CS388: Natural Language Processing

Lecture 3: Multiclass Classification

Greg Durrett

TEXAS
The University of Texas at Austin

Some slides adapted from Vivek Srikumar, University of Utah

"Now! ... That should clear up a few things around here!"

Administrivia

- Course enrollment
- Mini 1 due Thursday at midnight (submit writeup on Gradescope + code/output on Canvas)

Recall: Feature Extraction

O	O	PER	O	PER	O	O	O	O	O
On	*Sunday,*	*Thomas*	*and*	*Mary*	*went*	*to*	*the*	*farmer's*	*market*
i = *0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*

- Feature extractor: function from (sentence, position) => sparse feature vector describing that position in the sentence
  - "Current word": what is the word at this index?
  - "Previous word": what is the word that precedes the index?

	[currWord=*Thomas*]	[currWord=*Mary*]	[prevWord = *and*]	
$f(x, i{=}4) = [$	0	1	1	…

- Feature vector only has 2 nonzero entries out of 10k+ possible
- All features coexist in the same space! Other feats (char level, …) possible

Recall: Binary Classification

Logistic regression: $P(y=1|x) = \frac{\exp\left(\sum_{i=1}^{n} w_i x_i\right)}{\left(1+\exp\left(\sum_{i=1}^{n} w_i x_i\right)\right)}$ these sums are sparse!

Decision rule: $P(y=1|x) \geq 0.5 \Leftrightarrow w^\top x \geq 0$

Gradient: differentiate the log likelihood: $x(y - P(y=1|x))$

- This is the gradient of a single example. Can then apply stochastic gradient (or related optimization methods like Adagrad, etc.)
- ML pipeline: input -> feature representation, train model on labeled data (with stochastic gradient methods), then test on new data

This Lecture

- Sentiment analysis
- Multiclass fundamentals
- Feature extraction
- Multiclass logistic regression

Sentiment Analysis

Sentiment Analysis

this movie was great! would watch again +

the movie was gross and overwrought, but I liked it +

this movie was not really very enjoyable —

- Bag-of-words doesn't seem sufficient (discourse structure, negation)
- There are some ways around this: extract bigram feature for "*not* X" for all X following the *not*

Bo Pang, Lillian Lee, Shivakumar Vaithyanathan (2002)

Sentiment Analysis

	Features	# of features	frequency or presence?	NB	ME	SVM
(1)	unigrams	16165	freq.	**78.7**	N/A	72.8
(2)	unigrams	"	pres.	81.0	80.4	**82.9**
(3)	unigrams+bigrams	32330	pres.	80.6	80.8	**82.7**
(4)	bigrams	16165	pres.	77.3	**77.4**	77.1
(5)	unigrams+POS	16695	pres.	81.5	80.4	**81.9**
(6)	adjectives	2633	pres.	77.0	**77.7**	75.1
(7)	top 2633 unigrams	2633	pres.	80.3	81.0	**81.4**
(8)	unigrams+position	22430	pres.	81.0	80.1	**81.6**

- Simple feature sets can do pretty well!

Bo Pang, Lillian Lee, Shivakumar Vaithyanathan (2002)

Sentiment Analysis

Method	RT-s	MPQA
MNB-uni	77.9	85.3
MNB-bi	**79.0**	**86.3**
SVM-uni	76.2	86.1
SVM-bi	77.7	**86.7**
NBSVM-uni	**78.1**	85.3
NBSVM-bi	**79.4**	**86.3**
RAE	76.8	85.7
RAE-pretrain	77.7	**86.4**
Voting-w/Rev.	63.1	81.7
Rule	62.9	81.8
BoF-noDic.	75.7	81.8
BoF-w/Rev.	76.4	84.1
Tree-CRF	77.3	86.1
BoWSVM	–	–
Kim (2014) CNNs	**81.5**	**89.5**

Naive Bayes is doing well!

Ng and Jordan (2002) — NB can be better for small data

Before neural nets had taken off — results weren't that great

Wang and Manning (2012)

Sentiment Analysis

- Stanford Sentiment Treebank (SST) binary classification
- Best systems now: large pretrained networks
- 90 -> 97 over the last 2 years

Model	Accuracy	Paper / Source	Code
XLNet-Large (ensemble) (Yang et al., 2019)	96.8	XLNet: Generalized Autoregressive Pretraining for Language Understanding	Official
MT-DNN-ensemble (Liu et al., 2019)	96.5	Improving Multi-Task Deep Neural Networks via Knowledge Distillation for Natural Language Understanding	Official
Snorkel MeTaL(ensemble) (Ratner et al., 2018)	96.2	Training Complex Models with Multi-Task Weak Supervision	Official
MT-DNN (Liu et al., 2019)	95.6	Multi-Task Deep Neural Networks for Natural Language Understanding	Official
Bidirectional Encoder Representations from Transformers (Devlin et al., 2018)	94.9	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	Official
...			
Neural Semantic Encoder (Munkhdalai and Yu, 2017)	89.7	Neural Semantic Encoders	
BLSTM-2DCNN (Zhou et al., 2017)	89.5	Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling	

https://github.com/sebastianruder/NLP-progress/blob/master/english/sentiment_analysis.md

Multiclass Fundamentals

Text Classification

A Cancer Conundrum: Too Many Drug Trials, Too Few Patients

Breakthroughs in immunotherapy and a rush to develop profitable new treatments have brought a crush of clinical trials scrambling for patients.

