Machine Learning Recap (multiclass classification)

Wei Xu

(many slides from Greg Durrett, Vivek Srikumar, Stanford CS231n)

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.

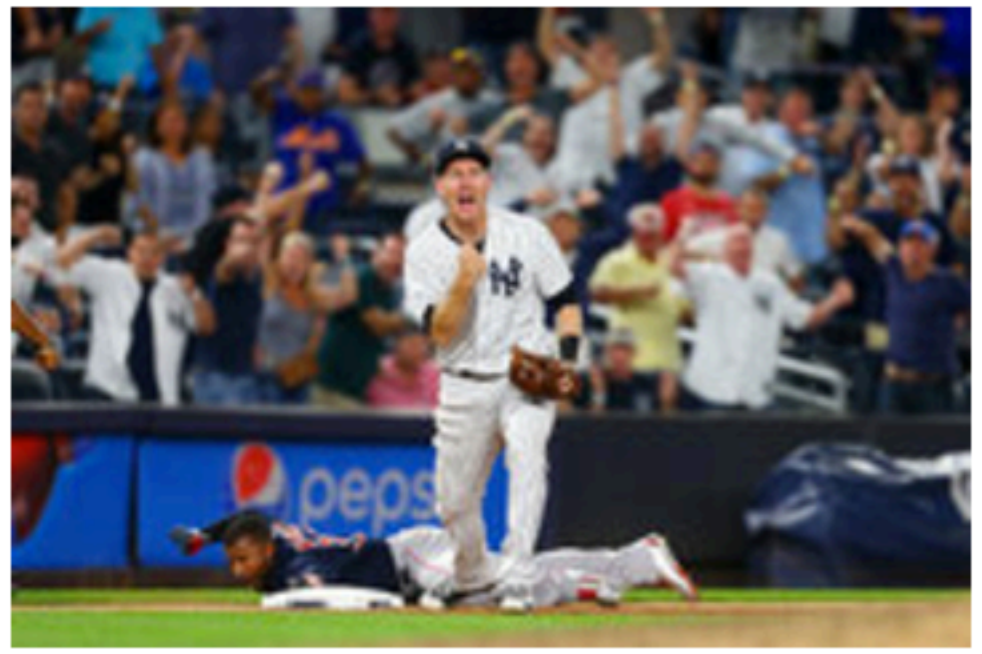
By GINA KOLATA

→ 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

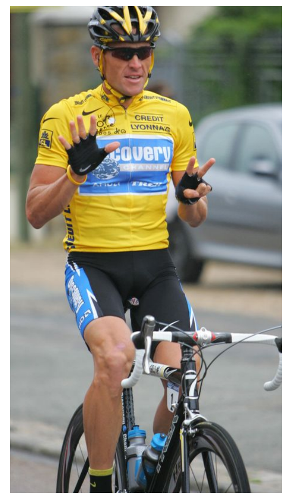
Dog

Car

- Thousands of classes (ImageNet)

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…

?

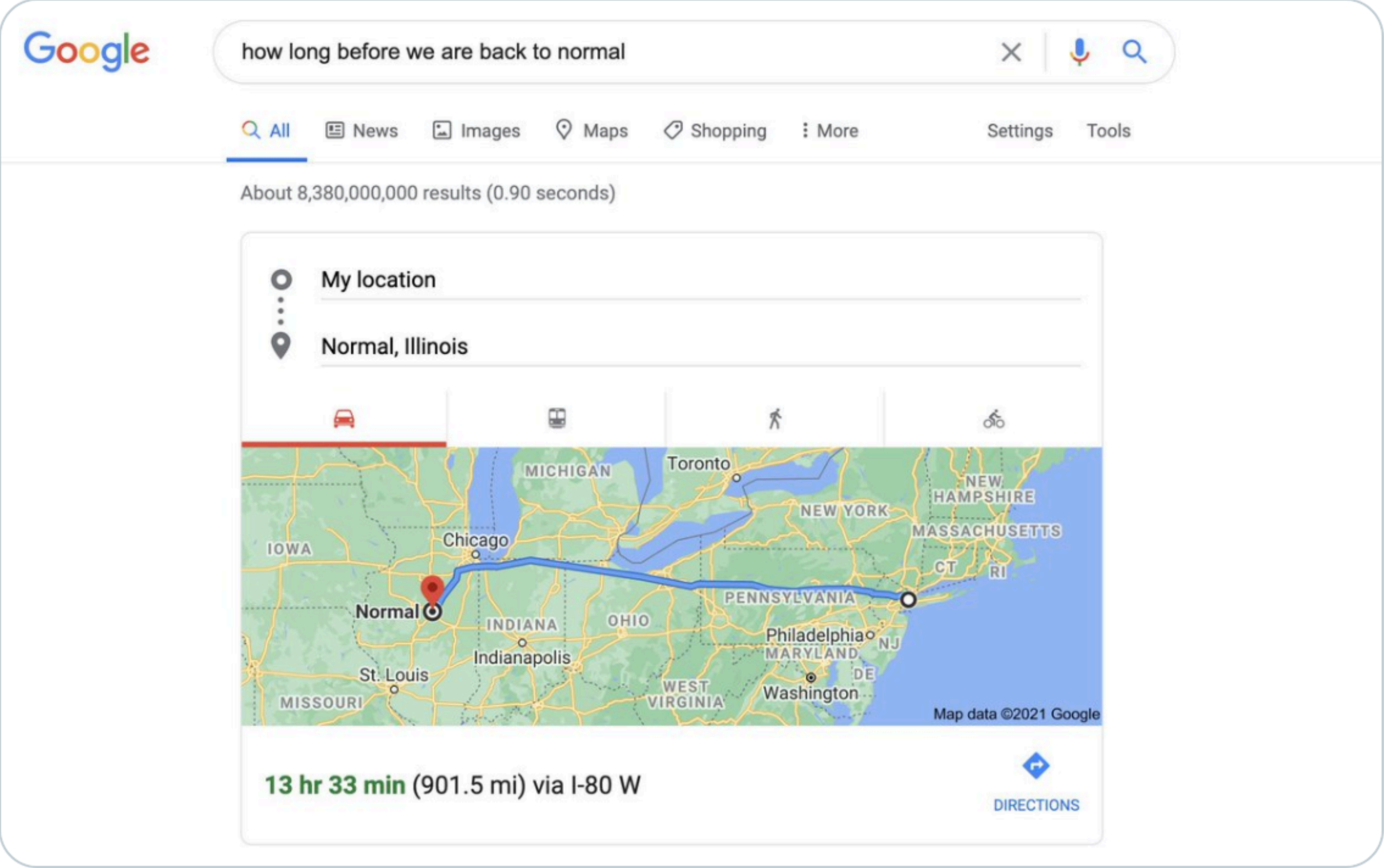
?

