CS395T: Structured Models for NLP
Lecture 2: Machine Learning Review

Greg Durrett

Some slides adapted from Vivek Srikumar, University of Utah

Administrivia

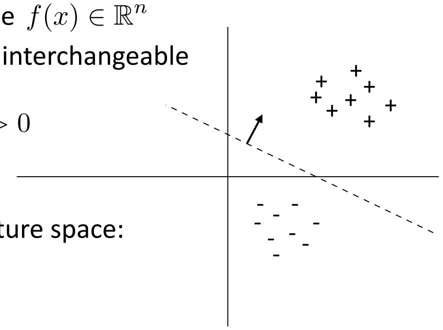
- Course enrollment
- Lecture slides posted on website

This Lecture

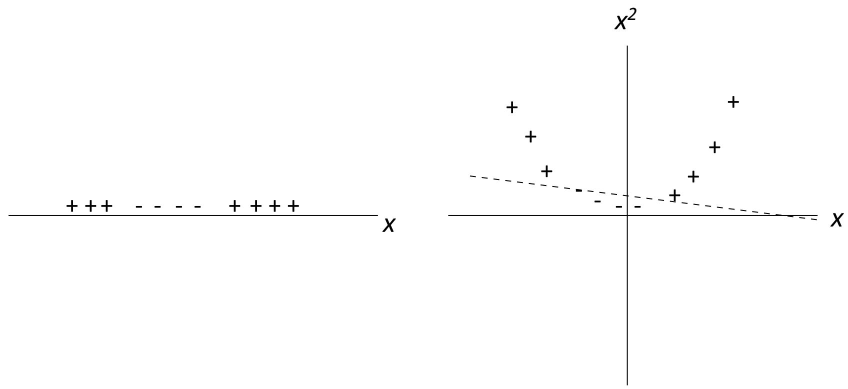
- Linear classification fundamentals
- Naive Bayes, maximum likelihood in generative models
- Three discriminative models: logistic regression, perceptron, SVM
  - Different motivations but very similar update rules / inference!

Classification

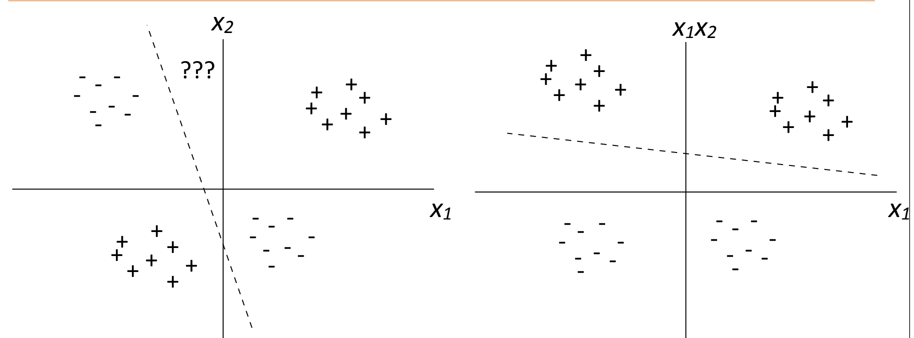
- Datapoint x with label $y \in \{0, 1\}$
- Embed datapoint in a feature space $f(x) \in \mathbb{R}^n$
  but in this lecture $f(x)$ and x are interchangeable
- Linear decision rule: $w^\top f(x) + b > 0$
  $w^\top f(x) > 0$
- Can delete bias if we augment feature space:
  $f(x)$ = [0.5, 1.6, 0.3]
  ↓
  [0.5, 1.6, 0.3, **1**]



Linear functions are powerful!



Linear functions are powerful!



$f(x) = [x_1, x_2]$ $f(x) = [x_1, x_2, x_1^2, x_2^2, x_1x_2]$

- "Kernel trick" does this for "free," but is too expensive to use in NLP applications, training is $O(n^2)$ instead of $O(n \cdot (\text{num feats}))$

Classification: Sentiment Analysis

this movie was great! would watch again Positive

this movie was not really very enjoyable Negative

- Doing well at this is going to require structure, but let's start with simple approaches

Text Classification: Ham or Spam

Hi, I just wanted to send over the latest results from training the LSTM model. In the attachment. What do you think of the performance?

Ham

hi i have very valuable business proposition for you. you make lots of $$$ I just need you to send a small amount of funds

Spam

- Surface cues can basically tell you what's going on here
- Machine learning is good at this! Lots of data, simple pattern recognition task, hard to write rules by hand

Text Classification: Ham or Spam

Hi, I just wanted to send over the latest results from training the LSTM model. In the attachment. What do you think of the performance?

hi i have very valuable business proposition for you. you make lots of $$$ I just need you to send a small amount of funds

Ham

Spam

???