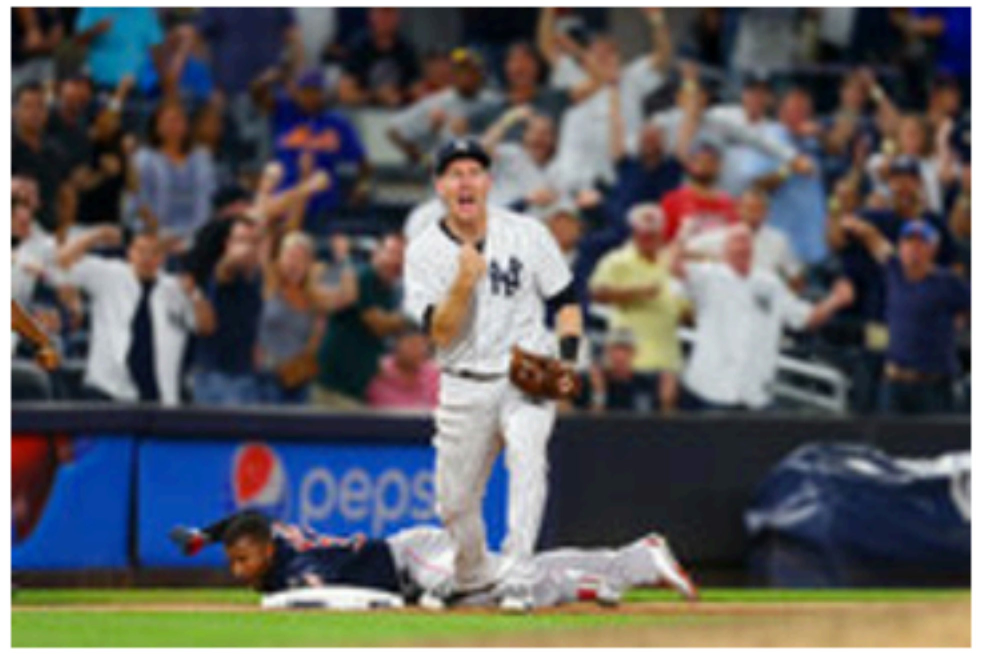
By GINA KOLATA

→ Health

Yankees and Mets Are on Opposite Tracks This Subway Series

As they meet for a four-game series, the Yankees are playing for a postseason spot, and the most the Mets can hope for is to play spoiler.

By FILIP BONDY



→ Sports

~20 classes

Image Classification

Dog

Car

- Thousands of classes (ImageNet)

Entailment

- Three-class task over sentence pairs
- Not clear how to do this with simple bag-of-words features

A soccer game with multiple males playing.

ENTAILS

Some men are playing a sport.

A black race car starts up in front of a crowd of people.

CONTRADICTS

A man is driving down a lonely road

A smiling costumed woman is holding an umbrella.

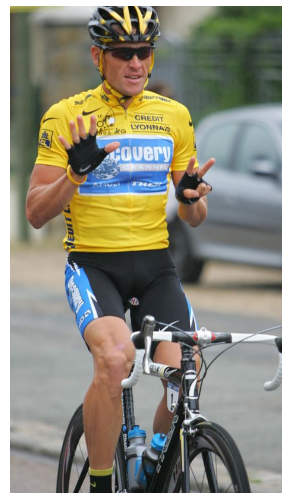
NEUTRAL

A happy woman in a fairy costume holds an umbrella.

Bowman et al. (2015)

Entity Linking

Although he originally won the event, the United States Anti-Doping Agency announced in August 2012 that they had disqualified **Armstrong** from his seven consecutive Tour de France wins from 1999–2005.



Lance Edward Armstrong is an American former professional road cyclist

Armstrong County is a county in Pennsylvania…

?

?

- 4,500,000 classes (all articles in Wikipedia)

Reading Comprehension

One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home.

3) Where did James go after he went to the grocery store?
A) his deck
B) his freezer
C) a fast food restaurant
D) his room

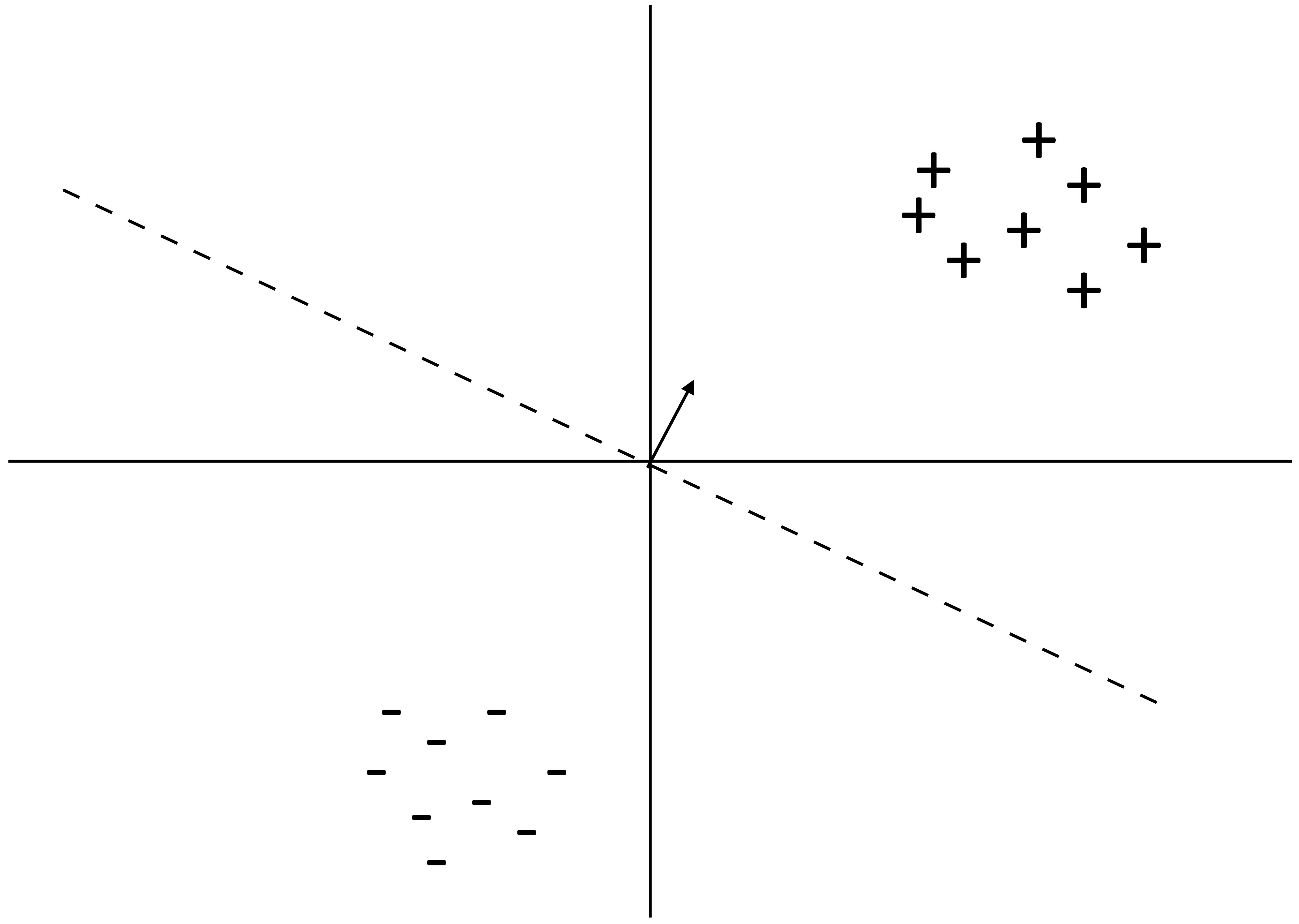
After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