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

Entity Linking

Kara Schechtman
@karaschechtman

this is just decidedly not what I meant

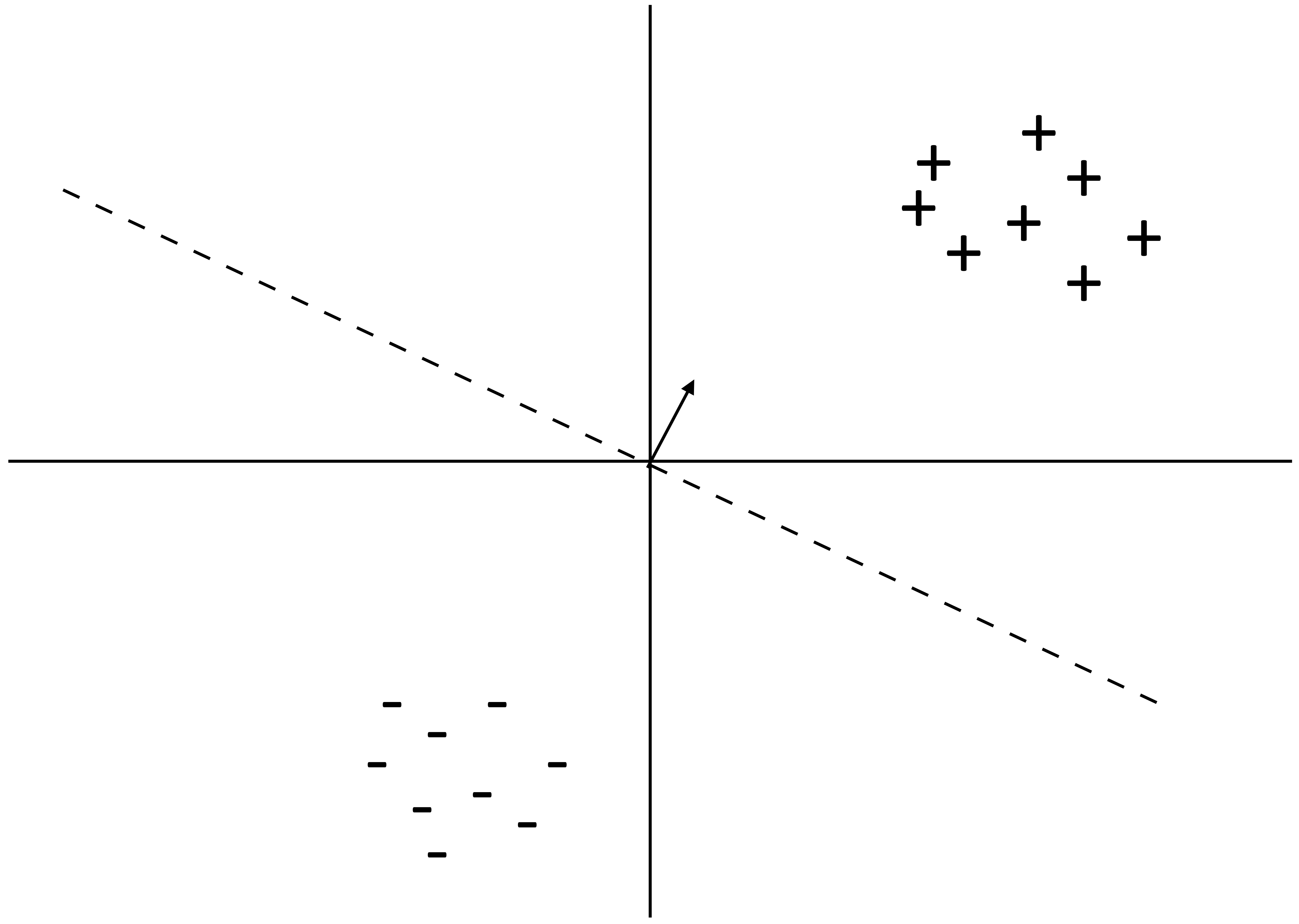


8:58 PM · Jan 30, 2021 · Twitter Web App

80 Retweets **16** Quote Tweets **700** Likes

