CSE 255 – Lecture 3

Data Mining and Predictive Analytics

Supervised learning – Classification

Last week...

Last week we started looking at **supervised learning problems**

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

Last week...

We studied **linear regression**, in order to learn linear relationships between features and parameters to predict **real-valued** outputs

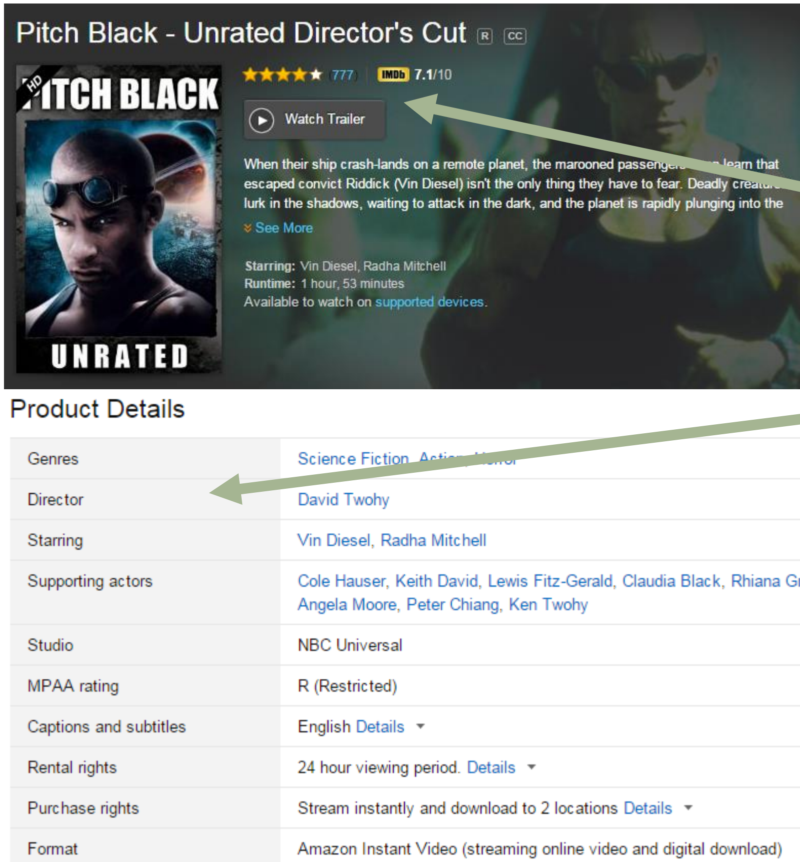
$$X\theta = y$$

matrix of features
(data)

unknowns
(which features are relevant)

vector of outputs
(labels)

Last week...



ratings

features

$$f(\text{user features}, \text{movie features}) \overset{?}{\rightarrow} \text{star rating}$$

Four important ideas from last week:

1) Regression can be cast in terms of **maximizing a likelihood**

$$y_i = x_i \cdot \theta + \mathcal{N}(0, \sigma)$$

$$y_i - x_i \cdot \theta \sim \mathcal{N}(0, \sigma)$$

$$\prod_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i \cdot \theta - y_i)^2}{2\sigma^2}}$$

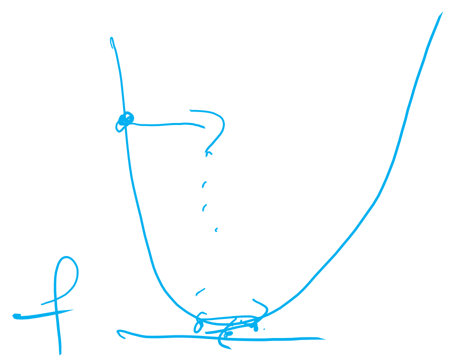
$$\sum_i (x_i \cdot \theta - y_i)^2 \longleftarrow \text{minimize}$$

Four important ideas from last week:

2) Gradient descent for model optimization

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

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



Four important ideas from last week:

3) Regularization & Occam's razor

Regularization is the process of penalizing model complexity during training

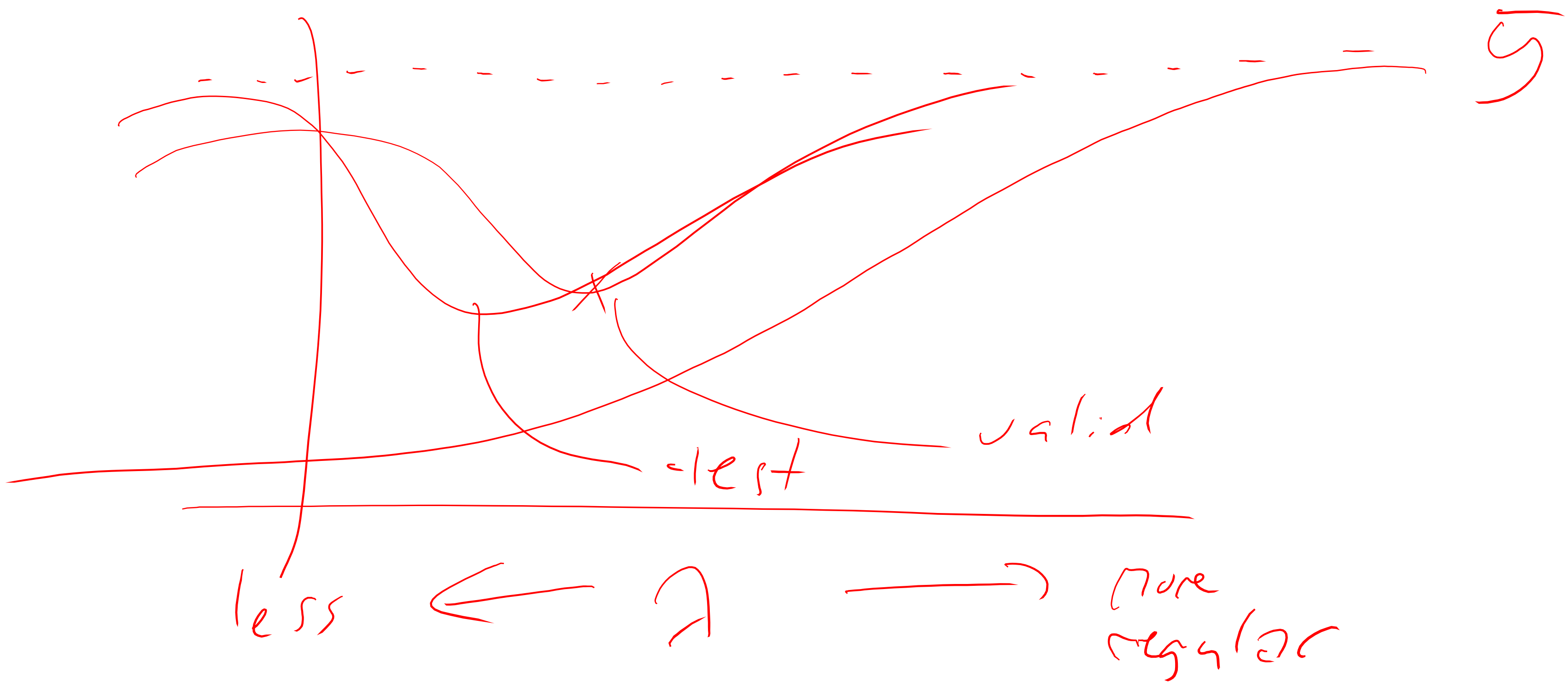
$$\arg\min_{\theta} = \frac{1}{N}\|y - X\theta\|_2^2 + \lambda\|\theta\|_2^2$$

How much should we trade-off accuracy versus complexity?

Four important ideas from last week:

4) Regularization pipeline

1. Training set – select model parameters
2. Validation set – to choose amongst models (i.e., hyperparameters)
3. Test set – just for testing!



Today...

How can we predict **binary** or **categorical** variables?

$$f(\text{data}) \xrightarrow{?} \text{labels}$$

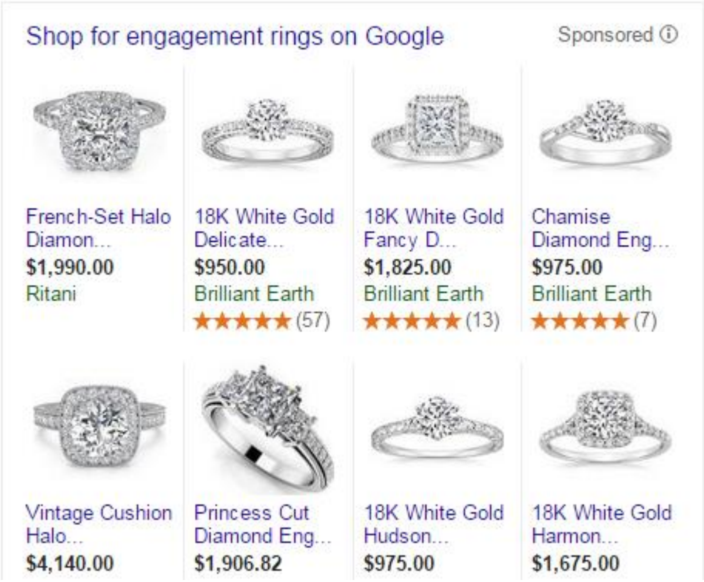
{0,1}, {True, False}

{1, ... , N}

Today…

Will I **purchase** this product?

