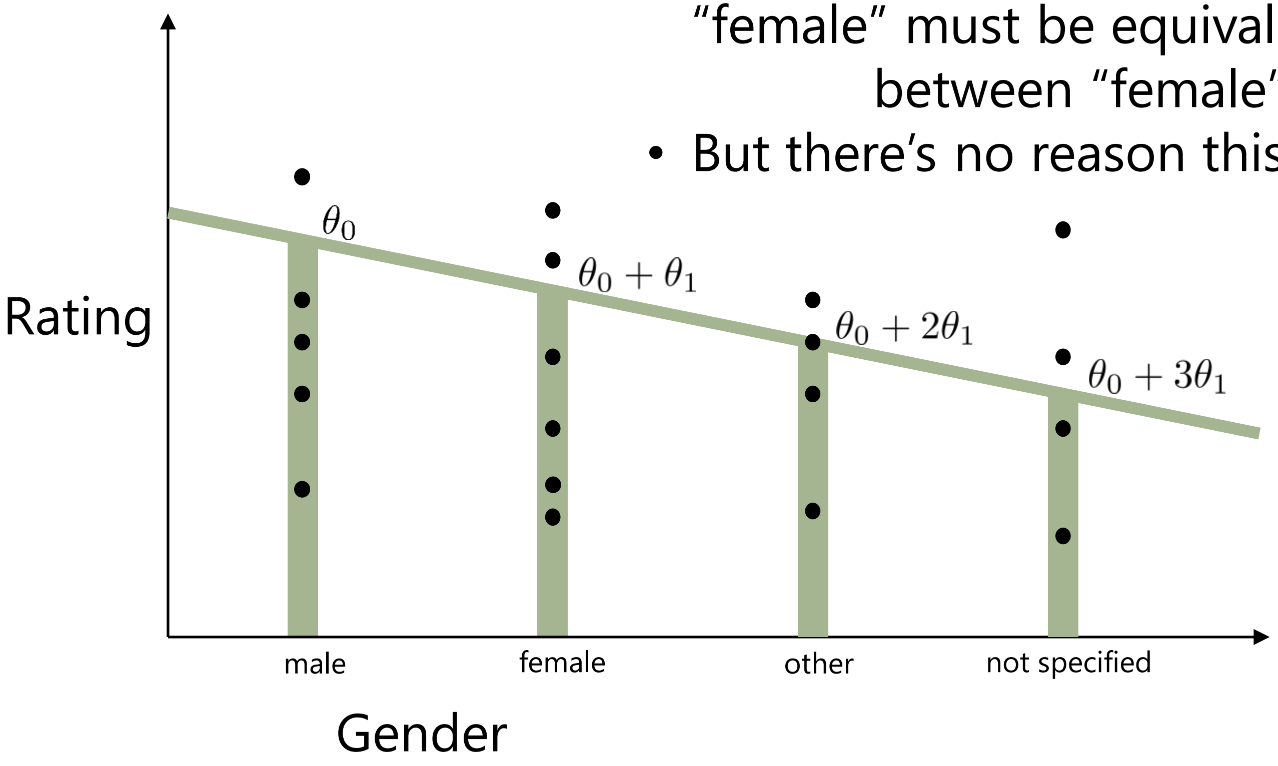
What if we had more than two values?

(e.g {“male”, “female”, “other”, “not specified”})