- Multiple choice questions, 4 classes (but classes change per example)

Richardson (2013)

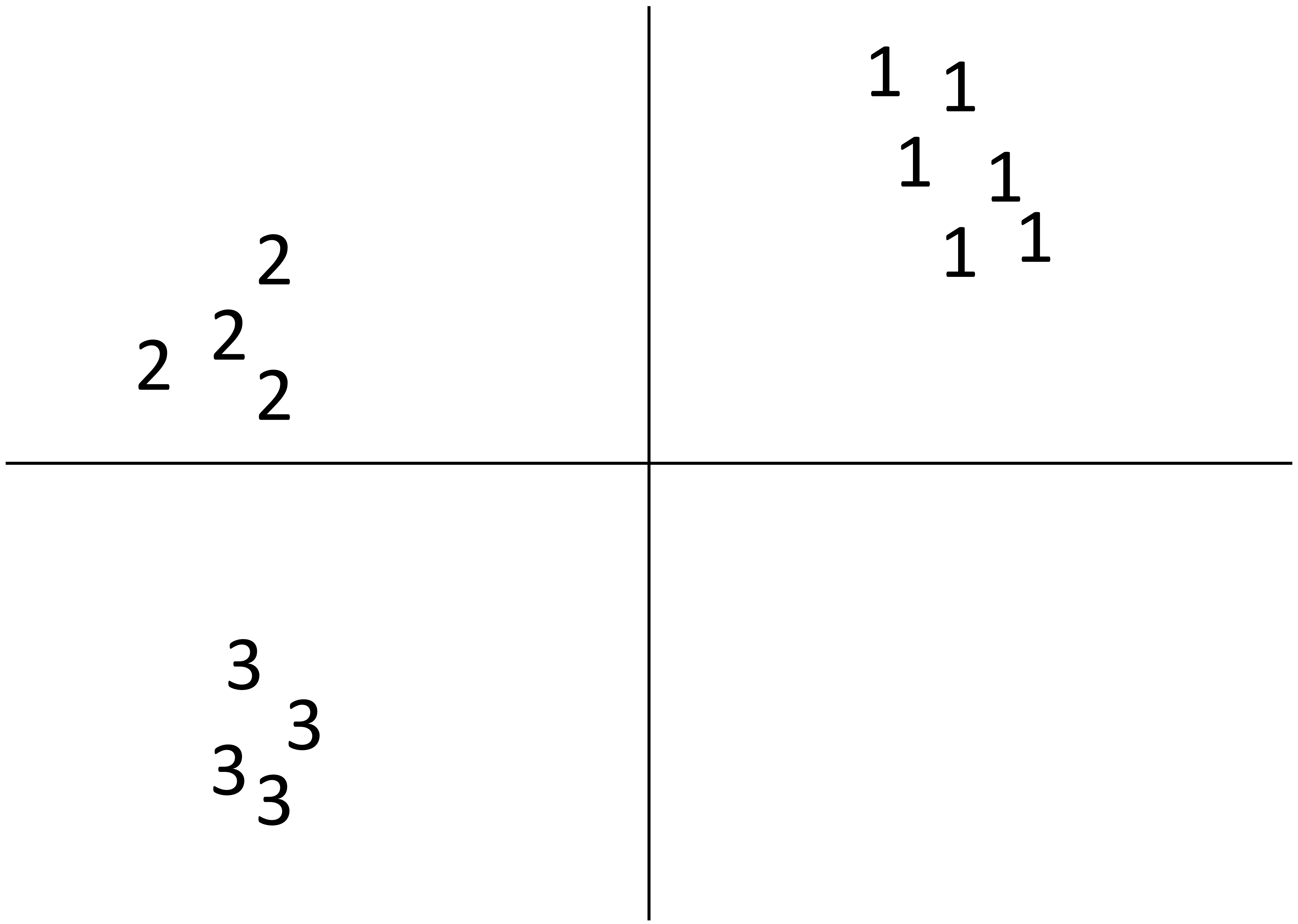
Binary Classification

- Binary classification: one weight vector defines positive and negative classes



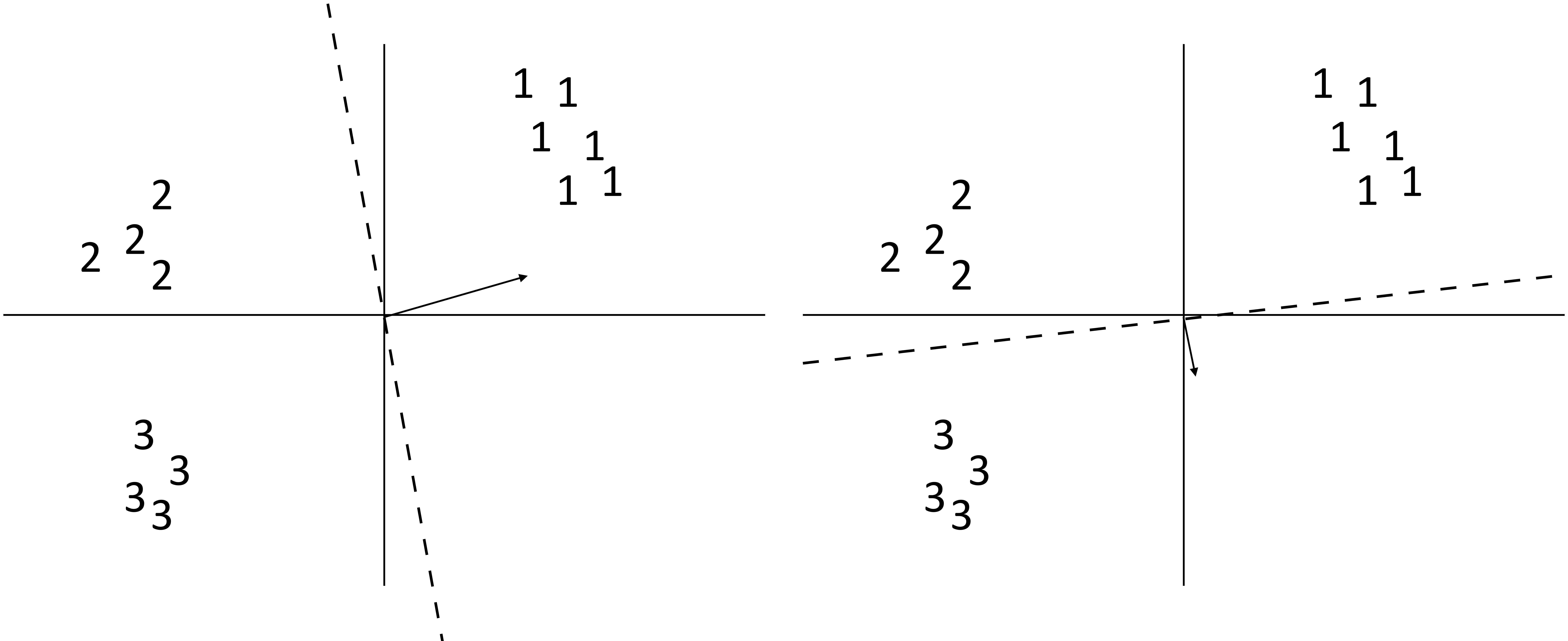
Multiclass Classification

- Can we just use binary classifiers here?