(yes)



Will I **click on** this ad?

(no)

Today...

What animal appears in this image?

(mandarin duck)

Today...

What are the **categories** of the item being described?

(book, fiction, philosophical fiction)

From Booklist

Houellebecq's deeply philosophical novel is about an alienated young man searching for happiness in the computer age. Bored with the world and too weary to try to adapt to the foibles of friends and coworkers, he retreats into himself, descending into depression while attempting to analyze the passions of the people around him. Houellebecq uses his nameless narrator as a vehicle for extended exploration into the meanings and manifestations of love and desire in human interactions. Ironically, as the narrator attempts to define love in increasingly abstract terms, he becomes less and less capable of experiencing that which he is so desperate to understand. Intelligent and well written, the short novel is a thought-provoking inspection of a generation's confusion about all things sexual. Houellebecq captures precisely the cynical disillusionment of disaffected youth. *Bonnie Johnston --This text refers to an out of print or unavailable edition of this title.*

Today…

We'll attempt to build **classifiers** that make decisions according to rules of the form

$$y_i = \begin{cases} 1 & \text{if } X_i \cdot \theta > 0 \\ 0 & \text{otherwise} \end{cases}$$

This week…

1. Naïve Bayes

Assumes an **independence** relationship between the features and the class label and "learns" a simple model by counting

2. Logistic regression

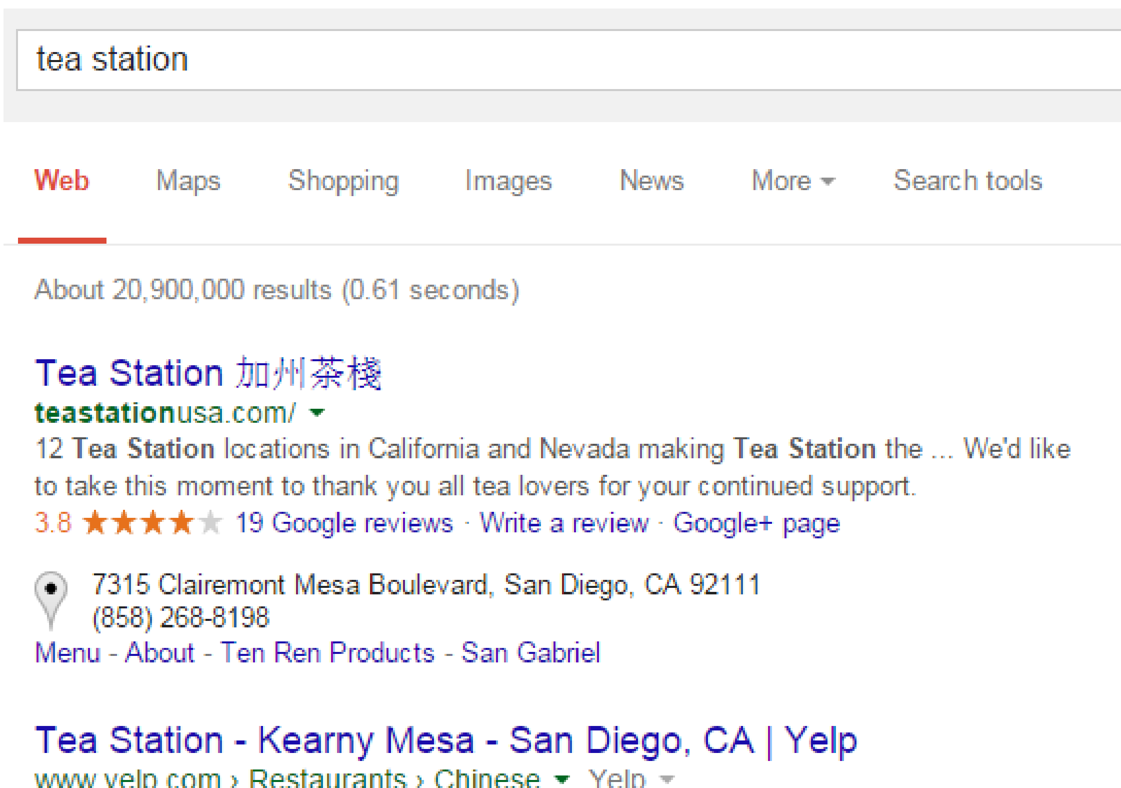
Adapts the **regression** approaches we saw last week to binary problems

3. Support Vector Machines

Learns to classify items by finding a hyperplane that separates them

This week…

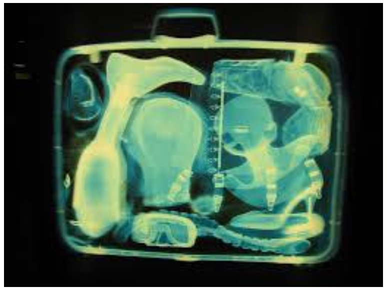
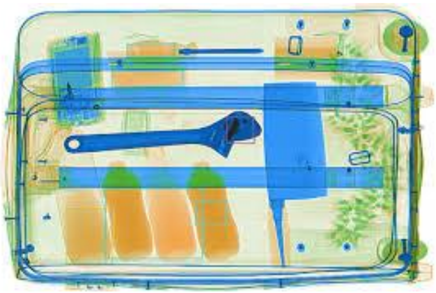
Ranking results in order of how likely they are to be relevant



This week…

Evaluating classifiers

- False positives are nuisances but false negatives are disastrous (or vice versa)
- Some classes are very rare
- When we only care about the "most confident" predictions

e.g. which of these bags contains a weapon?

Naïve Bayes

We want to associate a probability with a label and its negation:

$$p(label|data)$$

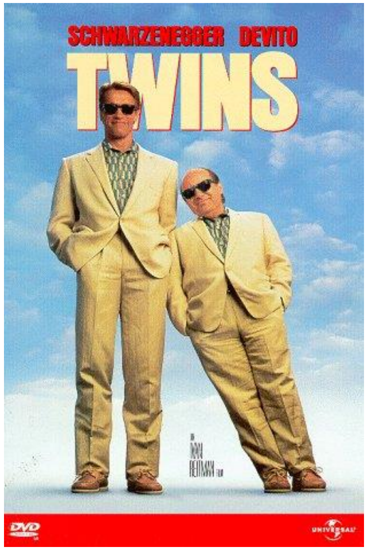
$$p(\neg label|data)$$

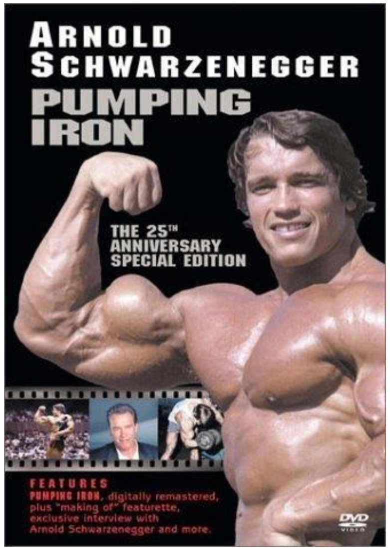
(classify according to whichever probability is greater than 0.5)

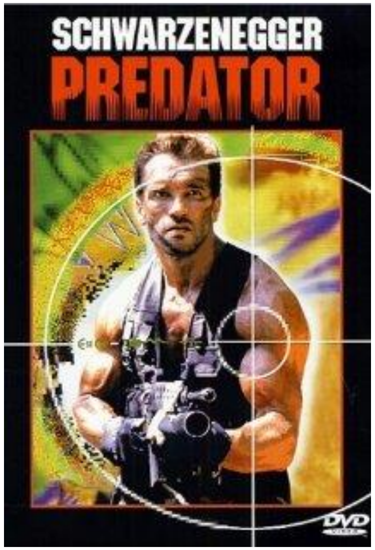
Q: How far can we get just by counting?

Naïve Bayes

e.g. p(movie is “action” | schwarzenneger in cast)







Just count!

#fims with Arnold = 45

#**action** films with Arnold = 32

p(movie is “action” | schwarzenneger in cast) = 32/45

Naïve Bayes

What about:

p(movie is “action” |
schwarzenneger in cast **and**
release year = 2015 **and**
mpaa rating = PG **and**
budget < $1000000
)

#(training) fims with Arnold, released in 2015, rated PG, with a budged below $1M = 0

#(training) action fims with Arnold, released in 2015, rated PG, with a budged below $1M = 0

Naïve Bayes

Q: If we've never seen this combination of features before, what can we conclude about their probability?