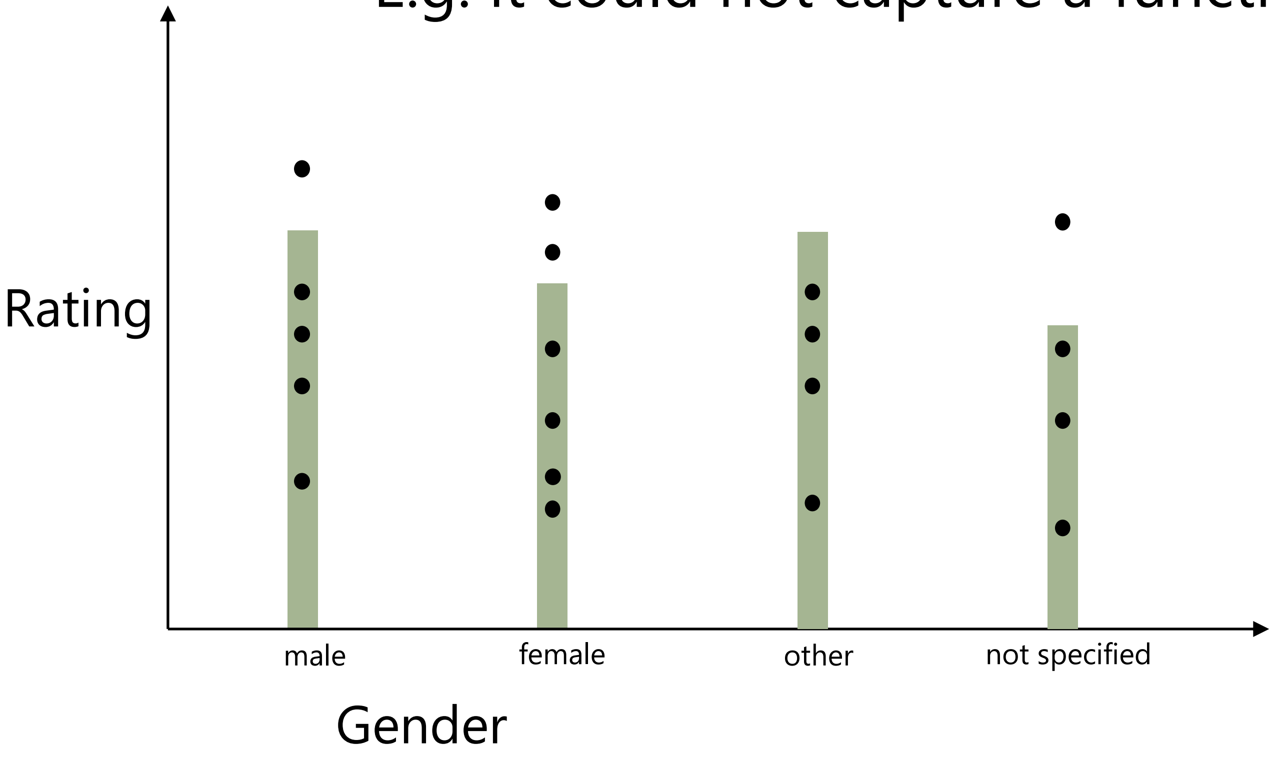
Motivating examples

- This model is **valid,** but won't be very **effective**
- It assumes that the difference between "male" and "female" must be equivalent to the difference between "female" and "other"
- But there's no reason this should be the case!



Motivating examples

E.g. it could not capture a function like:



Motivating examples

Instead we need something like:

$$\text{Rating} = \theta_0$$ **if male**

$$\text{Rating} = \theta_0 + \theta_1$$ **if female**

$$\text{Rating} = \theta_0 + \theta_2$$ **if other**

$$\text{Rating} = \theta_0 + \theta_3$$ **if not specified**

Motivating examples

This is equivalent to:

$$(\theta_0, \theta_1, \theta_2, \theta_3) \cdot (1; \text{feature})$$

where feature = [1, 0, 0] for “female”
feature = [0, 1, 0] for “other”
feature = [0, 0, 1] for “not specified”

Concept: One-hot encodings

feature = [1, 0, 0] for "female"
feature = [0, 1, 0] for "other"
feature = [0, 0, 1] for "not specified"

- This type of encoding is called a **one-hot encoding** (because we have a feature vector with only a single "1" entry)
- Note that to capture 4 possible categories, we only need three dimensions (a dimension for "male" would be redundant)
- This approach can be used to capture a variety of categorical feature types, as well as objects that belong to multiple categories

Linearly dependent features

Linearly dependent features

Learning Outcomes

- Showed how to use categorical features within regression algorithms
- Introduced the concept of a **"one-hot"** encoding
- Discussed linear dependence of features