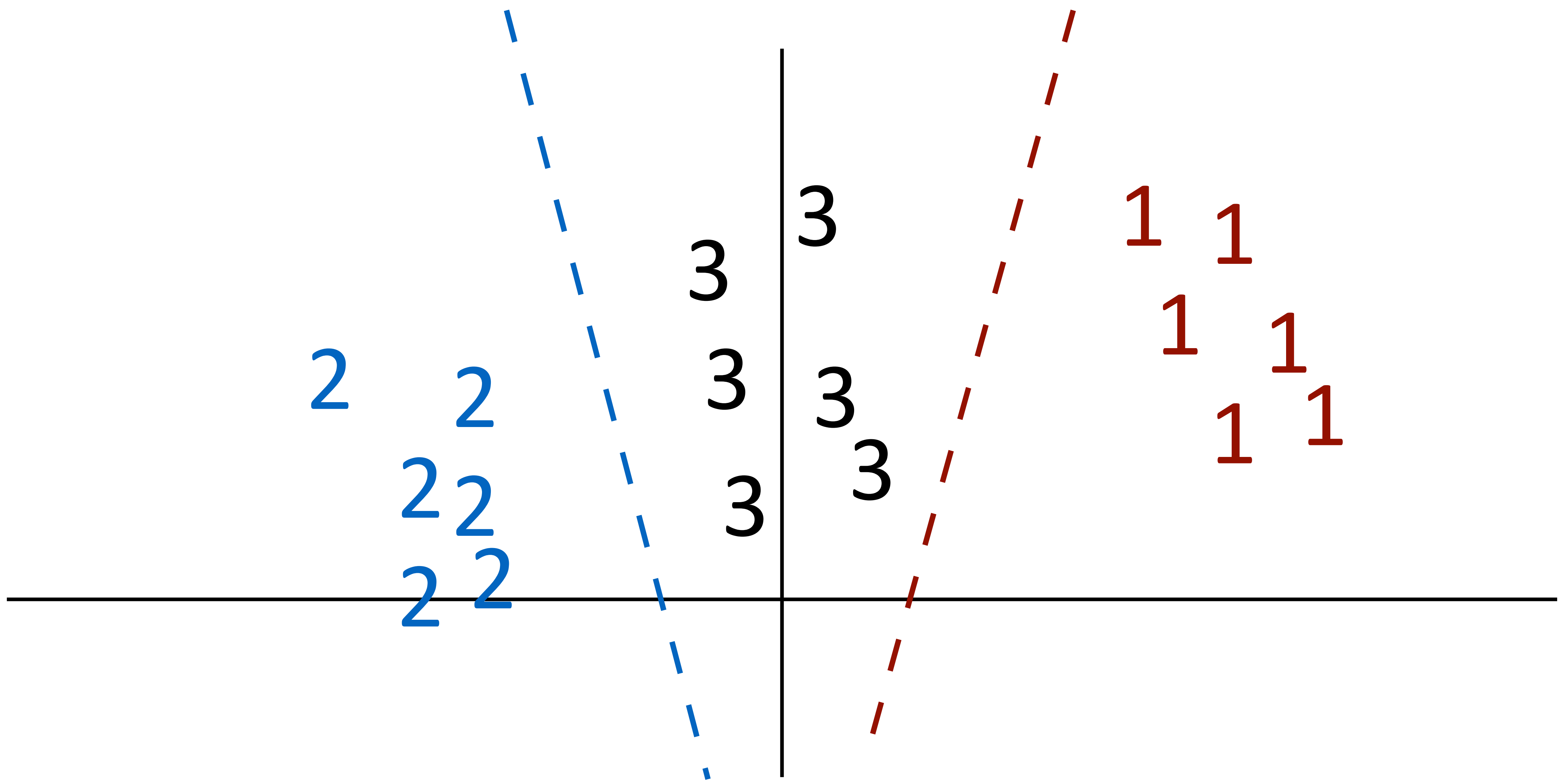
Multiclass Classification

- One-vs-all: train *k* classifiers, one to distinguish each class from all the rest