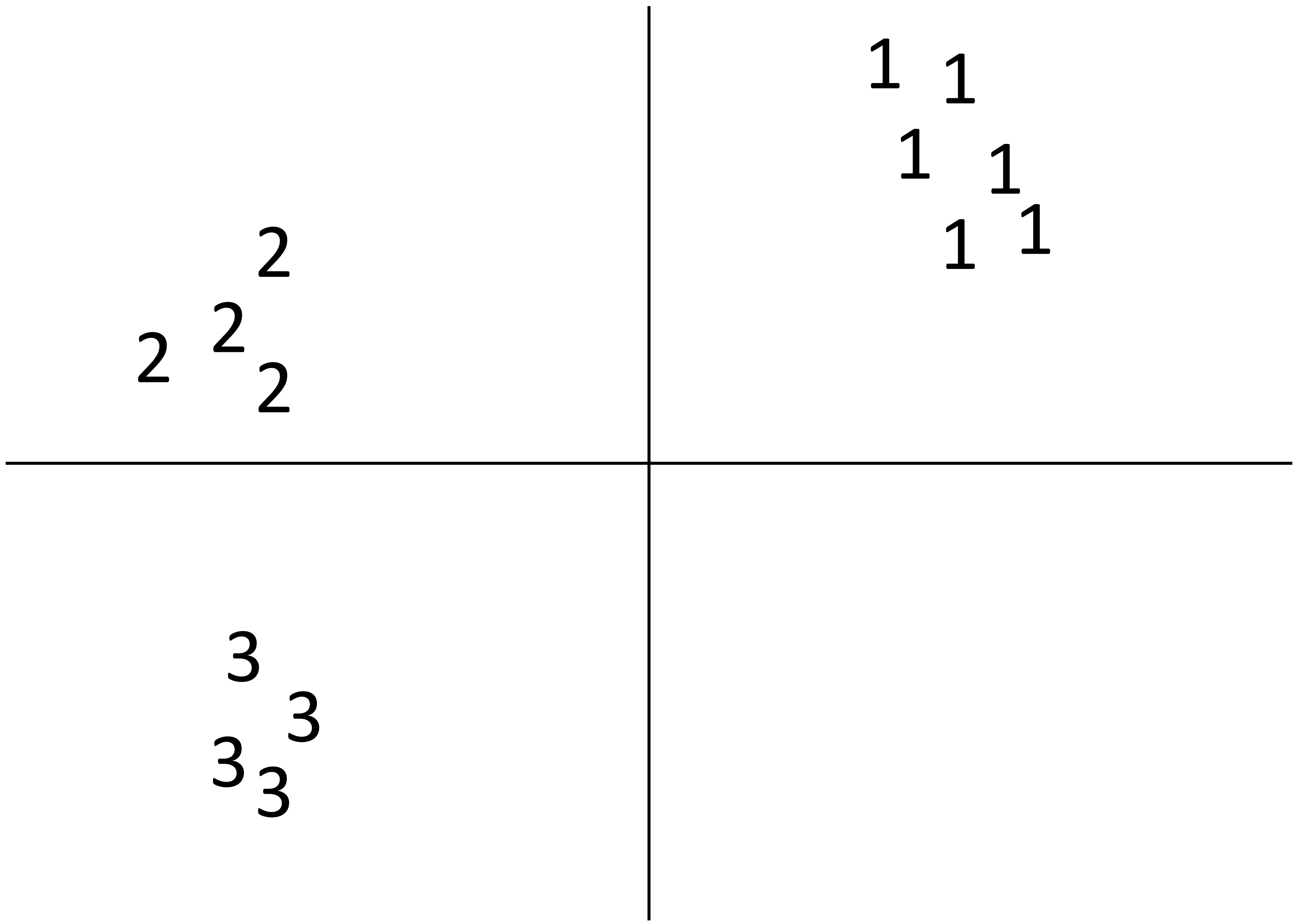
Binary Classification

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



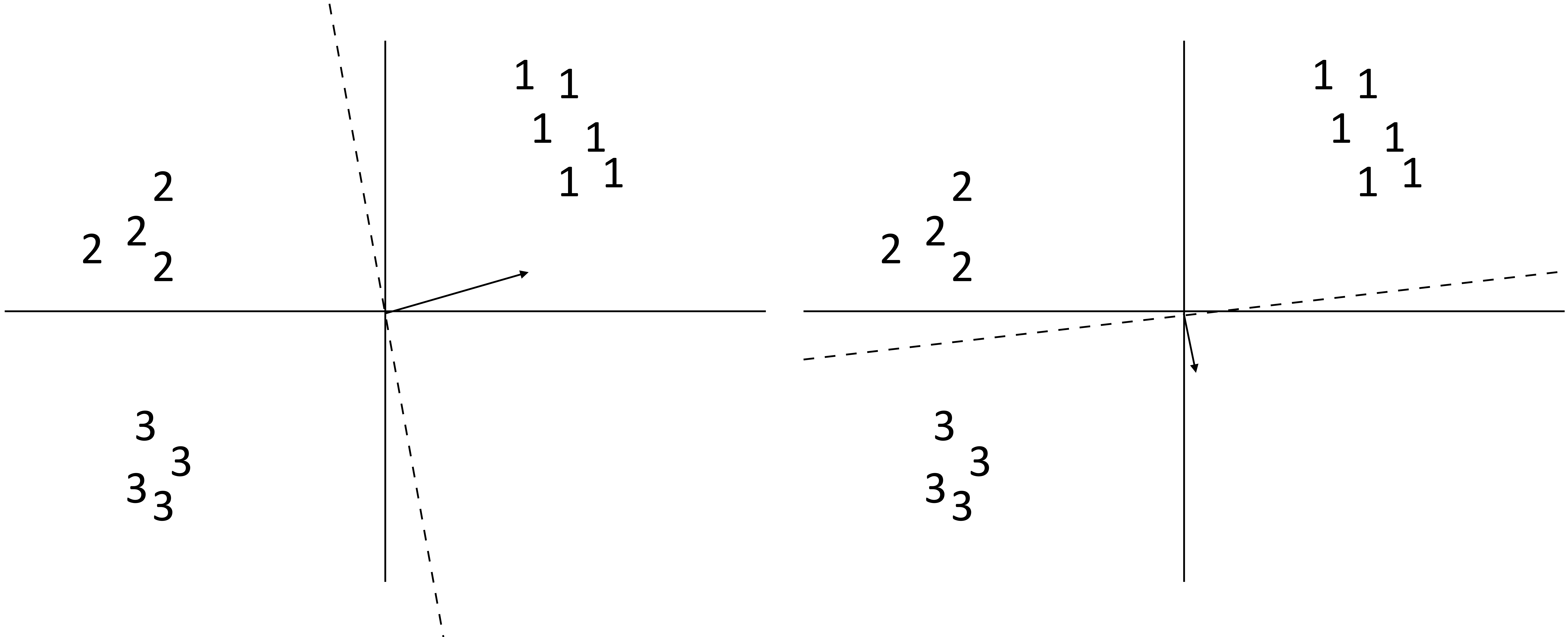
Multiclass Classification

- Can we just use binary classifiers here?