Web Mining and Recommender Systems

Regression – Temporal Features

Learning Goals

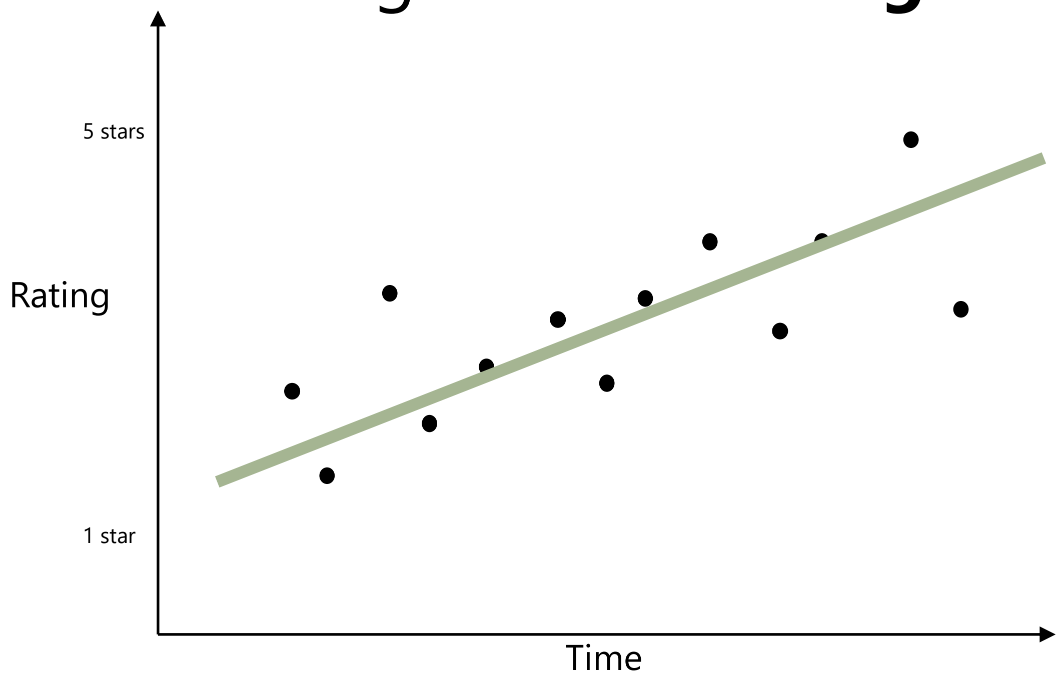
- Explain how to use temporal features within regression algorithms

Example

How would you build a feature to represent the **month**, and the impact it has on people's rating behavior?

Motivating examples

E.g. How do **ratings** vary with **time**?



Motivating examples

E.g. How do **ratings** vary with **time**?

- In principle this picture looks okay (compared our previous example on categorical features) – we're predicting a **real valued** quantity from **real valued** data (assuming we convert the date string to a number)
- So, what would happen if (e.g. we tried to train a predictor based on the month of the year)?

Motivating examples

E.g. How do **ratings** vary with **time**?

- Let's start with a simple feature representation, e.g. map the month name to a month number:

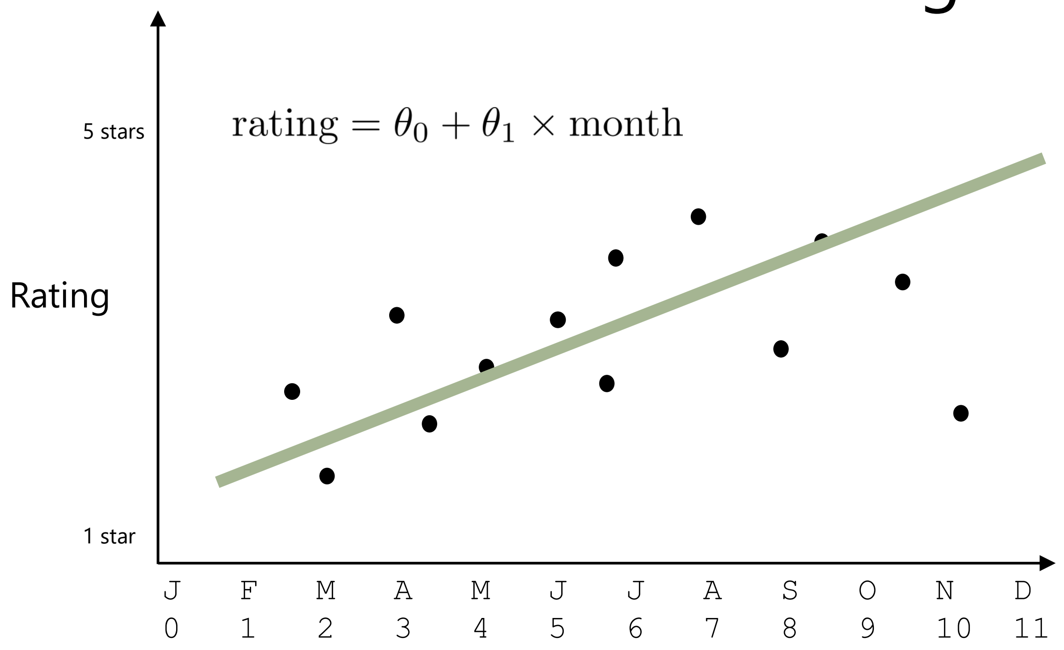
$$\text{rating} = \theta_0 + \theta_1 \times \text{month}$$

