CS395T: Structured Models for NLP
Lecture 3: Multiclass Classification

Greg Durrett

Some slides adapted from Vivek Srikumar, University of Utah

Administrivia

- Course enrollment
- Project 1 out next Tuesday

Recall: Binary Classification

- Logistic regression: $P(y=1|x) = \dfrac{\exp\left(\sum_{i=1}^{n} w_i x_i\right)}{\left(1+\exp\left(\sum_{i=1}^{n} w_i x_i\right)\right)}$

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

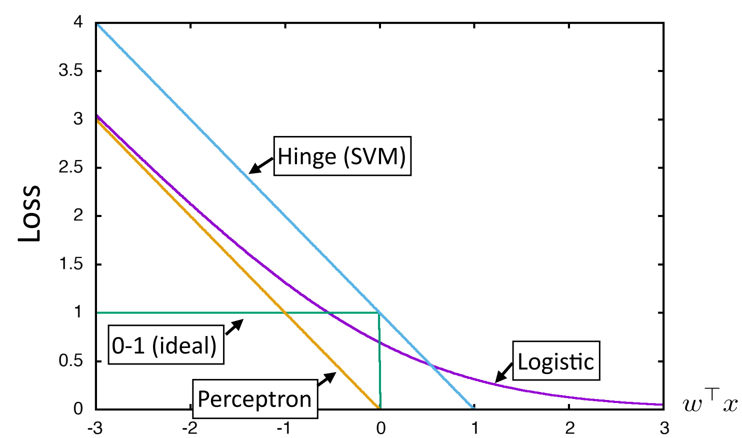
  Gradient (unregularized): $x(y - P(y=1|x))$

- SVM: quadratic program to minimize weight vector norm w/slack

  Decision rule: $w^\top x \geq 0$

  (Sub)gradient (unregularized): 0 if correct with margin of 1, else $x(2y-1)$

Loss Functions



Sentiment Analysis

- Classify sentence as positive or negative sentiment

Positive

Negative

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

Wang and Manning (2012)

Method	RT-s	MPQA	CR	Subj.
MNB-uni	77.9	85.3	79.8	**92.6**
MNB-bi	**79.0**	**86.3**	80.0	**93.6**
SVM-uni	76.2	86.1	79.0	90.8
SVM-bi	77.7	**86.7**	80.8	91.7
NBSVM-uni	**78.1**	85.3	80.5	92.4
NBSVM-bi	**79.4**	**86.3**	**81.8**	**93.2**
RAE	76.8	85.7	–	–
RAE-pretrain	77.7	**86.4**	–	–
Voting-w/Rev.	63.1	81.7	74.2	–
Rule	62.9	81.8	74.3	–
BoF-noDic.	75.7	81.8	79.3	–
BoF-w/Rev.	76.4	84.1	**81.4**	–
Tree-CRF	77.3	86.1	**81.4**	–
BoWSVM	–	–	–	90.0
	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

Two years later Kim (2014) with neural networks

This Lecture

- Multiclass fundamentals
- Feature extraction
- Multiclass logistic regression
- Multiclass SVM
- Optimization

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.

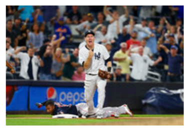
By GINA KOLATA

→ 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)

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

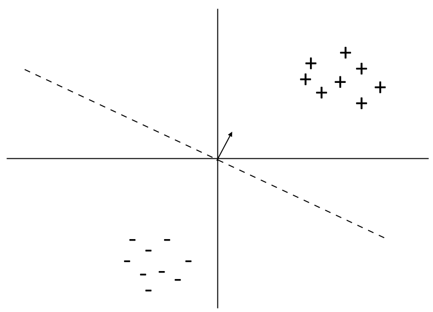
After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

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

Richardson (2013)