- How do we reconcile multiple positive predictions? Highest score?



Multiclass Classification

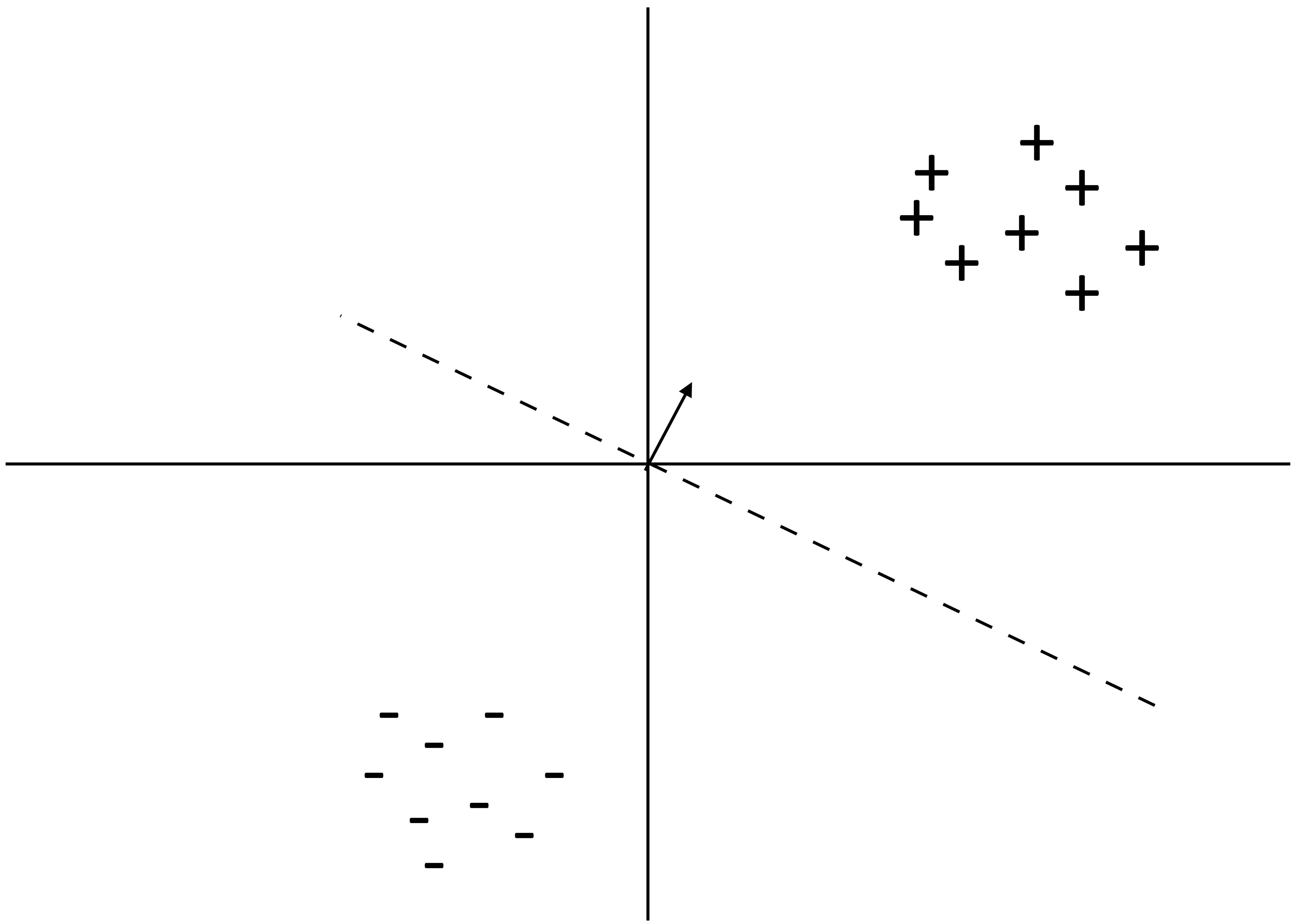
- Not all classes may even be separable using this approach