where

```
Jan = [0]
Feb = [1]
Mar = [2]
etc.
```

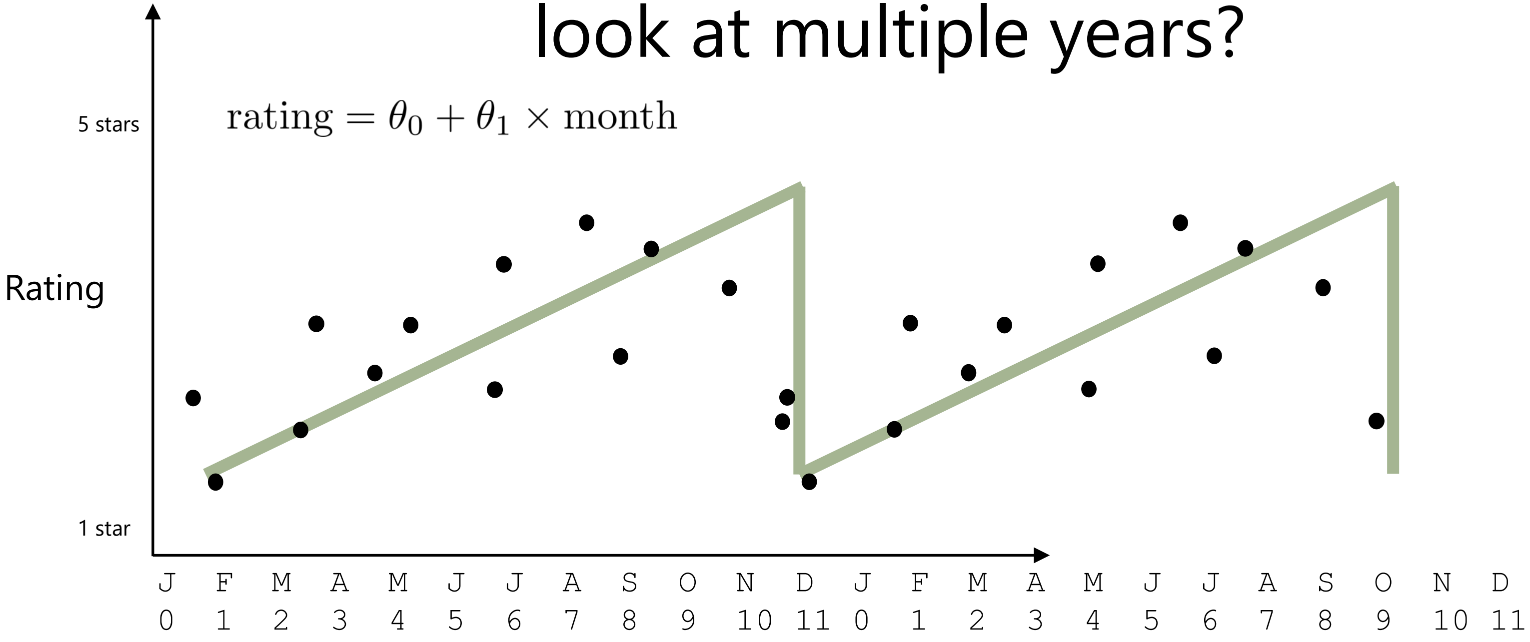
Motivating examples

The model we'd learn might look something like:



Motivating examples

This seems fine, but what happens if we look at multiple years?