A: We need some **simplifying assumption** in order to associate a probability with this feature combination

Naïve Bayes

Naïve Bayes assumes that features are **conditionally independent** given the label

$$(feature_i \perp\!\!\!\perp feature_j | label)$$

Conditional independence?

$$(a \perp\!\!\!\perp b | c)$$

(a is conditionally independent of b, given c)

"if you know **c**, then knowing **a** provides no additional information about **b**"

$$(\text{I remembered my umbrella} \perp\!\!\!\perp \text{the streets are wet} \mid \text{it's raining})$$

Naïve Bayes

$$(feature_i \perp\!\!\!\perp feature_j | label)$$

$$p(f_i \wedge f_2 \,|\, label)$$

$$= p(f_i \,|\, label)\, p(f_j \,|\, label)$$

Naïve Bayes

posterior

prior

likelihood

$$p(label|features) = \frac{p(label)\, p(features|label)}{p(features)}$$

evidence

$> 0.5 \rightarrow T$

$\leq 0.5 \rightarrow F$

$$= \frac{p(label) \prod_i p(f_i|label)}{p(features)}$$

Naïve Bayes

$$p(label|features) = \frac{p(label)\prod_i p(feature_i|label)}{\underbrace{p(features)}_{?}}$$

$$\frac{p(label|features)}{p(\neg label|features)} = \frac{p(label)\prod_i p(f_i|label)}{p(\neg label)\prod_i p(f_i|\neg label)}$$

The denominator doesn't matter, because we really just care about

$$p(label|features) \quad \text{vs.} \quad p(\neg label|features)$$

both of which have the same denominator

Naïve Bayes

The denominator doesn't matter, because we really just care about

$$p(label|features) \quad \text{vs.} \quad p(\neg label|features)$$

both of which have the same denominator

Example 1

Amazon editorial descriptions:

Amazon.com Review

For most children, summer vacation is something to look forward to. But not for our 13-year-ol
uncle, and cousin who detest him. The third book in J.K. Rowling's Harry Potter series catapults
Dursleys' dreadful visitor Aunt Marge to inflate like a monstrous balloon and drift up to the ceili
(and from officials at Hogwarts School of Witchcraft and Wizardry who strictly forbid students t
out into the darkness with his heavy trunk and his owl Hedwig.

As it turns out, Harry isn't punished at all for his errant wizardry. Instead he is mysteriously res
triple-decker, violently purple bus to spend the remaining weeks of summer in a friendly inn ca
his third year at Hogwarts explains why the officials let him off easily. It seems that Sirius Blac
loose. Not only that, but he's after Harry Potter. But why? And why do the Dementors, the guar
are unaffected? Once again, Rowling has created a mystery that will have children and adults cl
Fortunately, there are four more in the works. (Ages 9 and older) --*Karin Snelson* --*This text re*

50k descriptions:

http://jmcauley.ucsd.edu/cse255/data/amazon/book_descriptions_50000.json

Example 1

P(book is a children's book |
"wizard" is mentioned in the description **and**
"witch" is mentioned in the description)

Code available on:

http://jmcauley.ucsd.edu/cse255/code/week2.py

Example 1

Conditional independence assumption:

"if you know **a book is for children**, then knowing that **wizards are mentioned** provides no additional information about whether **witches are mentioned**"

obviously ridiculous

Double-counting

Q: What would happen if we trained two regressors, and attempted to "naively" combine their parameters?

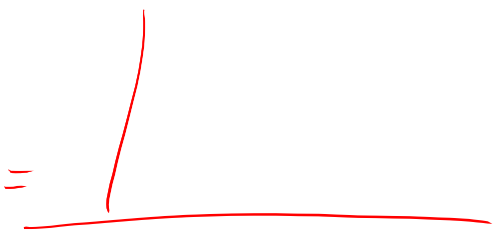
$$\text{height(cm)} = \theta_1 \times \text{weight(kg)}$$
$$220 = 1.1 \times \text{weight}$$

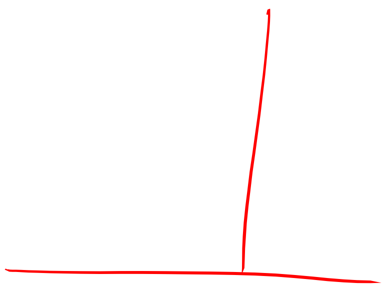
$$\text{height} = \theta_2 \times \text{shoe size}$$
$$= 20 \times \text{shoe size}$$

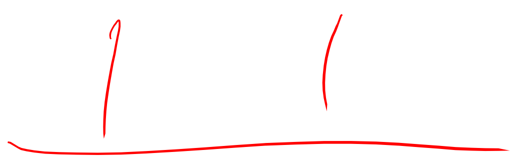
$$\text{height} = 1.1 \times \text{weight} + 20 \times \text{shoe size}$$

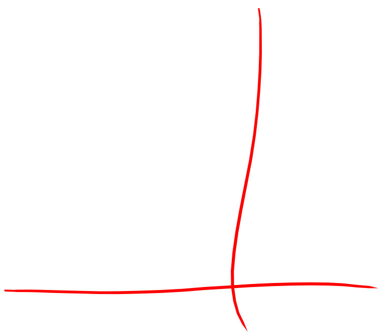
Double-counting

$h = \theta_1 \times w + \theta_2 \times \text{shoe}$

$m_1 =$ 

m_2 

$m_{1+2} =$ 

m'_{1+2} 

Double-counting

A: Since both features encode essentially the same information, we'll end up **double-counting** their effect

Logistic regression

Logistic Regression also aims to model

$$p(label|data)$$

By training a classifier of the form

$$y_i = \begin{cases} 1 & \text{if } X_i \cdot \theta > 0 \\ 0 & \text{otherwise} \end{cases}$$

Logistic regression

Last week: regression

$$y_i = X_i \cdot \theta$$

This week: logistic regression

$$y_i = \begin{cases} 1 & \text{if } X_i \cdot \theta > 0 \\ 0 & \text{otherwise} \end{cases}$$

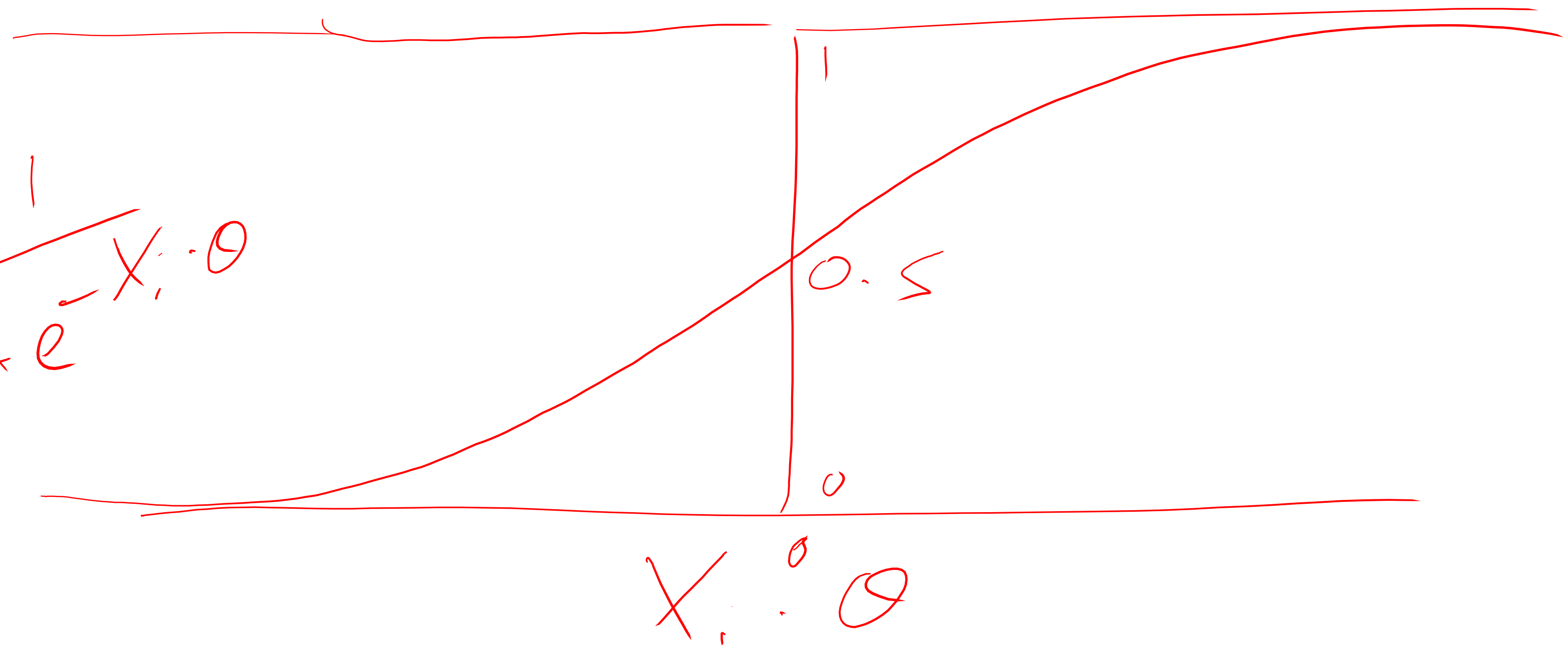
Logistic regression

Q: How to convert a real-valued expression ($X_i \cdot \theta \in \mathbb{R}$)
Into a probability
($p_\theta(y_i|X_i) \in [0, 1]$)