- Why do we think this?
- Conditional probabilities (chance of spam given $$$ is high)

Text Classification: Ham or Spam

Hi, I just wanted to send over the latest results from training the LSTM model. In the attachment. What do you think of the performance?

hi i have very valuable business proposition for you. you make lots of $$$ I just need you to send a small amount of funds

Ham

Spam

???

- Feature representation: Indicator[doc contains *$$$*], Indicator[doc contains *training*], Indicator[doc contains *send*]…
- Convert a document to a vector: [1, 0, 1, …] (~50,000 long)
  - Requires *indexing* the features (mapping them to axes)
- Very high dimensional space! How do we learn feature weights?

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 $P(y|x)$ and then label an example with $\text{argmax}_y P(y|x)$

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

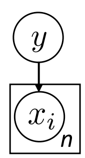
Bayes' Rule

constant: irrelevant for finding the max

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

"Naive" assumption:

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



linear model!

$$\text{argmax}_y P(y|x) = \text{argmax}_y \log P(y|x) = \text{argmax}_y \left[\log P(y) + \sum_{i=1}^{n} \log P(x_i|y)\right]$$

Text Classification: Ham or Spam

Hi, I just wanted to send over the latest results from training the LSTM model. In the attachment. What do you think of the performance?

hi i have very valuable business proposition for you. you make lots of $$$ I just need you to send a small amount of funds

Ham

Spam

???

$$\text{argmax}_y \log P(y|x) = \text{argmax}_y \left[\log P(y) + \sum_{i=1}^{n} \log P(x_i|y)\right]$$

$P(x_{\text{funds}} = 0|\text{spam}) = 0.9$ $\quad P(x_{\text{funds}} = 1|\text{spam}) = 0.1$

$P(x_{\text{funds}} = 0|\text{ham}) = 0.99$ $\quad P(x_{\text{funds}} = 1|\text{ham}) = 0.01$

spam gets more points in the final posterior

- Note that this is not $P(y|x)$ — not the probability of ham given the word

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

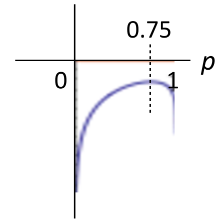
- Equivalent to maximizing logarithm of data likelihood:

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

Maximum Likelihood Estimation

- Imagine a coin flip which is heads with probability *p*
- Observe (H, H, H, T) and maximize log likelihood

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



- Maximum likelihood parameters for multinomial = read counts off of the data

Maximum Likelihood for Naive Bayes

Hi, I just wanted to send over the latest results from training the LSTM model. In the attachment. What do you think of the performance?

hi i have very valuable business proposition for you. you make lots of $$$ I just need you to send a small amount of funds

Ham

Spam

???

$P(y = \text{ham}) = 0.5$

$P(x_{\text{funds}} = 1|\text{spam}) = 1$ $\quad$ $P(x_{\text{funds}} = 0|\text{spam}) = 0$

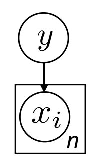
- Smoothing: add very small counts for each entry to avoid zeroes (bias-variance tradeoff)

$P(x_{\text{funds}} = 1|\text{spam}) = 0.99$ $\quad$ $P(x_{\text{funds}} = 0|\text{spam}) = 0.01$

Naive Bayes: Summary

- Model

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



- Inference

$$\text{argmax}_y \log P(y|x) = \text{argmax}_y \left[\log P(y) + \sum_{i=1}^{n} \log P(x_i|y) \right]$$

- Alternatively: $\log P(y = \text{spam}|x) - \log P(y = \text{ham}|x) > 0$

$$\Leftrightarrow \log \frac{P(y = \text{spam}|x)}{P(y = \text{ham}|x)} + \sum_{i=1}^{n} \log \frac{P(x_i|y = \text{spam})}{P(x_i|y = \text{ham})} > 0$$

- Learning: maximize $P(x, y)$ by reading counts off the data

Problems with Naive Bayes

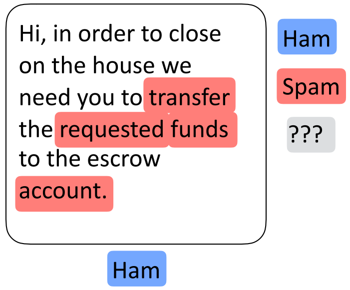
- Features are correlated

$P(x_{\text{funds}} = 1|\text{spam}) = 0.1$

$P(x_{\text{funds}} = 1|\text{ham}) = 0.01$

$P(x_{\text{transfer}} = 1|\text{spam}) = 0.1$

$P(x_{\text{transfer}} = 1|\text{ham}) = 0.01$



- This one sentence will make the probability of spam very high!
- Bad independence assumption in NB: these words are not independent!
- Solution: better model, algorithms that explicitly minimize loss rather than maximizing data likelihood