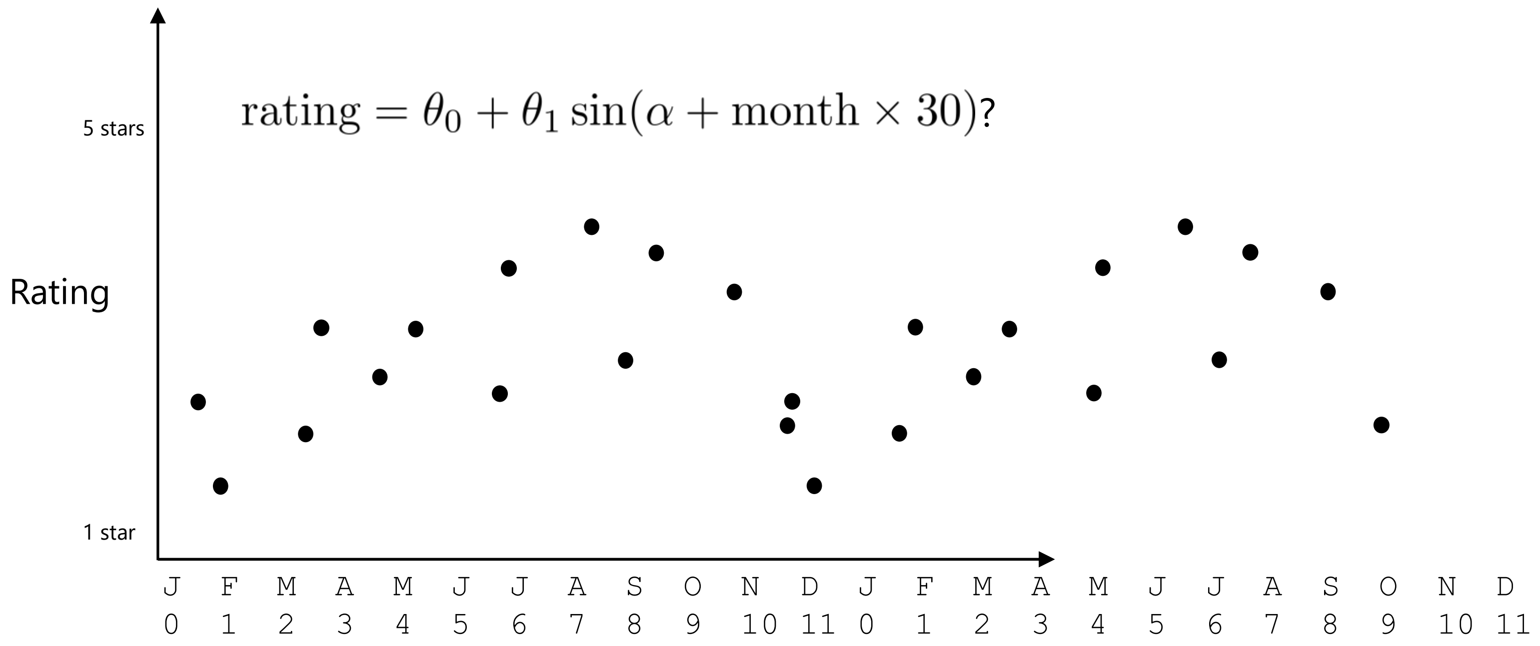
Modeling temporal data

This seems fine, but what happens if we look at multiple years?

- This representation implies that the model would "wrap around" on December 31 to its January 1st value.
- This type of "sawtooth" pattern probably isn't very realistic

Modeling temporal data

What might be a more realistic shape?