Logistic regression

A: sigmoid function: $\sigma(t) = \frac{1}{1+e^{-t}}$

$$\frac{1}{1+e^{-X_i \cdot \theta}}$$



Logistic regression

Training:

$X_i \cdot \theta$ should be maximized when y_i is positive and minimized when y_i is negative

$$\arg\max_\theta \underbrace{\prod_{y_i=1} \sigma(X_i \cdot \theta)}_{p(y_i \mid X_i)} \underbrace{\prod_{y_i=0} (1-\sigma(X_i \cdot \theta))}_{p(\neg y_i \mid X_i)}$$

Logistic regression

How to optimize?

$$L_\theta(y|X) = \prod_{y_i=1} p_\theta(y_i|X_i) \prod_{y_i=0} (1 - p_\theta(y_i|X_i))$$

- Take logarithm
- **Subtract** regularizer
- Compute gradient
- Solve using gradient **ascent**

(solve on blackboard)

Logistic regression

$$L_\theta(y|X) = \prod_{y_i=1} p_\theta(y_i|X_i) \prod_{y_i=0} (1 - p_\theta(y_i|X_i))$$

$$\sum_{y_i=1} \log\left(\frac{1}{1+e^{-X_i\cdot\theta}}\right) + \sum_{y_i=0} \log\left(\frac{e^{-X_i\cdot\theta}}{1+e^{-X_i\cdot\theta}}\right)$$

$$\sum_{y_i} -\log\left(1+e^{-X_i\cdot\theta}\right) + \sum_{y_i=0} -X_i\cdot\theta$$

$$-\lambda\|\theta\|_2^2$$

Logistic regression

$$l_\theta(y|X) = \sum_i -\log(1+e^{-X_i\cdot\theta}) + \sum_{y_i=0} -X_i\cdot\theta - \lambda\|\theta\|_2^2$$

$$\frac{\partial l}{\partial \theta_k} = \sum_{y_i} \underbrace{\frac{X_{ik}e^{-X_i\cdot\theta}}{1+e^{-X_i\cdot\theta}}}_{X_{ik}(1-\sigma(X_i\cdot\theta))} + \sum_{y_i=0} -X_{ik} - 2\lambda\theta_k$$

Multiclass classification

The most common way to generalize **binary** classification (output in {0,1}) to **multiclass** classification (output in {1 … N}) is simply to train a binary predictor for each class

e.g. based on the description of this book:

- Is it a Children’s book? {yes, no}
- Is it a Romance? {yes, no}
- Is it Science Fiction? {yes, no}
- …

In the event that predictions are inconsistent, choose the one with the highest confidence

Questions?

Further reading:

- On Discriminative vs. Generative classifiers: A comparison of logistic regression and naïve Bayes (Ng & Jordan '01)
- Boyd-Fletcher-Goldfarb-Shanno algorithm (BFGS)