Generative vs. Discriminative Models

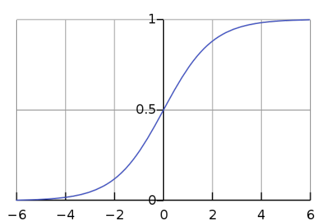
- Generative models: $P(x, y)$
  - Bayes nets / graphical models
  - Some of the model capacity goes to explaining the distribution of *x*; prediction uses Bayes rule post-hoc
  - Can sample new instances (*x, y*)
- Discriminative models: $P(y|x)$
  - SVMs, logistic regression, CRFs, most neural networks
  - Model is trained to be good at prediction, but doesn't model *x*
- We'll come back to this distinction throughout this class

Break!

Logistic Regression

$$P(y = \text{spam}|x) = \text{logistic}(w^\top x)$$

$$P(y = \text{spam}|x) = \frac{\exp\left(\sum_{i=1}^n w_i x_i\right)}{1 + \exp\left(\sum_{i=1}^n w_i x_i\right)}$$



- How to set the weights *w*?
- (Stochastic) gradient ascent to maximize log likelihood

$$\mathcal{L}(x_j, y_j = \text{spam}) = \log P(y_j = \text{spam}|x_j)$$
$$= \sum_{i=1}^n w_i x_{ji} - \log\left(1 + \exp\left(\sum_{i=1}^n w_i x_{ji}\right)\right)$$

Logistic Regression

$$\mathcal{L}(x_j, y_j = \text{spam}) = \log P(y_j|x_j) = \sum_{i=1}^n w_i x_{ji} - \log\left(1 + \exp\left(\sum_{i=1}^n w_i x_{ji}\right)\right)$$

$$\frac{\partial \mathcal{L}(x_j, y_j)}{\partial w_i} = x_{ji} - \frac{\partial}{\partial w_i}\log\left(1 + \exp\left(\sum_{i=1}^n w_i x_{ji}\right)\right)$$

$$= x_{ji} - \frac{1}{1 + \exp\left(\sum_{i=1}^n w_i x_{ji}\right)} \frac{\partial}{\partial w_i}\left(1 + \exp\left(\sum_{i=1}^n w_i x_{ji}\right)\right)$$ deriv of log

$$= x_{ji} - \frac{1}{1 + \exp\left(\sum_{i=1}^n w_i x_{ji}\right)} x_{ji} \exp\left(\sum_{i=1}^n w_i x_{ji}\right)$$ deriv of exp

$$= x_{ji} - x_{ji}\frac{\exp\left(\sum_{i=1}^n w_i x_{ji}\right)}{1 + \exp\left(\sum_{i=1}^n w_i x_{ji}\right)} = x_{ji}(1 - P(y_j = \text{spam}|x_j))$$

Logistic Regression

- Gradient of w_i on positive example $= x_{ji}(1 - P(y_j = \text{spam}|x_j))$

  If P(spam) is close to 1, make very little update

  Otherwise make w_i look more like x_{ji}, which will increase P(spam)

- Gradient of wi on negative example $= x_{ji}(-P(y_j = \text{spam}|x_j))$

  If P(spam) is close to 0, make very little update

  Otherwise make w_i look less like x_{ji}, which will decrease P(spam)

- Final gradient: $x_j(y_j - P(y_j = 1|x_j))$

Regularization

- Can end up making extreme updates to fit the training data

  w_{funds} = +1000

  $w_{transfer}$ = -900

  w_{send} = +742

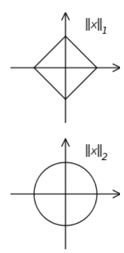
  w_{the} = +203

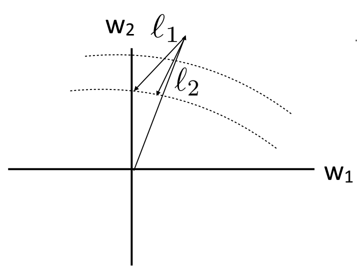
- All examples have P(correct) > 0.999, but classifier does crazy things on new examples

Regularization

- Can end up making extreme updates to fit the training data
- Rather than optimizing likelihood alone, impose a penalty on the norm of the weight vector (can also view as a Gaussian prior)
- Maximize

$$\sum_{j=1}^{m} \mathcal{L}(x_j, y_j) - \lambda \|w\|_2^2$$





Logistic Regression: Summary

- Model

$$P(y = \text{spam}|x) = \frac{\exp\left(\sum_{i=1}^{n} w_i x_i\right)}{1 + \exp\left(\sum_{i=1}^{n} w_i x_i\right)}$$