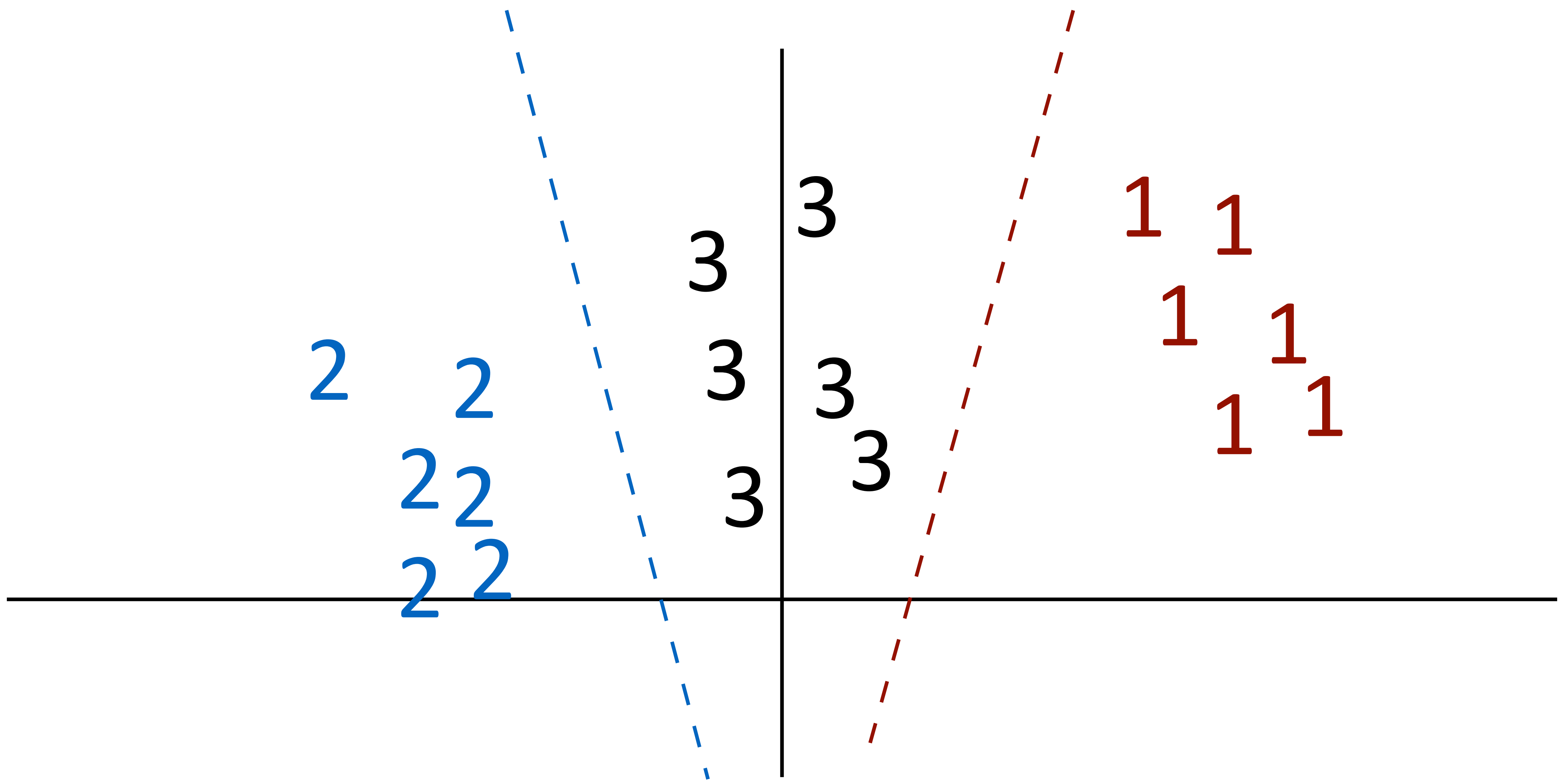
Multiclass Classification

- 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

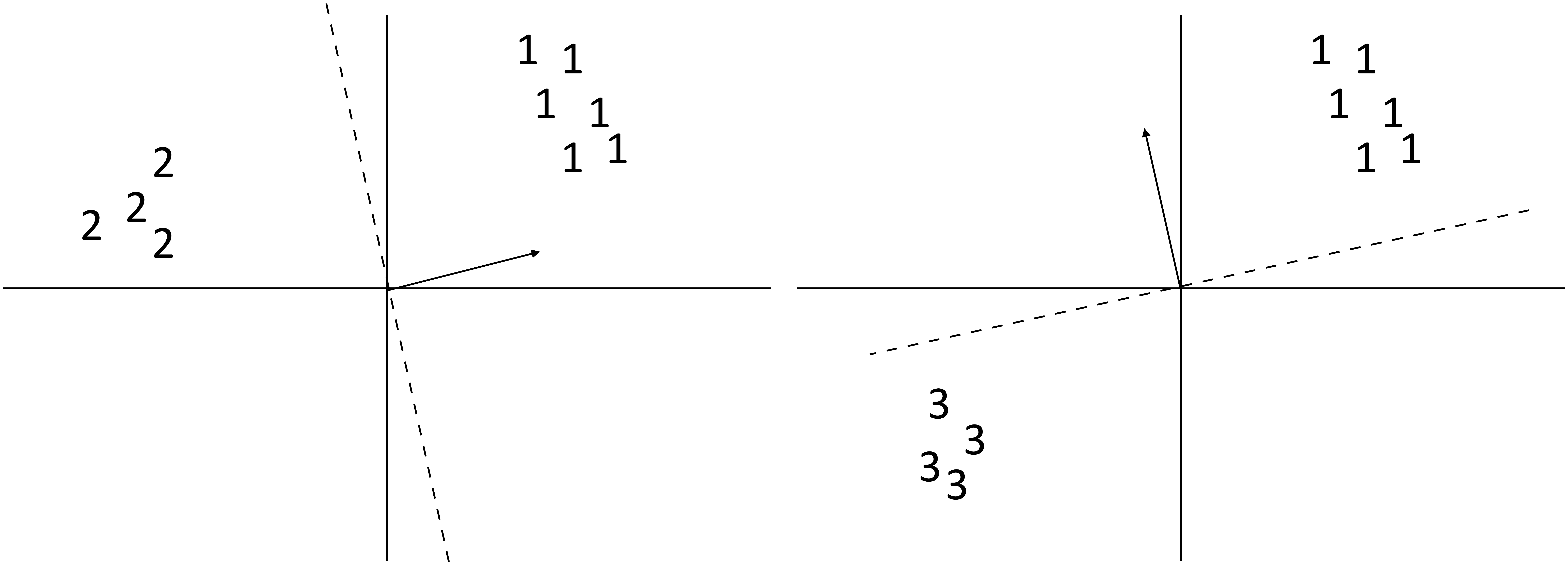
- Not all classes may even be separable using this approach