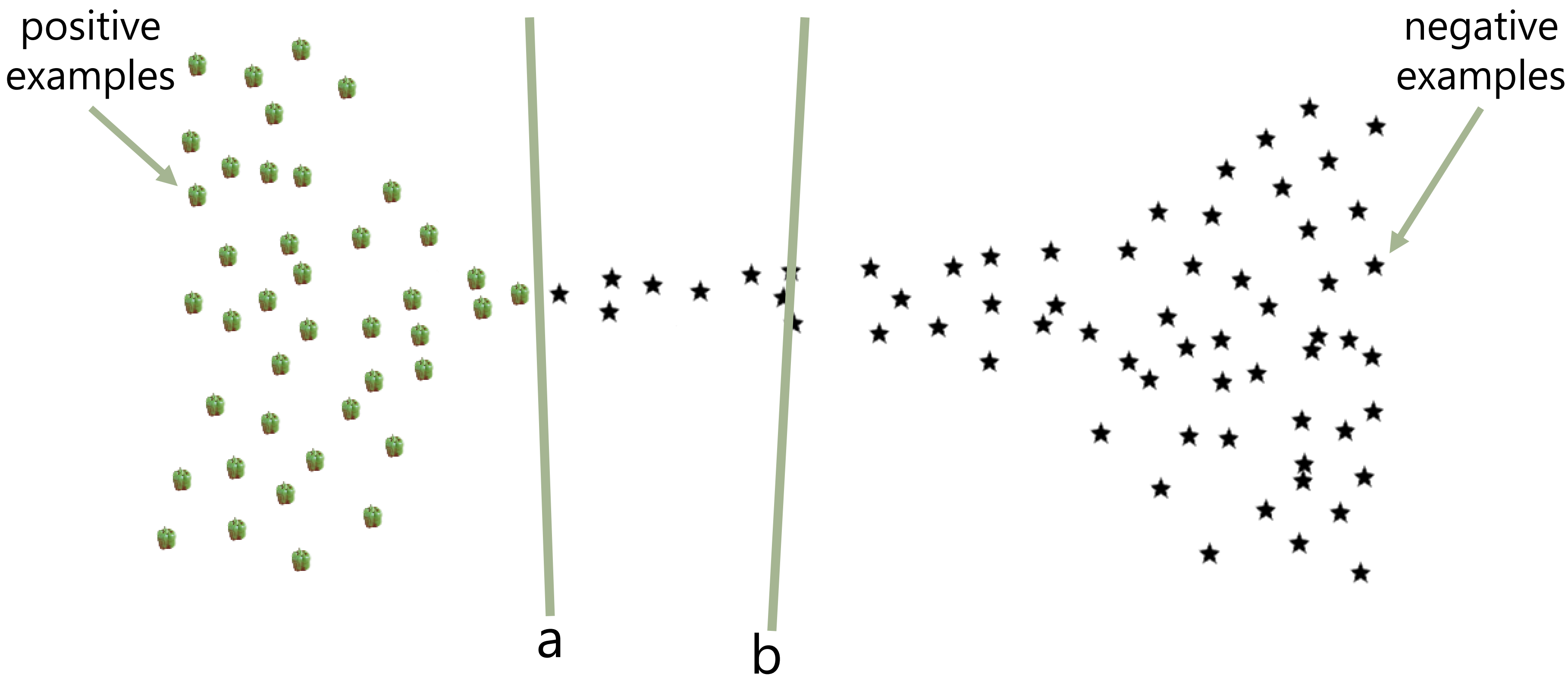
CSE 255 – Lecture 3

Data Mining and Predictive Analytics

Supervised learning – SVMs

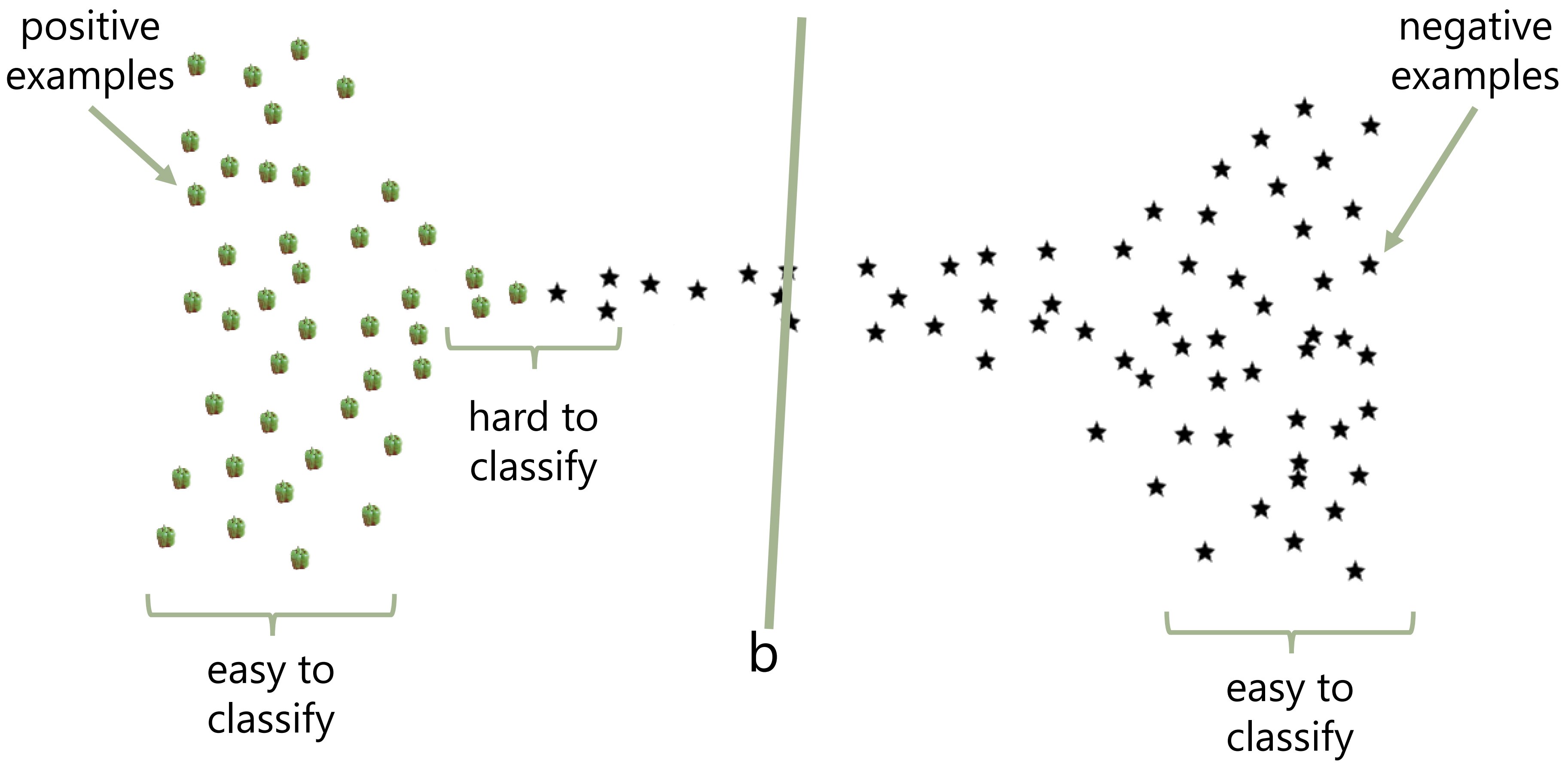
Logistic regression

Q: Where would a logistic regressor place the decision boundary for these features?



Logistic regression

Q: Where would a logistic regressor place the decision boundary for these features?



Logistic regression

- Logistic regressors don't optimize the number of "mistakes"
- No special attention is paid to the "difficult" instances – every instance influences the model
- But "easy" instances can affect the model (and in a bad way!)
- How can we develop a classifier that optimizes the number of mislabeled examples?

Support Vector Machines

This is essentially the intuition behind Support Vector Machines (SVMs) – train a classifier that focuses on the "difficult" examples by minimizing the misclassification error

We still want a classifier of the form

$$y_i = \begin{cases} 1 & \text{if } X_i \cdot \theta - \alpha > 0 \\ -1 & \text{otherwise} \end{cases}$$

But we want to minimize the number of misclassifications:

$$\arg\min_\theta \sum_i \delta(y_i(X_i \cdot \theta - \alpha) \leq 0)$$

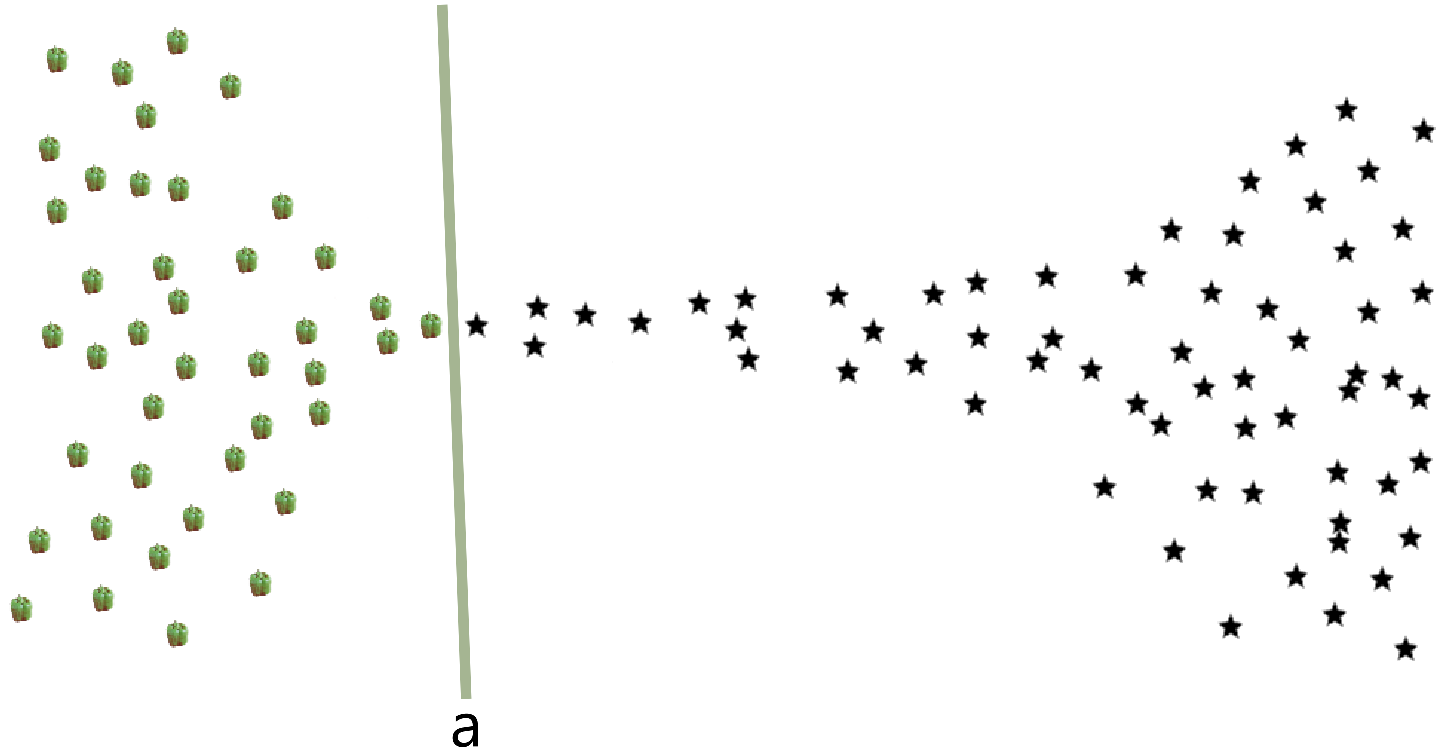
1 iff argument is true

Support Vector Machines

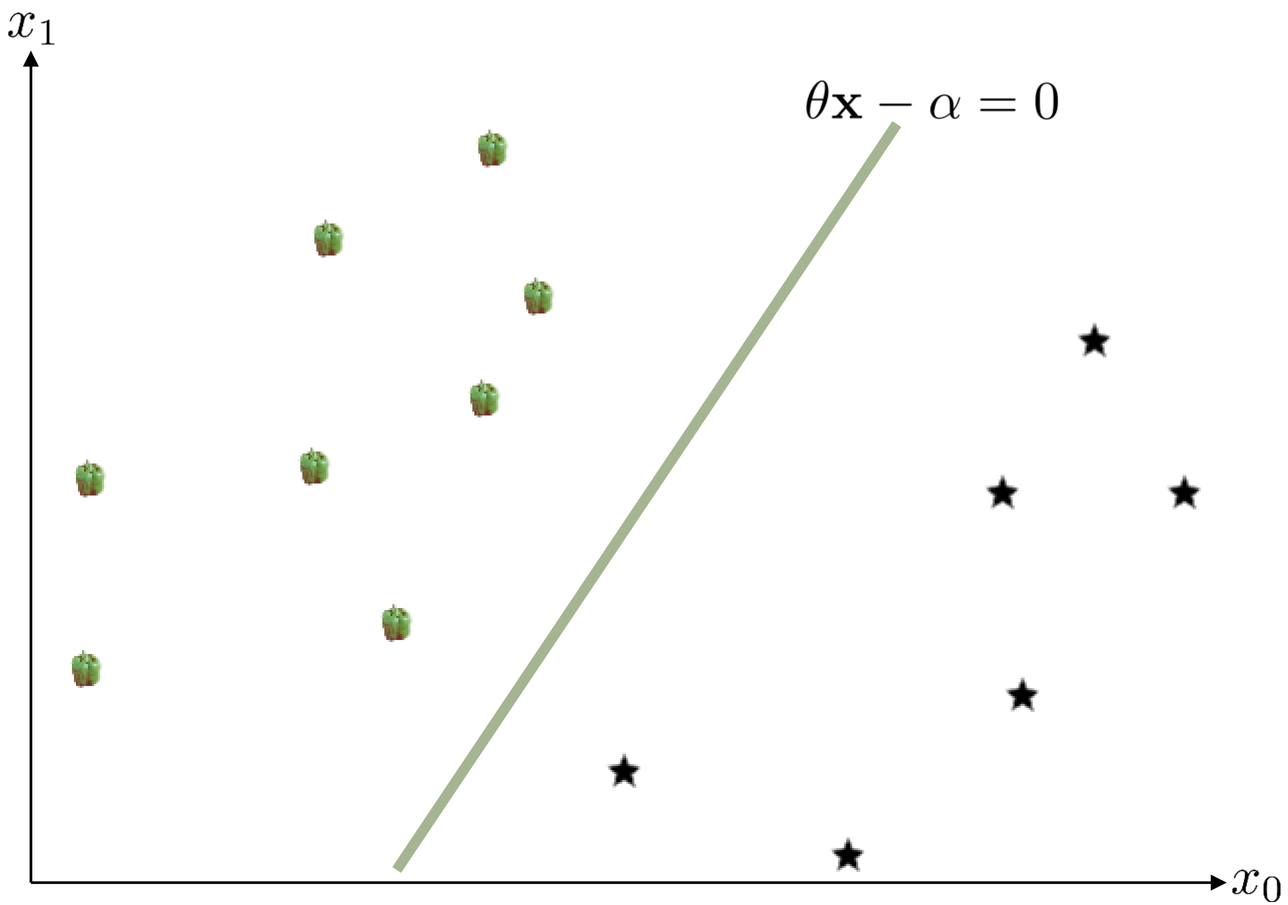
$$\arg\min_\theta \sum_i \delta(y_i(X_i \cdot \theta - \alpha) \leq 0)$$