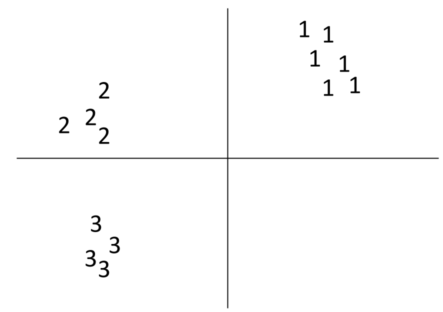
Binary Classification

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



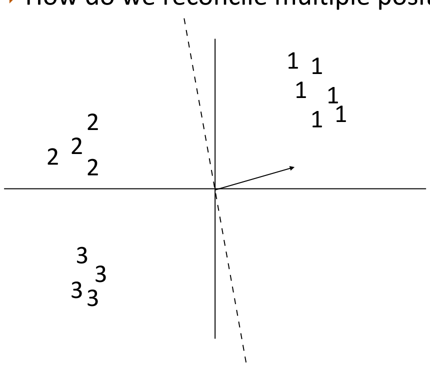
Multiclass Classification

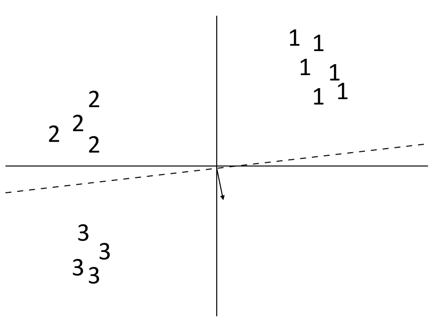
- Can we just use binary classifiers here?



Multiclass Classification

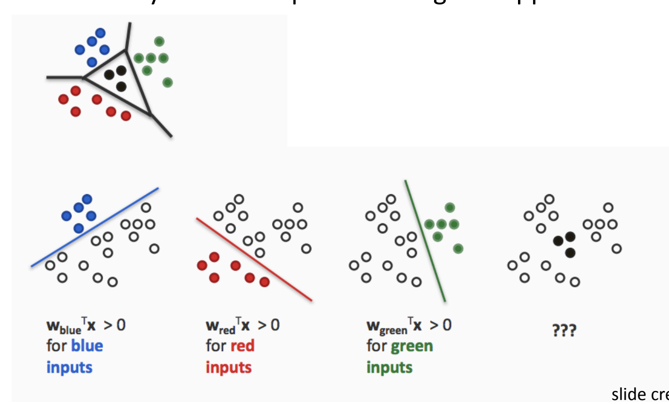
- One-vs-all: train *k* classifiers, one to distinguish each class from all the rest
- How do we reconcile multiple positive predictions? Highest score?





Multiclass Classification

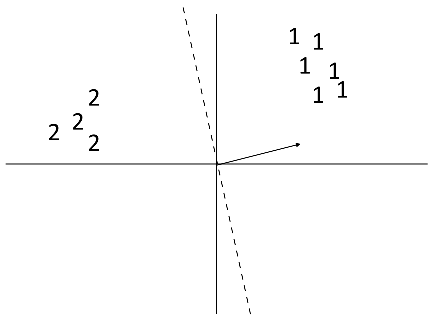
- Not all classes may even be separable using this approach

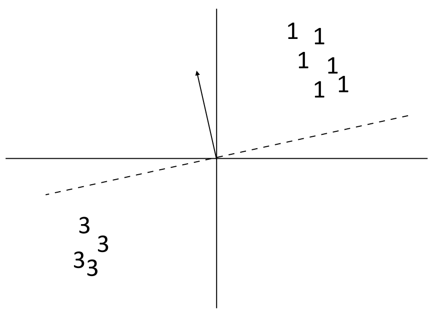


slide credit: Vivek Srikumar

Multiclass Classification

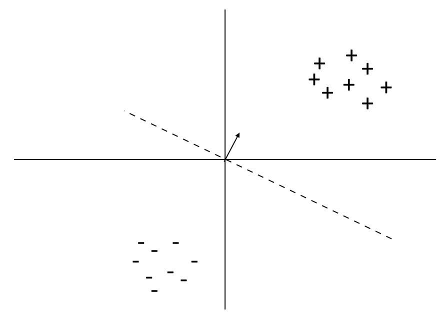
- All-vs-all: train n(n-1)/2 classifiers to differentiate each pair of classes
- Again, how to reconcile?