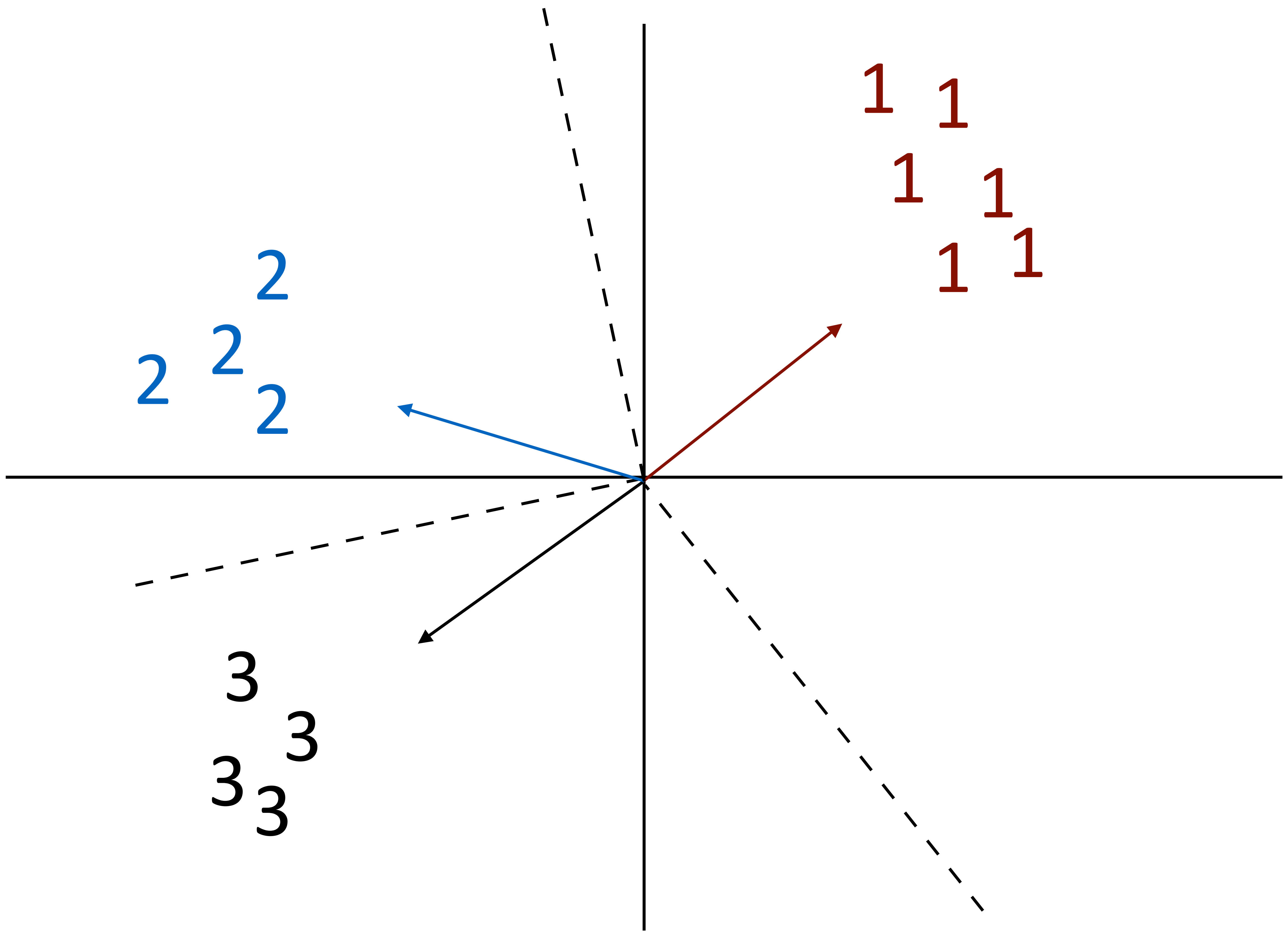
- Can separate 1 from 2+3 and 2 from 1+3 but not 3 from the others (with these features)

Multiclass Classification

- Binary classification: one weight vector defines both classes



- Multiclass classification: different weights and/or features per class



Multiclass Classification

- Formally: instead of two labels, we have an output space $\mathcal{Y}$ containing a number of possible classes
  - Same machinery that we'll use later for exponentially large output spaces, including sequences and trees
- Decision rule: $\text{argmax}_{y \in \mathcal{Y}} w^\top f(x, y)$

features depend on choice of label now! note: this isn't the gold label

  - Multiple feature vectors, one weight vector
- Can also have one weight vector per class: $\text{argmax}_{y \in \mathcal{Y}} w_y^\top f(x)$

Different Weights vs. Different Features

- Different features: $\text{argmax}_{y \in \mathcal{Y}} w^\top f(x, y)$
  - Suppose $\mathcal{Y}$ is a structured label space (part-of-speech tags for each word in a sentence). *f*(*x*,*y*) extracts features over shared parts of these
- Different weights: $\text{argmax}_{y \in \mathcal{Y}} w_y^\top f(x)$
  - Generalizes to neural networks: *f*(*x*) is the first *n*-1 layers of the network, then you multiply by a final linear layer at the end
- For linear multiclass classification with discrete classes, these are identical

Feature Extraction

Block Feature Vectors

- Decision rule: $\mathrm{argmax}_{y \in \mathcal{Y}} w^\top f(x, y)$

too many drug trials, too few patients

Health

Sports

Science

- Base feature function:

$f(x)$ = I[contains *drug*], I[contains *patients*], I[contains *baseball*] = [1, 1, 0]

feature vector blocks for each label

$f(x, y =$ Health $) =$ [1, 1, 0, 0, 0, 0, 0, 0, 0] I[contains *drug* & label = Health]

$f(x, y =$ Sports $) =$ [0, 0, 0, 1, 1, 0, 0, 0, 0]

- Equivalent to having three weight vectors in this case
- We are NOT looking at the gold label! Instead looking at the candidate label

Making Decisions

too many drug trials, too few patients → Health, Sports, Science

$f(x)$ = I[contains *drug*], I[contains *patients*], I[contains *baseball*]

$f(x, y =$ Health $) =$ [1, 1, 0, 0, 0, 0, 0, 0, 0]

$f(x, y =$ Sports $) =$ [0, 0, 0, 1, 1, 0, 0, 0, 0]

"word drug in Science article" = +1.1

$w =$ [+2.1, +2.3, -5, -2.1, -3.8, +5.2, +1.1, -1.7, -1.3]