Support Vector Machines

Simple (seperable) case: there exists a perfect classifier

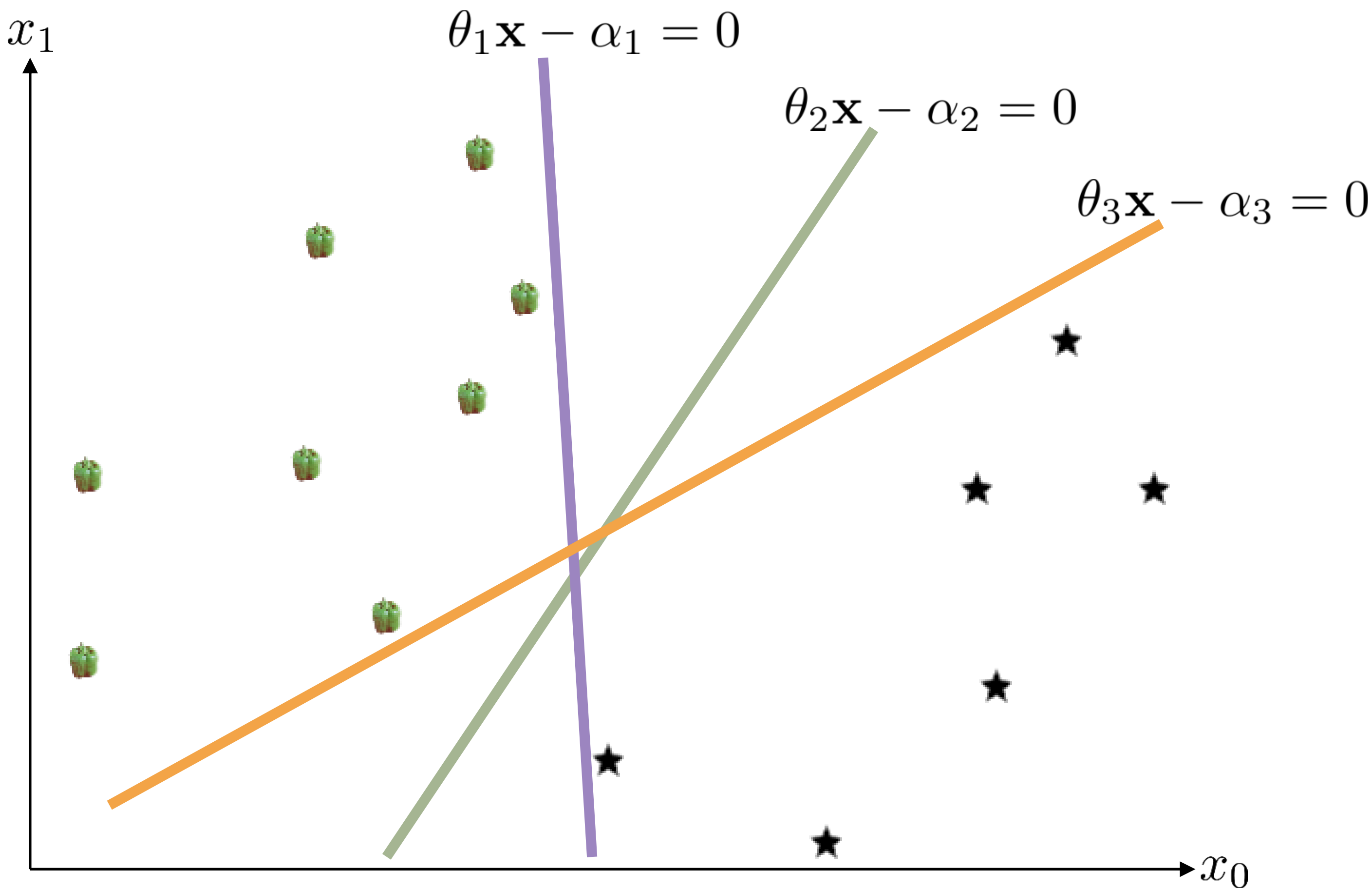


Support Vector Machines



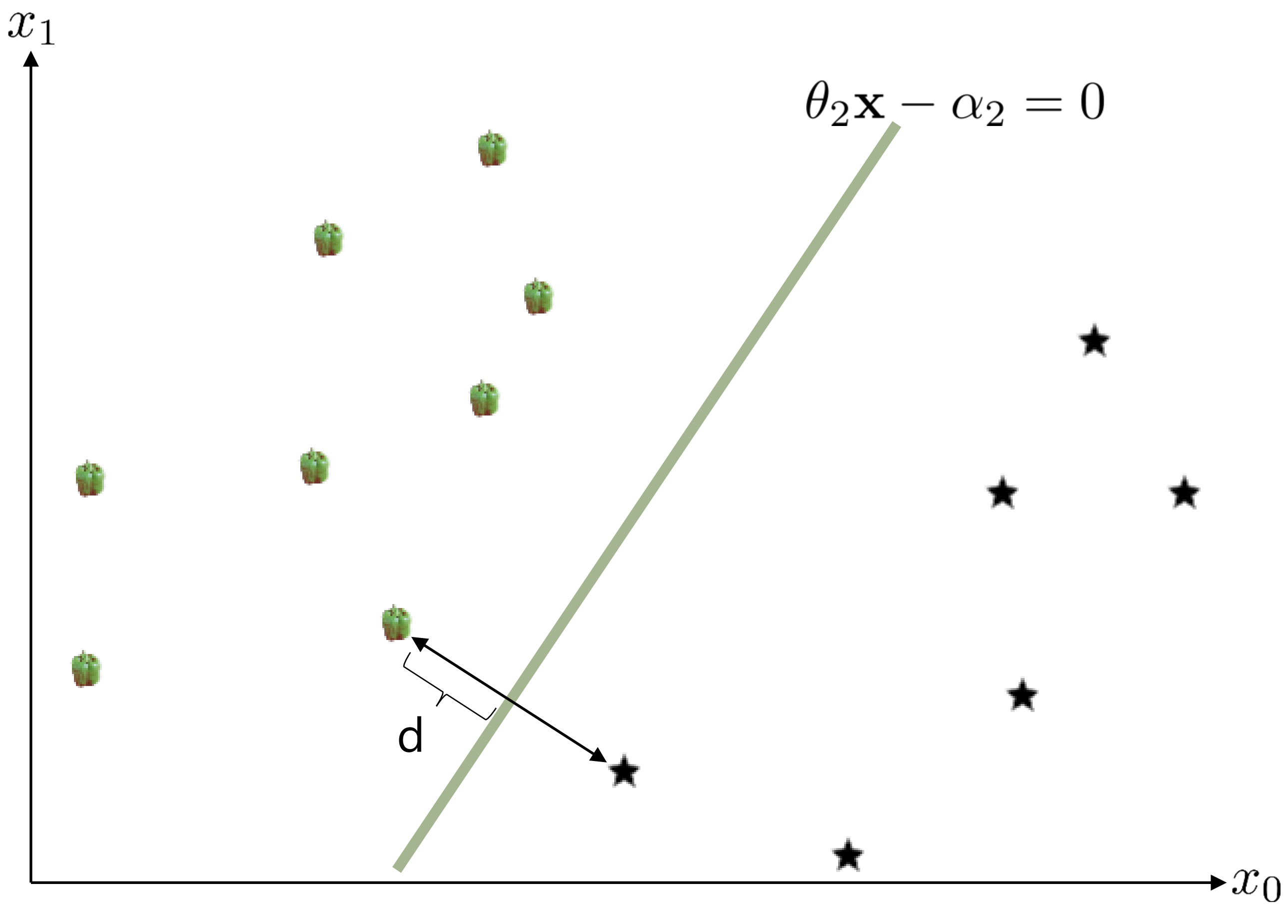
The classifier is defined by the hyperplane $\theta\mathbf{x} - \alpha = 0$

Support Vector Machines



Q: Is one of these classifiers preferable over the others?