Modeling temporal data

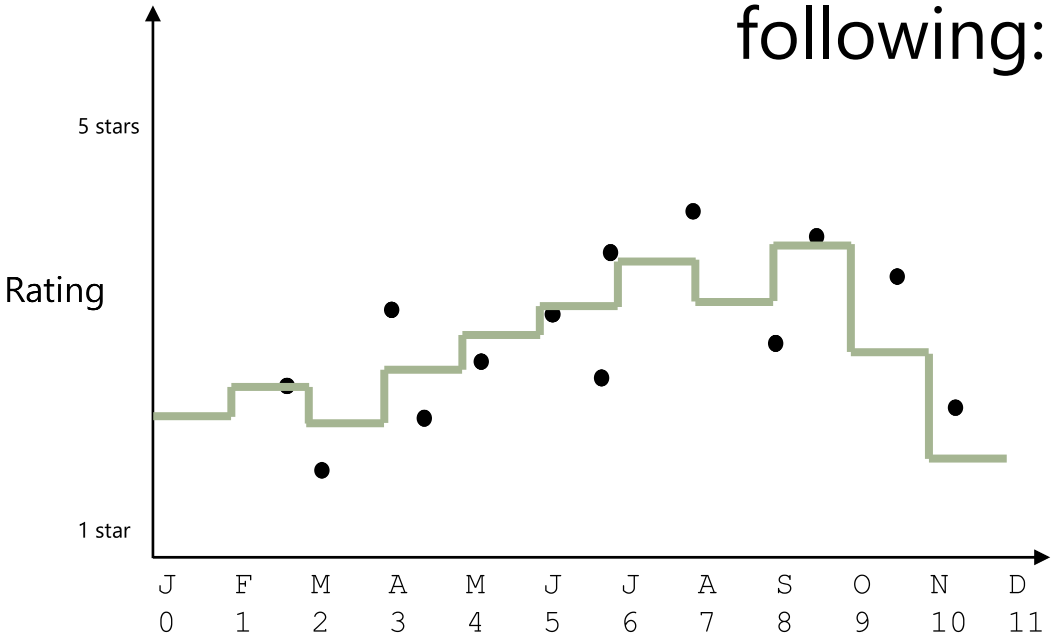
Fitting some periodic function like a sin wave would be a valid solution, but is difficult to get right, and fairly inflexible

- Also, it's not a **linear model**

- **Q:** What's a class of functions that we can use to capture a more flexible variety of shapes?
- **A:** Piecewise functions!

Concept: Fitting piecewise functions

We'd like to fit a function like the following:



Fitting piecewise functions

In fact this is very easy, even for a linear model! This function looks like:

$$\text{rating} = \theta_0 + \theta_1 \times \delta(\text{is Feb}) + \theta_2 \times \delta(\text{is Mar}) + \theta_3 \times \delta(\text{is Apr}) \cdots$$

1 if it's Feb, 0 otherwise

- Note that we don't need a feature for January
- i.e., theta_0 captures the January value, theta_1 captures the *difference* between February and January, etc.

Fitting piecewise functions

Or equivalently we'd have features as follows:

$$\text{rating} = \theta \cdot x$$ where

```
x = [1,1,0,0,0,0,0,0,0,0,0,0] if February
    [1,0,1,0,0,0,0,0,0,0,0,0] if March
    [1,0,0,1,0,0,0,0,0,0,0,0] if April
    ...
    [1,0,0,0,0,0,0,0,0,0,0,1] if December
```

Fitting piecewise functions

Note that this is still a form of **one-hot** encoding, just like we saw in the "categorical features" example

- This type of feature is very flexible, as it can handle complex shapes, periodicity, etc.
- We could easily increase (or decrease) the resolution to a week, or an entire season, rather than a month, depending on how fine-grained our data was

Concept: Combining one-hot encodings

We can also extend this by combining several one-hot encodings together:

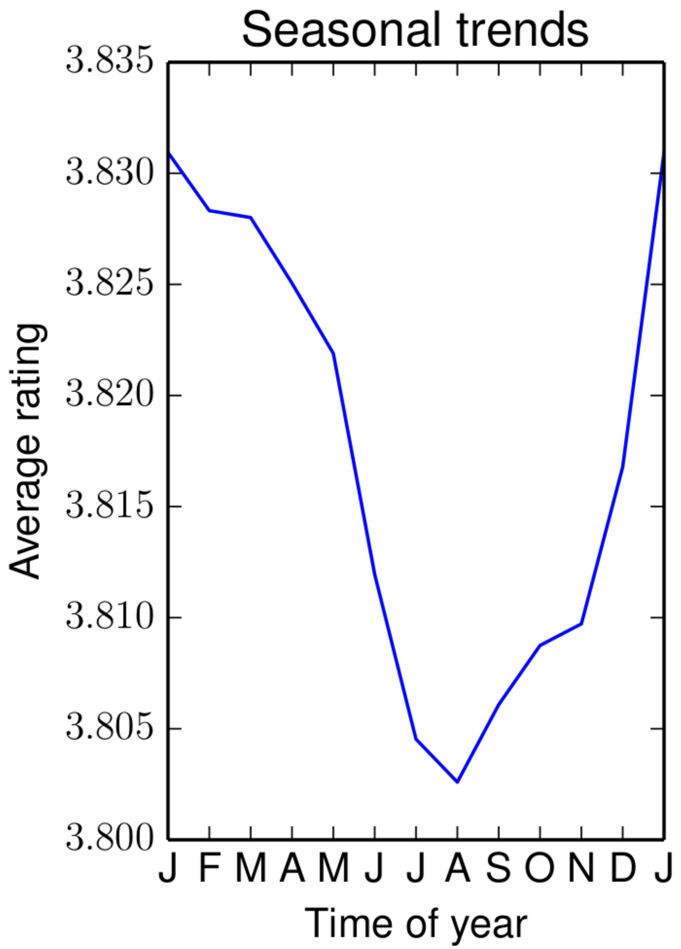
$$\text{rating} = \theta \cdot x = \theta \cdot [x_1; x_2] \quad \text{where}$$

```
x1 = [1,1,0,0,0,0,0,0,0,0,0,0] if February
     [1,0,1,0,0,0,0,0,0,0,0,0] if March
     [1,0,0,1,0,0,0,0,0,0,0,0] if April
     ...
     [1,0,0,0,0,0,0,0,0,0,0,1] if December
```

```
x2 = [1,0,0,0,0,0] if Tuesday
     [0,1,0,0,0,0] if Wednesday
     [0,0,1,0,0,0] if Thursday
     ...
```

What does the data actually look like?

Season vs. rating (overall)



Learning Outcomes

- Explained how to use temporal features within regression algorithms
- Showed how to use one-hot encodings to capture trends in periodic data

Web Mining and Recommender Systems

Regression Diagnostics

Learning Goals

- Show how to **evaluate** regression algorithms

Today: Regression diagnostics

Mean-squared error (MSE)

$$\frac{1}{N}\|y - X\theta\|_2^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}(y_i - X_i \cdot \theta)^2$$

Regression diagnostics

Q: Why MSE (and not mean-absolute-error or something else)

Regression diagnostics

Regression diagnostics

Coefficient of determination

Q: How low does the MSE have to be before it's "low enough"?

A: It depends! The MSE is proportional to the **variance** of the data

Coefficient of determination

(R^2 statistic)

Mean:

Variance:

MSE:

Coefficient of determination

(R^2 statistic)

Mean: $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$

Variance: $Var(y) = \frac{1}{N}\sum_{i=1}^{N}(\bar{y} - y_i)^2$

MSE: $\frac{1}{N}\sum_{i=1}^{N}(X_i \cdot \theta - y_i)^2$

Coefficient of determination

(R^2 statistic)

$$FVU(f) = \frac{MSE(f)}{Var(y)}$$

(FVU = fraction of variance unexplained)

FVU(f) = 1 → Trivial predictor

FVU(f) = 0 → Perfect predictor

Regression diagnostics

Coefficient of determination

(R^2 statistic)

$$R^2 = 1 - FVU(f) = 1 - \frac{MSE(f)}{Var(y)}$$

R^2 = 0 → Trivial predictor

R^2 = 1 → Perfect predictor

Learning Outcomes

- Showed how to **evaluate** regression algorithms
- Introduced the **Mean Squared Error** and **R^2 coefficient**
- Explained the relationship between the MSE and the variance

Web Mining and Recommender Systems

Overfitting

Learning Goals

- Introduce the concepts of **overfitting** and **regularization**

Overfitting

Q: But can't we get an R^2 of 1 (MSE of 0) just by throwing in enough random features?