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

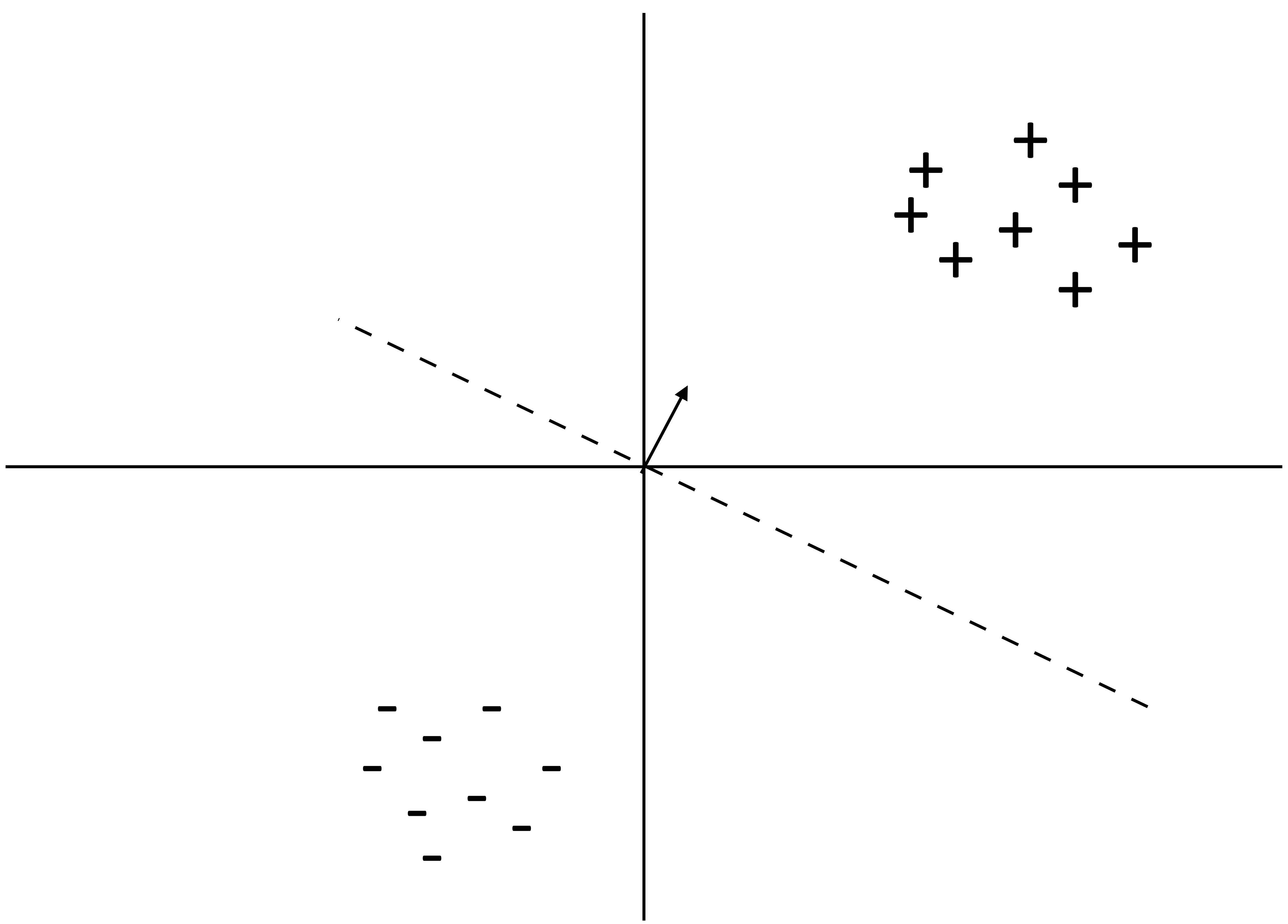
Multiclass Classification

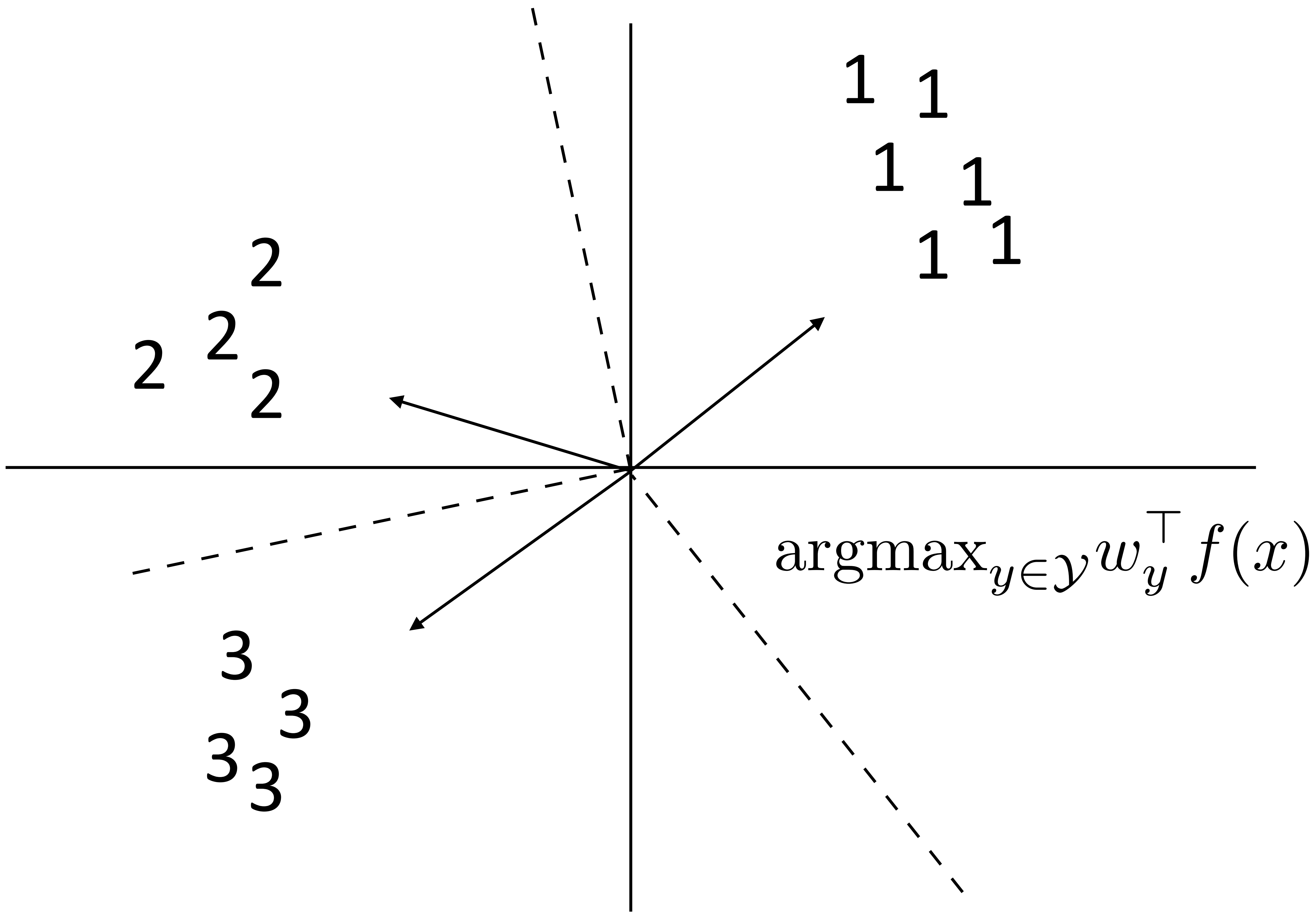
- All-vs-all: train n(n-1)/2 classifiers to differentiate each pair of classes
- Again, how to reconcile?



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: $\text{argmax}_{y \in \mathcal{Y}} w_y^\top f(x)$
  - The single weight vector approach will generalize to structured output spaces, whereas per-class weight vectors won't

Feature Extraction

Block Feature Vectors

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

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]

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

- Equivalent to having three weight vectors in this case $\text{argmax}_{y \in \mathcal{Y}} w_y^\top f(x)$

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]

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

"word drug in Science article" = +1.1

$w =$ [+2.1, +2.3, -5, -2.1, -3.8, 0, +1.1, -1.7, -1.3]

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

↖ argmax

Another example: POS tagging

- Classify *blocks* as one of 33 POS tags

- Example *x*: sentence with a word (in this case, *blocks*) highlighted

- Extract features with respect to this word:

the router *blocks* *the packets*

NNS
VBZ
NN
DT
…

$f(x, y=\text{VBZ})$ = 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

- Later lectures: sequence labeling!