Support Vector Machines



A: Choose the classifier that maximizes the distance to the nearest point

Support Vector Machines

Distance from a point to a line?

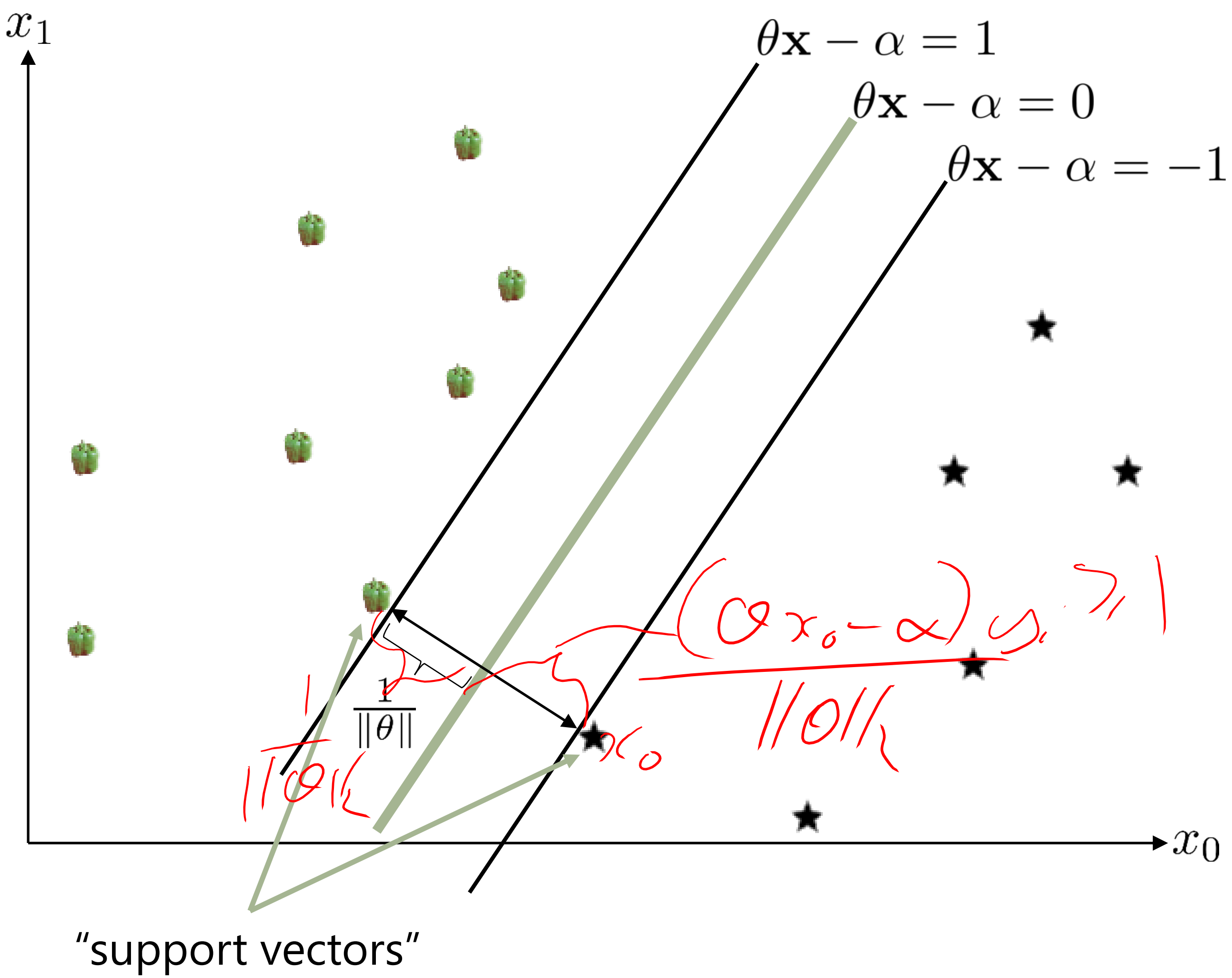
$$ax + by + c = 0 \qquad x_0, y_0$$

$$\frac{|a x_0 - b y_0 + c|}{\sqrt{a^2 + b^2}}$$

$$\theta x - \alpha = 0 \qquad \underline{x_0}$$

$$\frac{|\theta x_0 - \alpha|}{\|\theta\|_2}$$

Support Vector Machines



$$\arg\min_{\theta,\alpha} \frac{1}{2}\|\theta\|_2^2$$

such that

$$\forall_i y_i(\theta \cdot X_i - \alpha) \geq 1$$

Support Vector Machines

This is known as a "quadratic program" (QP) and can be solved using "standard" techniques