- Inference

  $\text{argmax}_y P(y|x)$ similar to Naive Bayes, but different model/learning

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

- Learning: gradient ascent on the (regularized) discriminative log-likelihood

Perceptron

- Simple error-driven learning approach similar to logistic regression
- Decision rule: $w^\top f(x) > 0$
  - If incorrect: if positive, $w \leftarrow w + x$

    if negative, $w \leftarrow w - x$

Logistic Regression

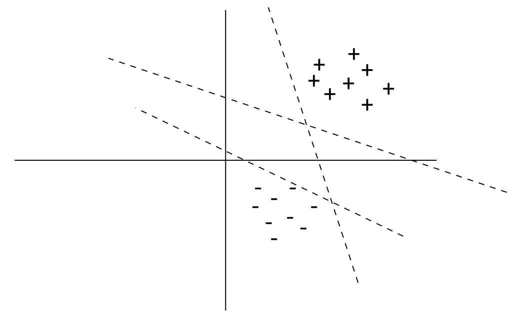
$w \leftarrow w + x(1 - P(y=1|x))$

$w \leftarrow w - xP(y=1|x)$

- Guaranteed to eventually separate the data if the data are separable, but does it learn a good boundary?

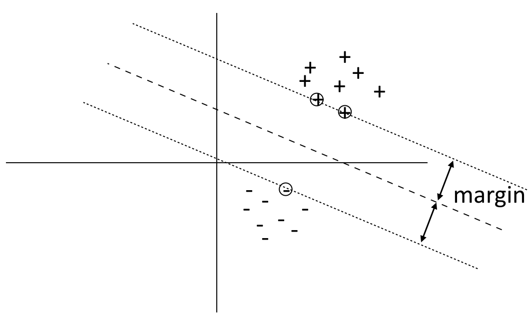
Support Vector Machines

- Many separating hyperplanes — is there a best one?



Support Vector Machines

- Many separating hyperplanes — is there a best one?



Support Vector Machines

- Constraint formulation: find *w* via following quadratic program:

Minimize $\|w\|_2^2$

s.t. $\forall j \quad w^\top x_j \geq 1 \text{ if } y_j = 1$

$\qquad w^\top x_j \leq -1 \text{ if } y_j = 0$

minimizing norm <=> maximizing margin

As a single constraint:

$$\forall j \quad y_j(w^\top x_j) + (1 - y_j)(-w^\top x_j) \geq 1$$

$$\longrightarrow \forall j \quad (2y_j - 1)(w^\top x_j) \geq 1$$

- What's wrong with this quadratic program for real data?
- Data is generally non-separable — need slack!

N-Slack SVMs

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

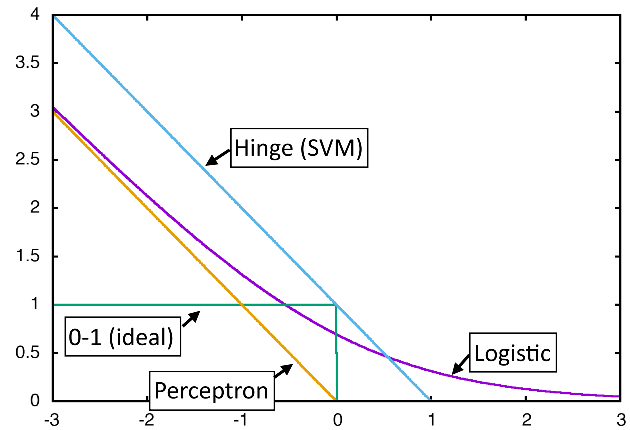
s.t. $\forall j \quad (2y_j - 1)(w^\top x_j) \geq 1 - \xi_j \qquad \forall j \quad \xi_j \geq 0$

- The ξ_j are a "fudge factor" to make all constraints satisfied
- (Sub-)gradient descent: focus on second part of objective

$$\frac{\partial}{\partial w_i}\xi_j = 0 \text{ if } \xi_j = 0 \qquad \frac{\partial}{\partial w_i}\xi_j = (2y_j - 1)x_{ji} \text{ if } \xi_j > 0$$
$$= x_{ji} \text{ if } y_j = 1, \; -x_{ji} \text{ if } y_j = 0$$

- Looks like the perceptron! But updates more frequently

Loss Functions



Optimization — next time...

- Haven't talked about optimization at all
- Range of techniques from simple gradient descent (works pretty well) to more complex methods (can work better)

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

Recap

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

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

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

- SVM:

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

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

Recap

- Logistic regression, SVM, and perceptron are closely related
- SVM and perceptron inference require taking maxes, logistic regression has a similar update but is "softer" due to its probabilistic nature
- All gradient updates: "make it look more like the right thing and less like the wrong thing"