A: Yes! This is why MSE and R^2 should always be evaluated on data that **wasn't** used to train the model

A good model is one that **generalizes to new data**

Overfitting

When a model performs well on **training** data but doesn't generalize, we are said to be **overfitting**

Overfitting

When a model performs well on **training** data but doesn't generalize, we are said to be **overfitting**

Q: What can be done to avoid overfitting?

Occam's razor

"Among competing hypotheses, the one with the fewest assumptions should be selected"

Occam's razor

$$X\theta = y$$

↑ "hypothesis"

Q: What is a "complex" versus a "simple" hypothesis?

Occam’s razor

A1: A “simple” model is one where theta has few non-zero parameters

(only a few features are relevant)

A2: A “simple” model is one where theta is almost uniform

(few features are significantly more relevant than others)

Occam's razor

A1: A "simple" model is one where theta has few non-zero parameters → $||\theta||_1$ is small

A2: A "simple" model is one where theta is almost uniform → $||\theta||_2$ is small

“Proof”

Regularization

Regularization is the process of penalizing model complexity during training

$$\arg\min_\theta = \frac{1}{N}||y - X\theta||_2^2 + \lambda||\theta||_2^2$$

MSE

(l2) model complexity

Regularization

Regularization is the process of penalizing model complexity during training

$$\arg\min_\theta = \frac{1}{N}\|y - X\theta\|_2^2 + \lambda\|\theta\|_2^2$$

How much should we trade-off accuracy versus complexity?

Optimizing the (regularized) model

$$\arg\min_\theta = \underbrace{\frac{1}{N}\|y - X\theta\|_2^2 + \lambda\|\theta\|_2^2}_{f(\theta)}$$

- Could look for a closed form solution as we did before
- Or, we can try to solve using **gradient descent**

Optimizing the (regularized) model

Gradient descent:

1. Initialize θ at random
2. While (not converged) do

$$\theta := \theta - \alpha f'(\theta)$$

All sorts of annoying issues:

- How to initialize theta?
- How to determine when the process has converged?
- How to set the step size alpha

These aren't really the point of this class though

Optimizing the (regularized) model

$$f(\theta) = \frac{1}{N}\|y - X\theta\|_2^2 + \lambda\|\theta\|_2^2$$

$$\frac{\partial f}{\partial \theta_k} \;?$$

Optimizing the (regularized) model

Gradient descent in scipy: code on course webpage

(see also “ridge regression” in the “sklearn” module)

Learning Outcomes

- Introduced the concepts of **overfitting** and **regularization**
- Showed how to regularize models using the l1 and l2 norms
- (very briefly) touched on gradient descent

Web Mining and Recommender Systems

Model Selection & Summary

Learning Goals

- Discuss **model selection** and **validation sets**
- Summarize our discussion on regression

Model selection

$$\arg\min_{\theta} = \frac{1}{N}\|y - X\theta\|_2^2 + \lambda\|\theta\|_2^2$$

How much should we trade-off accuracy versus complexity?

Each value of lambda generates a different model. **Q:** How do we select which one is the best?

Model selection

How to select which model is best?

A1: The one with the lowest training error?

A2: The one with the lowest test error?

We need a **third** sample of the data that is not used for training or testing

Model selection

A **validation set** is constructed to "tune" the model's parameters

- Training set: used to **optimize the model's parameters**
- Test set: used to report how well we expect the model to perform on **unseen data**
- Validation set: used to **tune** any model parameters that are not directly optimized

A few "theorems" about training, validation, and test sets

- The training error **increases** as lambda **increases**
- The validation and test error are at least as large as the training error (assuming infinitely large random partitions)
- The validation/test error will usually have a "sweet spot" between under- and over-fitting

Model selection

Summary: Regression

- Linear regression and least-squares
- (a little bit of) feature design
- Overfitting and regularization
- Gradient descent
- Training, validation, and testing
- Model selection