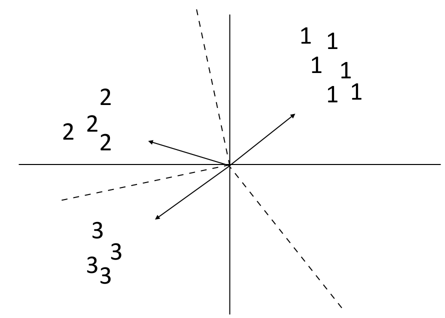




Multiclass Classification

- Binary classification: one weight vector defines both classes
- Multiclass classification: one weight vector per class, decision is argmax





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: $\text{argmax}_{y \in \mathcal{Y}} w^\top f(x, y)$
  - Multiple feature vectors, one weight vector
  - Can also have one weight vector per class: $\text{argmax}_{y \in \mathcal{Y}} w_y^\top f(x)$
  - Why do we do with separate feature vectors? Let's see!

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]

- Equivalent to having three weight vectors, but this formulation is more general if the features depend on *y*

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 ← argmax
Sports -5.9
Science -1.9

Multiclass Logistic Regression

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

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

- 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 $P(y|x) = \dfrac{\exp\left(w^\top f(x,y)\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(w^\top f(x,y')\right)}$
- Likelihood $\mathcal{L}(x_j, y_j^*) = w^\top f(x_j, y_j^*) - \log \sum_y \exp(w^\top f(x_j, y))$

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

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

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

model's expectation of feature value from

Logistic Regression: Summary

- Model: $P(y|x) = \dfrac{\exp\left(w^\top f(x,y)\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(w^\top f(x,y')\right)}$
- Inference: $\text{argmax}_y P(y|x)$
- Learning: gradient ascent on the discriminative log-likelihood

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

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

Training

- Are all decisions equally costly?

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

Predicted Sports: bad error

Predicted Science: not so bad

- We can define a loss function $\ell(y, y^*)$

$\ell($ Sports , Health $) = 3$

$\ell($Science, Health $) = 1$

Multiclass SVM

Minimize $\lambda\|w\|_2^2 + \sum_{j=1}^{m} \xi_j$ ← slack variables > 0 iff example is support vector

s.t. $\forall j \quad \xi_j \geq 0$

$\cancel{\forall j \quad (2y_j - 1)(w^\top x_j) \geq 1 - \xi_j}$

$\forall j \forall y \in \mathcal{Y} \quad w^\top f(x_j, y_j^*) \geq w^\top f(x_j, y) + \ell(y, y_j^*) - \xi_j$

Correct prediction now has to beat every other class

Score comparison is more explicit now

The 1 that was here is replaced by a loss function

Multiclass SVM

Minimize $\lambda\|w\|_2^2 + \sum_{j=1}^{m} \xi_j$

s.t. $\forall j \quad \xi_j \geq 0$

$\forall j \forall y \in \mathcal{Y} \quad w^\top f(x_j, y_j^*) \geq w^\top f(x_j, y) + \ell(y, y_j^*) - \xi_j$

- How does this quantification come into play?
- One slack variable per example, so it's set to be whatever the *most violated constraint* is for that example

$$\xi_j = \max_{y \in \mathcal{Y}} \boxed{w^\top f(x_j, y) + \ell(y, y_j^*)} - w^\top f(x_j, y_j^*)$$

- Plug in the gold *y* and you get 0, so slack is always nonnegative!