Multiclass Logistic Regression

Multiclass Logistic Regression

Softmax function

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

sum over output space to normalize

- Compare to binary:

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

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

Why? Interpret raw classifier scores as **probabilities**

too many drug trials, too few patients

$w^\top f(x,y)$		probabilities must be >= 0 (unnormalized probabilities)		probabilities must sum to 1 ($P_w(y\|x)$)
Health: +2.2	$\xrightarrow{\text{exp}}$	6.05	$\xrightarrow{\text{normalize}}$	0.21
Sports: +3.1		22.2		0.77
Science: -0.6		0.55		0.02

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

i.e. minimize negative log likelihood or cross-entropy loss

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

index of data points (*j*)

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

too many drug trials, too few patients

Health: +2.2
Sports: +3.1
Science: -0.6

$w^\top f(x,y)$

$\xrightarrow{\text{exp}}$

probabilities must be >= 0

6.05
22.2
0.55

unnormalized probabilities

$\xrightarrow{\text{normalize}}$

probabilities must sum to 1

0.21
0.77
0.02

$P_w(y|x)$

log(0.21) = - 1.56

$\xleftrightarrow{\text{compare}}$

$\mathcal{L}(x_j, y_j^*) = \log P(y_j^*|x_j)$

1.00
0.00
0.00

correct (gold) probabilities

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

i.e. minimize negative log likelihood or cross-entropy loss

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

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

index of data points (j)

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

gold feature value

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

model's expectation of feature value

too many drug trials, too few patients

y^* = Health

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

$P_w(y|x)$ = [0.21, 0.77, 0.02]

gradient: [1, 1, 0, 0, 0, 0, 0, 0, 0] — 0.21 [1, 1, 0, 0, 0, 0, 0, 0, 0]
— 0.77 [0, 0, 0, 1, 1, 0, 0, 0, 0] — 0.02 [0, 0, 0, 0, 0, 0, 1, 1, 0]

= [0.79, 0.79, 0, -0.77, -0.77, 0, -0.02, -0.02, 0]

update $w^\top$:
[1.3, 0.9, -5, 3.2, -0.1, 0, 1.1, -1.7, -1.3] + [0.79, 0.79, 0, -0.77, -0.77, 0, -0.02, -0.02, 0]
= [2.09, 1.69, 0, 2.43, -0.87, 0, 1.08, -1.72, 0]

new $P_w(y|x)$ = [0.89, 0.10, 0.01]

Multiclass Logistic Regression: Summary

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

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

- 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"

Optimization

- Gradient descent (or ascent)
  - **Batch update** for logistic regression
  - Each update is based on a computation over the entire dataset

Multiclass Logistic Regression

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

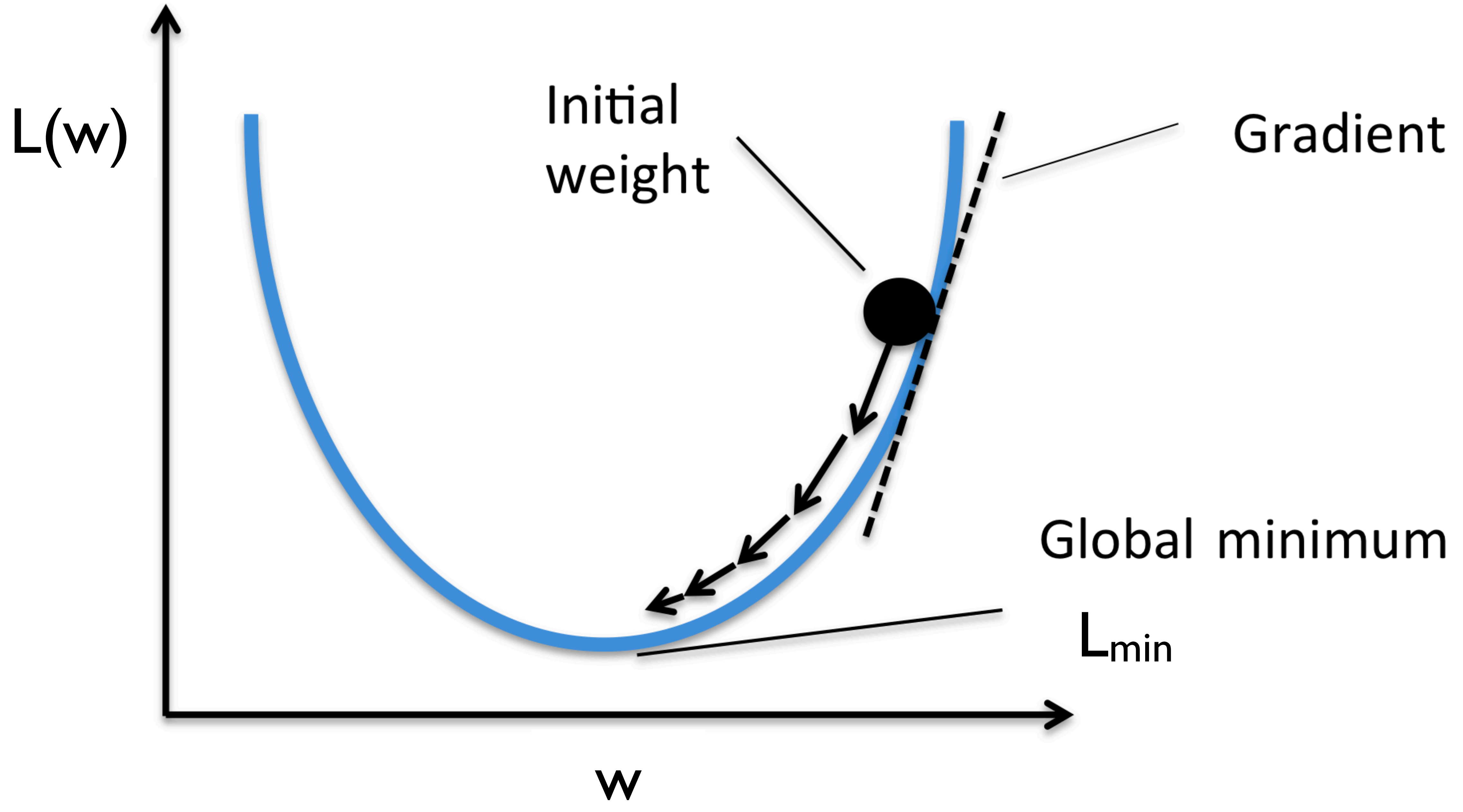
sum over output space to normalize

i.e. minimize negative log likelihood or cross-entropy loss

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

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

index of data points (*j*)