$w^\top f(x, y)$ = Health: +4.4 Sports: -5.9 Science: -0.6

argmax

Part-of-speech tagging as multiclass

- Classify *blocks* as one of 36 POS tags
- Example is a (sentence, index) pair $(x, i=2)$: the word *blocks* in this sentence
- Extract features with respect to this word:

the router *blocks* *the packets*

NNS
VBZ
NN
DT
…

$f(x, y{=}\mathrm{VBZ})$ = I[curr_word=*blocks* & tag = VBZ],
I[prev_word=*router* & tag = VBZ]
I[next_word=*the* & tag = VBZ]
I[curr_suffix=*s* & tag = VBZ]

not saying that *the* is tagged as VBZ! saying that *the* follows the VBZ word

- Next two lectures: sequence labeling

Multiclass Logistic Regression

Multiclass Logistic Regression

$$P_w(y|x) = \frac{\exp\left(w^\top f(x,y)\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(w^\top f(x,y')\right)}$$

sum over output space to normalize

- Compare to binary:

$$P(y=1|x) = \frac{\exp(w^\top f(x))}{1+\exp(w^\top f(x))}$$

negative class implicitly had *f*(*x*, *y*=0) = the zero vector

- exp/sum(exp): also called *softmax*

- Training: maximize $\mathcal{L}(x,y) = \sum_{j=1}^{n} \log P(y_j^*|x_j)$

$$= \sum_{j=1}^{n} \left(w^\top f(x_j, y_j^*) - \log \sum_{y} \exp(w^\top f(x_j, y)) \right)$$

Training

- Multiclass logistic regression $P_w(y|x) = \frac{\exp\left(w^\top f(x,y)\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(w^\top f(x,y')\right)}$

- Likelihood $\mathcal{L}(x_j, y_j^*) = w^\top f(x_j, y_j^*) - \log \sum_y \exp(w^\top f(x_j, y))$

$$\frac{\partial}{\partial w_i}\mathcal{L}(x_j, y_j^*) = f_i(x_j, y_j^*) - \frac{\sum_y f_i(x_j, y) \exp(w^\top f(x_j, y))}{\sum_y \exp(w^\top f(x_j, y))}$$

$$\frac{\partial}{\partial w_i}\mathcal{L}(x_j, y_j^*) = f_i(x_j, y_j^*) - \sum_y f_i(x_j, y) P_w(y|x_j)$$

$$\frac{\partial}{\partial w_i}\mathcal{L}(x_j, y_j^*) = f_i(x_j, y_j^*) - \mathbb{E}_y[f_i(x_j, y)]$$

model's expectation of feature value

gold feature value

Training

$$\frac{\partial}{\partial w_i}\mathcal{L}(x_j, y_j^*) = f_i(x_j, y_j^*) - \sum_y f_i(x_j, y) P_w(y|x_j)$$

too many drug trials, too few patients

y^* = Health

$f(x, y = \text{ Health }) = [1, 1, 0, 0, 0, 0, 0, 0, 0]$

$f(x, y = \text{ Sports }) = [0, 0, 0, 1, 1, 0, 0, 0, 0]$

$P_w(y|x)$ = [0.2, 0.5, 0.3]
(made up values)

gradient:

[1, 1, 0, 0, 0, 0, 0, 0, 0] — 0.2 [1, 1, 0, 0, 0, 0, 0, 0, 0] — 0.5 [0, 0, 0, 1, 1, 0, 0, 0, 0]
— 0.3 [0, 0, 0, 0, 0, 0, 1, 1, 0]

= [0.8, 0.8, 0, -0.5, -0.5, 0, -0.3, -0.3, 0]

"towards gold feature value, away from what the model thinks"

Multiclass Logistic Regression: Summary

- Model: $P_w(y|x) = \dfrac{\exp\left(w^\top f(x,y)\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(w^\top f(x,y')\right)}$

- Inference: $\text{argmax}_y P_w(y|x)$

- Learning: gradient ascent on the discriminative log-likelihood

$$f(x,y^*) - \mathbb{E}_y[f(x,y)] = f(x,y^*) - \sum_y [P_w(y|x) f(x,y)]$$

"towards gold feature value, away from expectation of feature value"

Generative vs. Discriminative Models

Learning in Probabilistic Models

- So far we have talked about discriminative classifiers (e.g., logistic regression which models $P(y|x)$)

- Cannot analytically compute optimal weights for such models, need to use gradient descent

- What about generative models?

Naive Bayes

- Data point $x = (x_1, ..., x_n)$, label $y \in \{0, 1\}$
- Formulate a probabilistic model that places a distribution $P(x, y)$
- Compute $P(y|x)$, predict $\text{argmax}_y P(y|x)$ to classify

$$P(y|x) = \frac{P(y)P(x|y)}{P(x)}$$

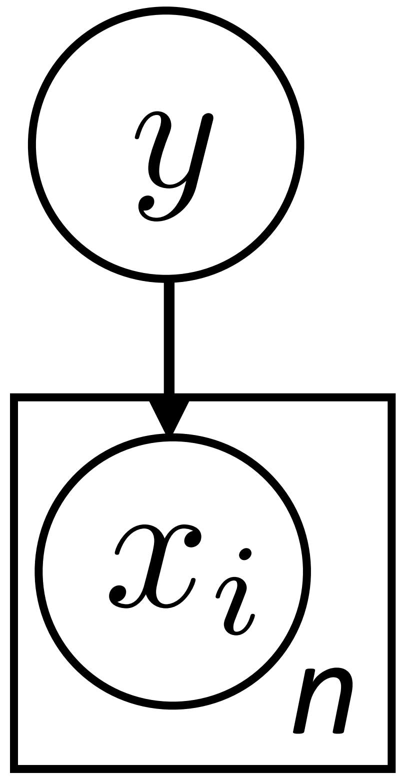
Bayes' Rule

constant: irrelevant for finding the max

$$\propto P(y)P(x|y)$$

"Naive" assumption:

$$= P(y) \prod_{i=1}^{n} P(x_i|y)$$



Maximum Likelihood Estimation

- Data points (x_j, y_j) provided (*j* indexes over examples)
- Find values of $P(y), \; P(x_i|y)$ that maximize data likelihood (generative):