Loss-Augmented Decoding

$$\xi_j = \max_{y \in \mathcal{Y}} \boxed{w^\top f(x_j, y) + \ell(y, y_j^*)} - w^\top f(x_j, y_j^*)$$

too many drug trials, too few patients Health

	$w^\top f(x, y)$	Loss	Total
Health	+2.4	0	2.4
Sports	+1.3	3	4.3 ← argmax
Science	+1.8	1	2.8

- Sports is most violated constraint, slack = 4.3 — 2.4 = 1.9
- Perceptron would make no update, regular SVM would pick Science

Computing the Subgradient

$$\text{Minimize } \lambda\|w\|_2^2 + \sum_{j=1}^{m}\xi_j$$
$$\text{s.t. } \forall j \quad \xi_j \geq 0$$
$$\forall j \forall y \in \mathcal{Y} \quad w^\top f(x_j, y_j^*) \geq w^\top f(x_j, y) + \ell(y, y_j^*) - \xi_j$$

- If $\xi_j = 0$, the example is not a support vector, gradient is zero
- Otherwise, $\xi_j = \boxed{\max_{y\in\mathcal{Y}} w^\top f(x_j, y) + \ell(y, y_j^*)} - w^\top f(x_j, y_j^*)$

$$\frac{\partial}{\partial w_i}\xi_j = f_i(x_j, \boxed{y_{\max}}) - f_i(x_j, y_j^*) \leftarrow$$ (update looks backwards — we're minimizing here!)

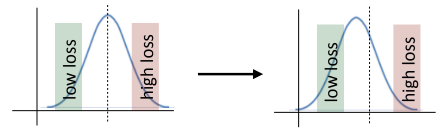
- Perceptron-like, but we update away from *loss-augmented* prediction

Softmax Margin

- Can we include a loss function in logistic regression?

$$P(y|x) = \frac{\exp\left(w^\top f(x,y) + \ell(y, y^*)\right)}{\sum_{y'} \exp\left(w^\top f(x,y') + \ell(y', y_j^*)\right)}$$

- Likelihood is artificially higher for things with high loss — training needs to work even harder to maximize the likelihood of the right thing!



- Biased estimator for original likelihood, but better loss

Gimpel and Smith (2010)

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.

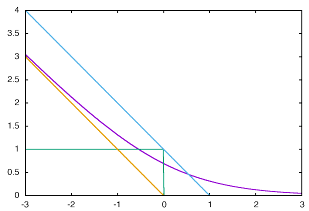
Lance Edward Armstrong

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 * log10(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]

Structured Prediction

- Four elements of a structured machine learning method:
  - Model: probabilistic, max-margin, deep neural network
  - Objective



  - Inference: just maxes so far, but will get harder
  - Training: gradient descent

Optimization

- Stochastic gradient *ascent*

$$w \leftarrow w + \alpha g, \quad g = \frac{\partial}{\partial w}\mathcal{L}$$

  - Very simple to code up
  - "First-order" technique: only relies on having gradient
  - Difficult to tune step size
- Newton's method

$$w \leftarrow w + \left(\frac{\partial^2}{\partial w^2}\mathcal{L}\right)^{-1} g$$

  - Second-order technique
  - Optimizes quadratic instantly

Inverse Hessian: *n* x *n* mat, expensive!

- Quasi-Newton methods: L-BFGS, etc.
  - Approximate inverse Hessian with gradients over time

AdaGrad

- Optimized for problems with sparse features
- Per-parameter learning rate: smaller updates are made to parameters that get updated frequently

$$w_i \leftarrow w_i + \alpha \frac{1}{\sqrt{\sum_{\tau=1}^{t} g_{\tau,i}^2}} g_{t_i}$$

accumulate sum of squared gradients from previous updates

- Generally much more robust, requires little tuning of learning rates
- Other techniques for optimizing deep models — more later!

Duchi et al. (2011)

Structured Prediction

- Design tradeoffs need to reflect interactions:
  - Model and objective are coupled: probabilistic model <-> maximize likelihood
  - ...but not always: a linear model or neural network can be trained to minimize any differentiable loss function
  - Inference governs what learning: need to be able to compute expectations to use logistic regression

Summary

- You've now seen everything you need to implement multi-class classification models

- Next time: HMMs (POS tagging)

- In 2 lectures: CRFs (NER)