Optimization

- Gradient descent
  - **Batch update** for logistic regression
  - Each update is based on a computation over the entire dataset

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

i.e. minimize negative log likelihood or cross-entropy loss

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

index of data points (j)

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

- Very simple to code up

```
# Vanilla Gradient Descent

while True:
  weights_grad = evaluate_gradient(loss_fun, data, weights)
  weights += - step_size * weights_grad # perform parameter update
```

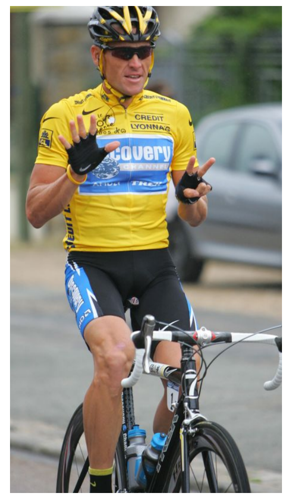
Another Example: Entity Linking

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.

?

?



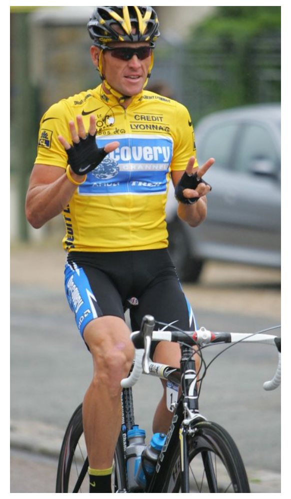
Lance Edward Armstrong is an American former professional road cyclist

Armstrong County is a county in Pennsylvania…

- 4.5M classes, not enough data to learn features like “Tour de France <-> en/wiki/Lance_Armstrong”
- Instead, features *f*(*x*, *y*) look at the actual article associated with *y*

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

Armstrong County

- tf-idf(doc, *w*) = freq of *w* in doc * log(4.5M/# Wiki articles *w* occurs in)
  - *the*: occurs in every article, tf-idf = 0
  - *cyclist*: occurs in 1% of articles, tf-idf = # occurrences * $\log_{10}(100)$
- tf-idf(doc) = vector of tf-idf(doc, *w*) for all words in vocabulary (50,000)
- *f*(*x*,*y*) = [cos(tf-idf(*x*), tf-idf(*y*)), … other features]

Entity Linking

Baseline Feature: $P(t\|m), P(t)$
Local Features: $\phi_i(t,m)$
cosine-sim(Text(t),Text(m)) : Naive/Reweighted
cosine-sim(Text(t),Context(m)): Naive/Reweighted
cosine-sim(Context(t),Text(m)): Naive/Reweighted
cosine-sim(Context(t),Context(m)): Naive/Reweighted
Global Features: $\psi_i(t_i,t_j)$
$I_{[t_i-t_j]}$*PMI(InLinks(t_i),InLinks(t_j)) : avg/max*
$I_{[t_i-t_j]}$*NGD(InLinks(t_i),InLinks(t_j)) : avg/max*
$I_{[t_i-t_j]}$*PMI(OutLinks(t_i),OutLinks(t_j)) : avg/max*
$I_{[t_i-t_j]}$*NGD(OutLinks(t_i),OutLinks(t_j)) : avg/max*
$I_{[t_i\leftrightarrow t_j]}$ *: avg/max*
$I_{[t_i\leftrightarrow t_j]}$*PMI(InLinks($t_i$),InLinks($t_j$)) : avg/max*
$I_{[t_i\leftrightarrow t_j]}$*NGD(InLinks($t_i$),InLinks($t_j$)) : avg/max*
$I_{[t_i\leftrightarrow t_j]}$*PMI(OutLinks($t_i$),OutLinks($t_j$)) : avg/max*
$I_{[t_i\leftrightarrow t_j]}$*NGD(OutLinks($t_i$),OutLinks($t_j$)) : avg/max*

Table 1: Ranker features. $I_{[t_i-t_j]}$ is an indicator variable which is 1 iff t_i links to t_j or vise-versa. $I_{[t_i\leftrightarrow t_j]}$ is 1 iff the titles point to each other.

Wikipedia titles. We are aware of two previous methods for estimating the relatedness between two Wikipedia concepts: (Strube and Ponzetto, 2006), which uses category overlap, and (Milne and Witten, 2008a), which uses the incoming link structure. Previous work experimented with two relatedness measures: NGD, and Specificity-weighted Cosine Similarity. Consistent with previous work, we found NGD to be the better-performing of the two. Thus we use only NGD along with a well-known Pontwise Mutual Information (PMI) relatedness measure. Given a Wikipedia title collection W, titles t_1 and t_2 with a set of incoming links L_1, and L_2 respectively, PMI and NGD are defined as follows:

$$NGD(L_1,L_2) = \frac{Log(Max(|L_1|,|L_2|)) - Log(|L_1 \cap L_2|)}{Log(|W|) - Log(Min(|L_1|,|L_2|))}$$

$$PMI(L_1,L_2) = \frac{|L_1 \cap L_2|/|W|}{|L_1|/|W||L_2|/|W|}$$

The NGD and the PMI measures can also be computed over the set of *outgoing* links, and we include these as features as well. We also included a fea-

- $f(x,y)$ = [cos(tf-idf(x), tf-idf(y)), ... other features]

Ratinov et al. (2011)

Next Up

- You’ve now seen everything you need to implement multi-class classification models
- Next up: Neural Network Basics!
- In 2 weeks: Sequential Models (HMM, CRF, …) for POS tagging, NER

QA Time