$$\arg\min_{\theta,\alpha} \frac{1}{2}\|\theta\|_2^2$$

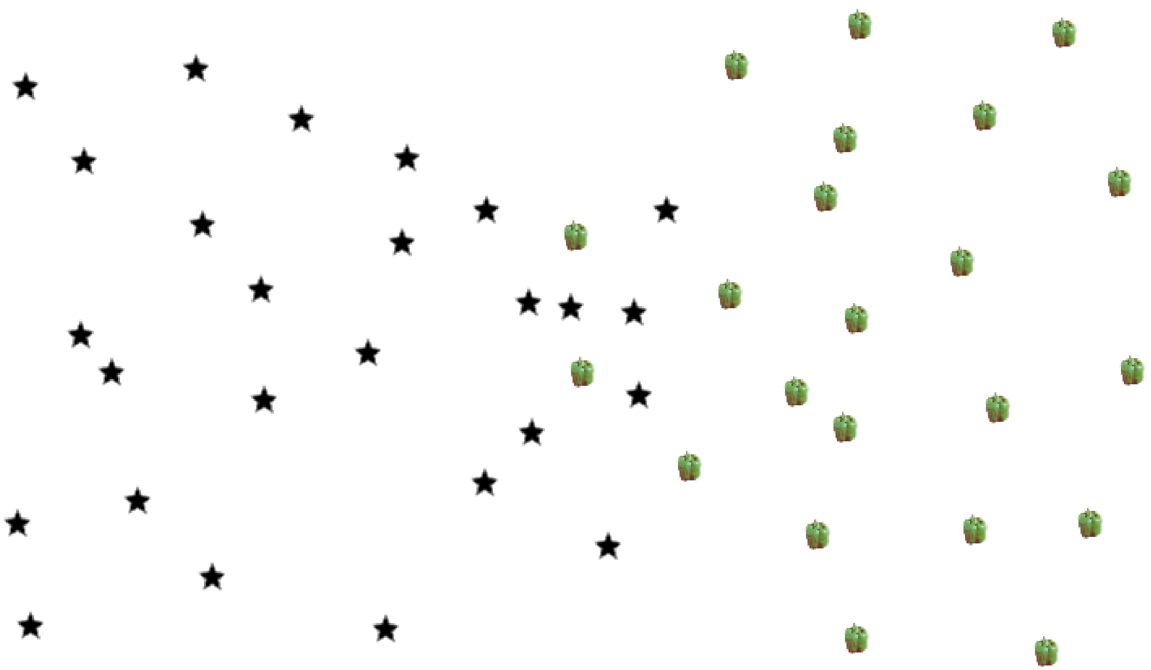
such that

$$\forall_i y_i(\theta \cdot X_i - \alpha) \geq 1$$

See e.g. Nocedal & Wright ("Numerical Optimization"), 2006

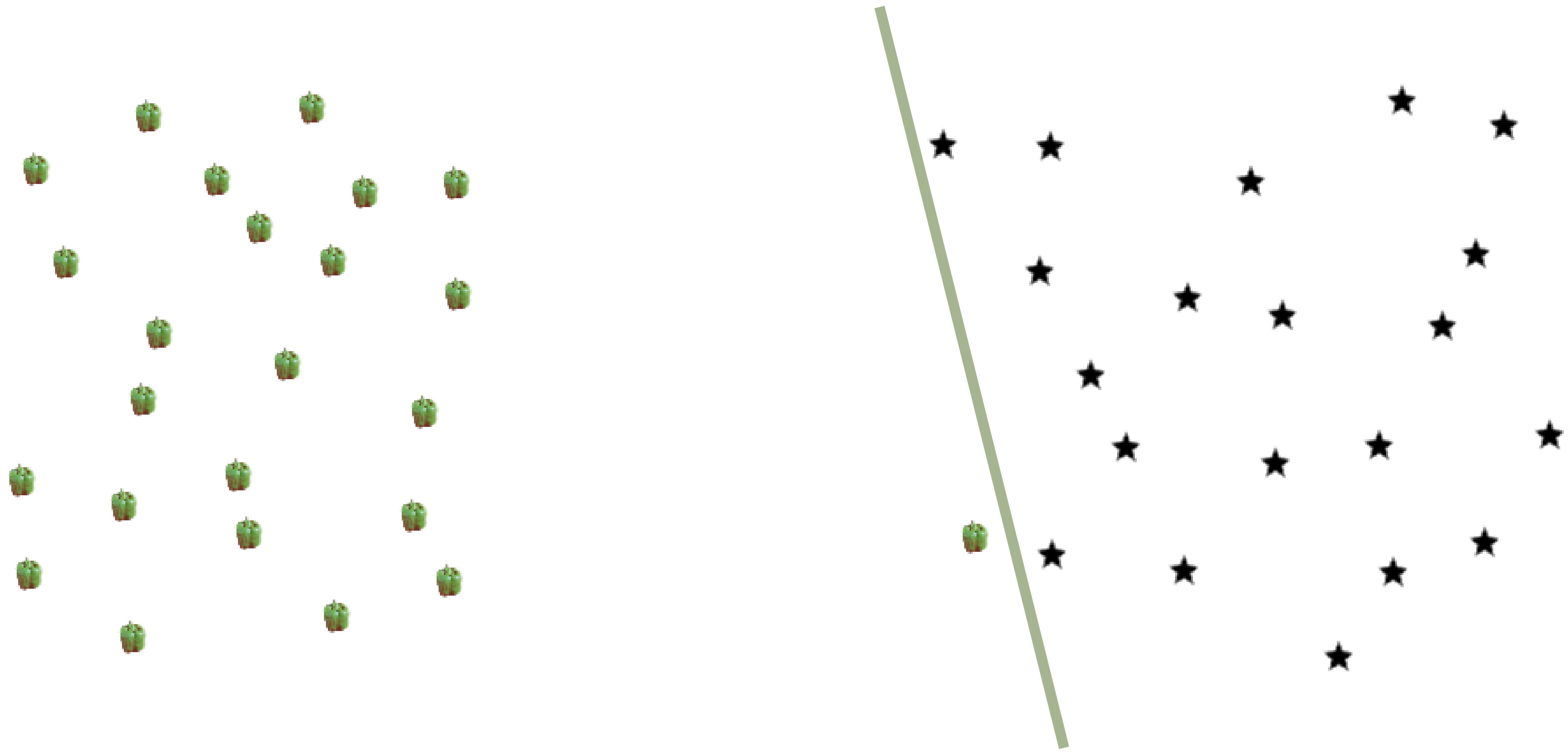
Support Vector Machines

But: is finding such a separating hyperplane even possible?



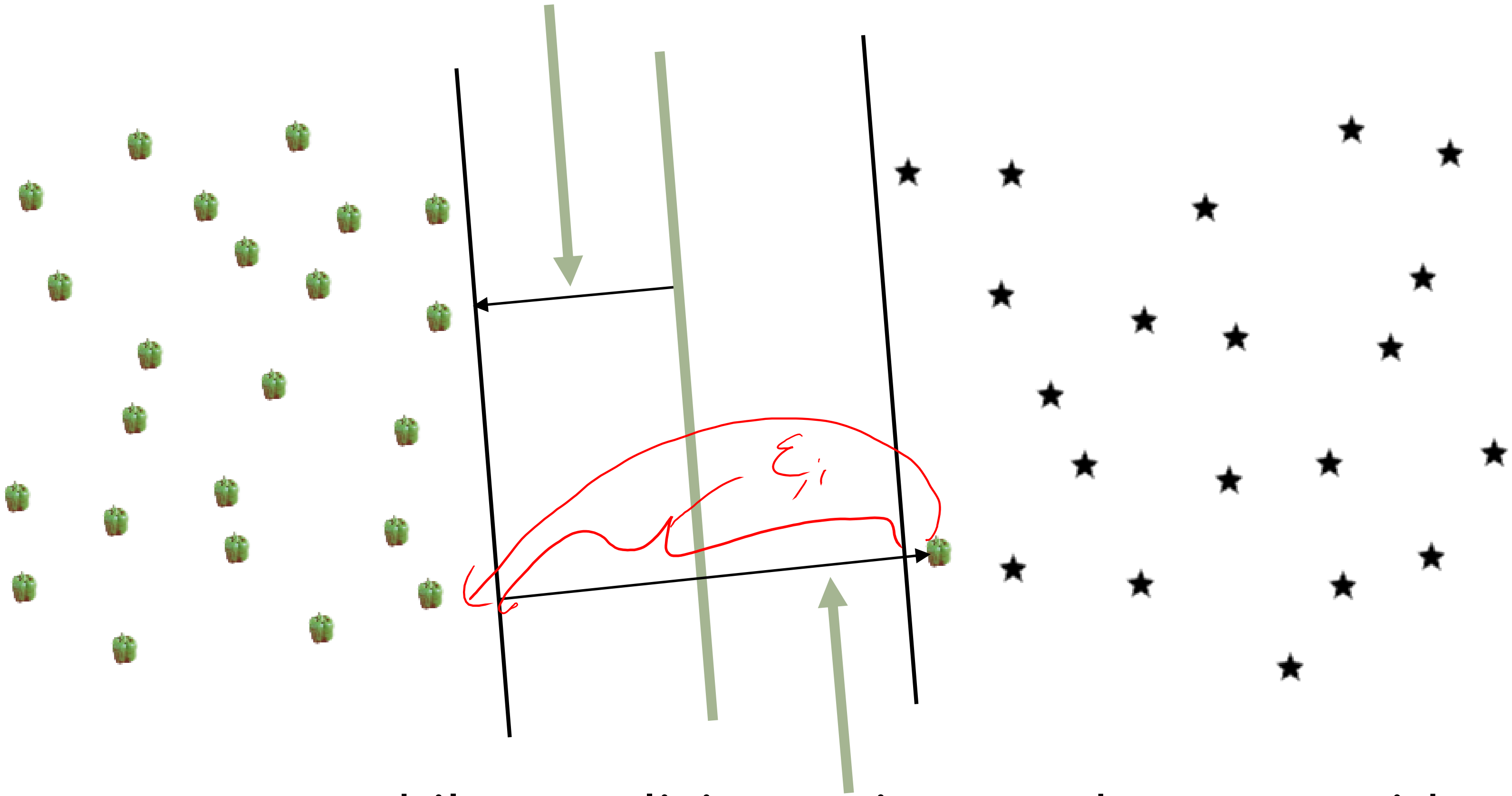
Support Vector Machines

Or: is it actually a good idea?



Support Vector Machines

Want the margin to be as wide as possible



While penalizing points on the wrong side of it

Support Vector Machines

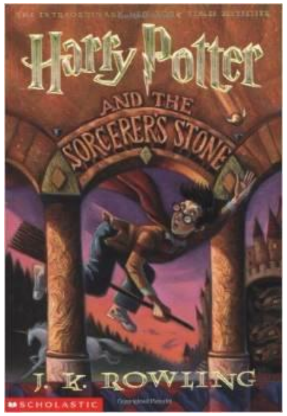
Soft-margin formulation:

$$\arg\min_{\theta,\alpha,\ \xi_i \geq 0} \frac{1}{2}\|\theta\|_2^2 + \sum_i \xi_i$$

such that

$$\forall_i y_i(\theta \cdot X_i - \alpha) \geq 1 - \xi_i$$

Judging a book by its cover



[0.723845, 0.153926, 0.757238, 0.983643, ...]

$\underbrace{\hphantom{0.723845, 0.153926, 0.757238, 0.983643, ...}}_{\text{4096-dimensional image features}}$

Images features are available for each book on
http://jmcauley.ucsd.edu/cse255/data/amazon/book_images_5000.json

http://caffe.berkeleyvision.org/

Judging a book by its cover

Example: train an SVM to predict whether a book is a children's book from its cover art

(code available on)
http://jmcauley.ucsd.edu/cse255/code/week2.py

Judging a book by its cover

- The number of errors we made was extremely low, yet our classifier doesn't seem to be very good – why?

(stay tuned next lecture!)

Summary

The classifiers we've seen today all attempt to make decisions by associating weights (theta) with features (x) and classifying according to

$$y_i = \begin{cases} 1 & \text{if } X_i \cdot \theta > 0 \\ 0 & \text{otherwise} \end{cases}$$

Summary

- **Naïve Bayes**
  - Probabilistic model (fits $p(label|data)$)
  - Makes a conditional independence assumption of the form $(feature_i \perp\!\!\!\perp feature_j | label)$ allowing us to define the model by computing $p(feature_i | label)$ for each feature
  - Simple to compute just by counting
- **Logistic Regression**
  - Fixes the "double counting" problem present in naïve Bayes
- **SVMs**
  - Non-probabilistic: optimizes the classification error rather than the likelihood

Questions?