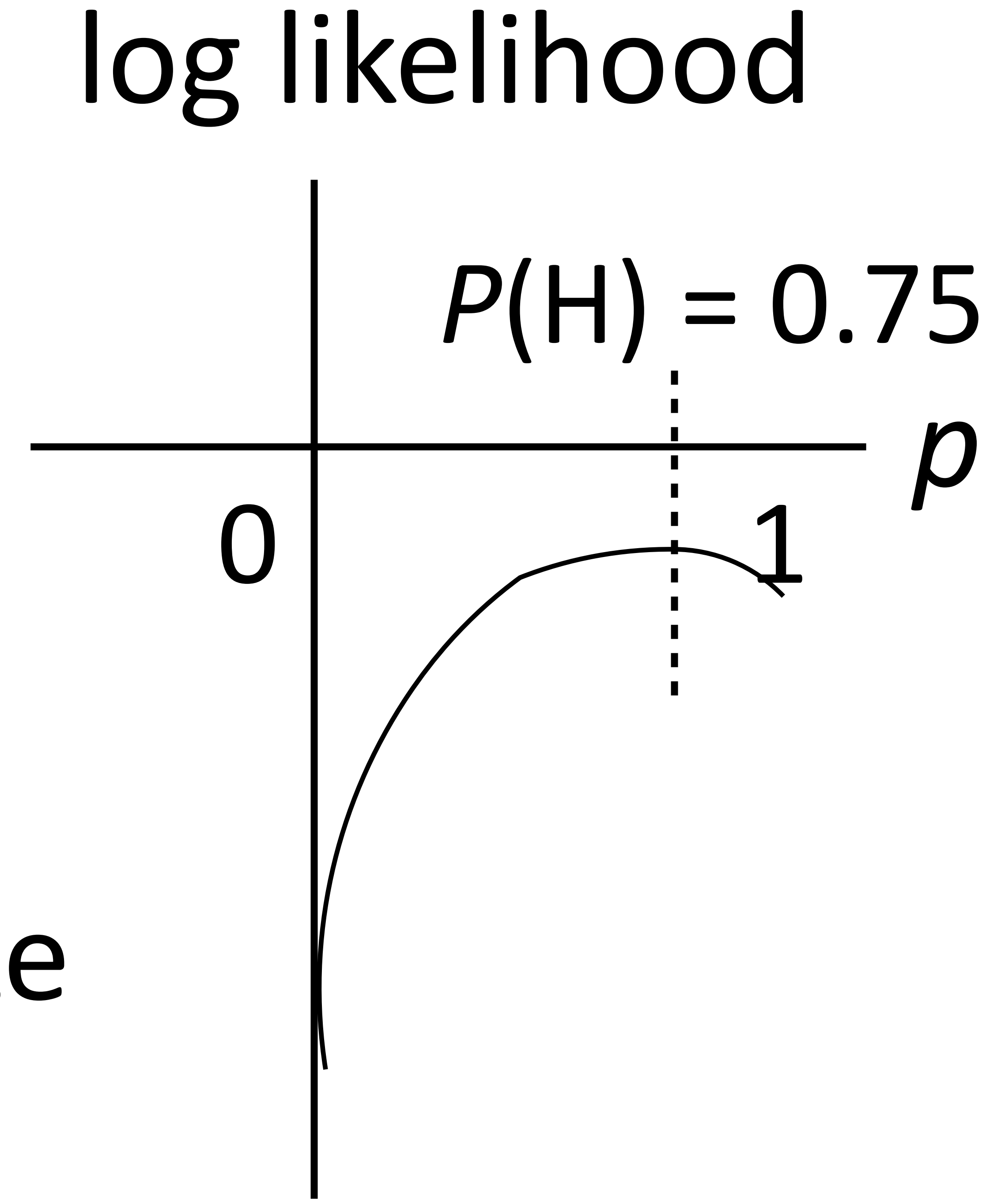
$$\prod_{j=1}^{m} P(y_j, x_j) = \prod_{j=1}^{m} P(y_j) \left[\prod_{i=1}^{n} P(x_{ji}|y_j) \right]$$

data points (*j*) features (*i*) *i*th feature of *j*th example

Maximum Likelihood Estimation

- Imagine a coin flip which is heads with probability *p*
- Observe (H, H, H, T) and maximize likelihood: $\prod_{j=1}^{m} P(y_j) = p^3(1-p)$
- Easier: maximize *log* likelihood

$$\sum_{j=1}^{m} \log P(y_j) = 3\log p + \log(1-p)$$



- Maximum likelihood parameters for binomial/ multinomial = read counts off of the data + normalize

Maximum Likelihood Estimation

- Data points (x_j, y_j) provided (*j* indexes over examples)
- Find values of $P(y), \;\; P(x_i|y)$ that maximize data likelihood (generative):

$$\prod_{j=1}^{m} P(y_j, x_j) = \prod_{j=1}^{m} P(y_j) \left[\prod_{i=1}^{n} P(x_{ji}|y_j) \right]$$

data points (*j*) features (*i*) *i*th feature of *j*th example

- Equivalent to maximizing log of data likelihood:

$$\sum_{j=1}^{m} \log P(y_j, x_j) = \sum_{j=1}^{m} \left[\log P(y_j) + \sum_{i=1}^{n} \log P(x_{ji}|y_j) \right]$$

- Can do this by counting and normalizing distributions!

Summary

- Next time: HMMs / POS tagging
  - Locally-normalized generative models, so easy to estimate from data
  - First thing we have that we could plausibly sample real sentences from
- In 2 lectures: CRFs (NER)
- You've now seen everything you need to implement multi-class classification models