Logistic regression for conditional probability estimation

Uncertainty in prediction

Can we usually expect to get a perfect classifier, if we have enough training data?

Uncertainty in prediction

Can we usually expect to get a perfect classifier, if we have enough training data?

Problem 1: Inherent uncertainty

The available features x simply do not contain enough information to perfectly predict y, e.g.,

- x = a heart patient's complete medical record
- y = will he/she survive another year?

Uncertainty in prediction, cont'd

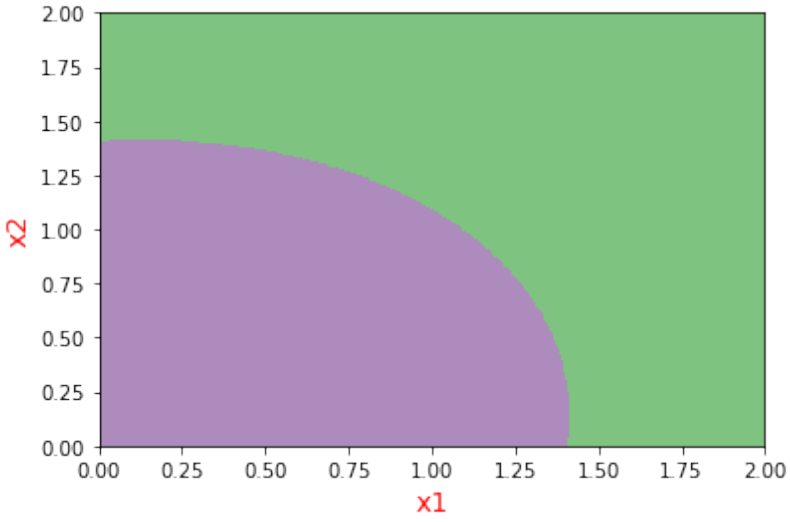
Can we usually expect to get a perfect classifier, if we have enough training data?

Uncertainty in prediction, cont'd

Can we usually expect to get a perfect classifier, if we have enough training data?

Problem 2: Limitations of the model class

The type of classifier being used does not capture the decision boundary, e.g. using linear classifiers with:



Conditional probability estimation for binary labels

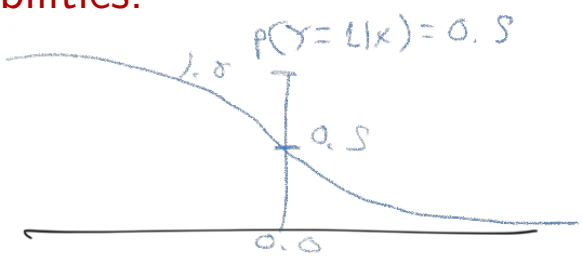
Given: a data set of pairs (x, y), where $x \in \mathbb{R}^d$ and $y \in \{-1, 1\}$.

- Return a classifier that also gives probabilities.
- That is, it gives $\Pr(y = 1|x)$.

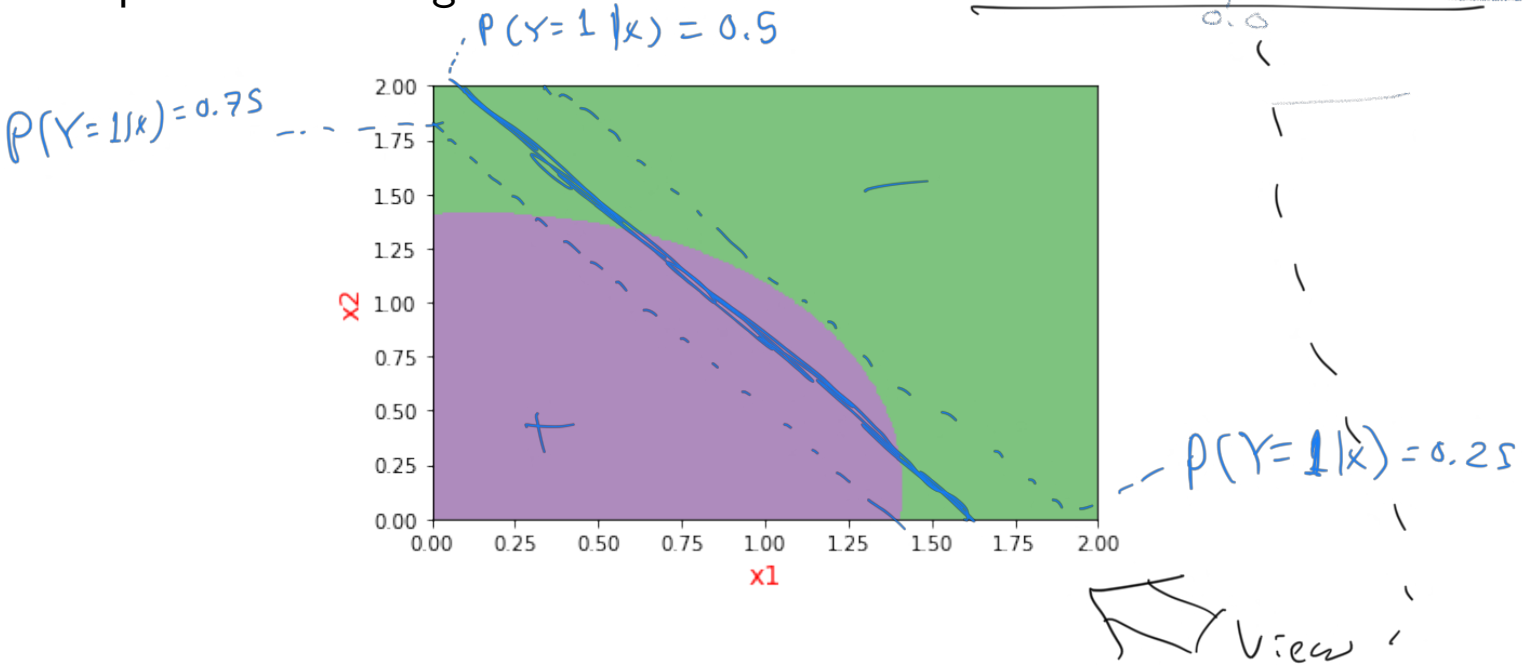
Conditional probability estimation for binary labels

Given: a data set of pairs (x, y), where $x \in \mathbb{R}^d$ and $y \in \{-1, 1\}$.

- Return a classifier that also gives probabilities.
- That is, it gives $\Pr(y = 1|x)$.



Simplest case: using a linear function of x.



A linear model for conditional probability estimation

Classify and return probabilities using a linear function

$$w_o + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d.$$

The probability of $y = 1$:

- Increases as the linear function grows.
- Is 50% when this linear function is zero.



A linear model for conditional probability estimation

Classify and return probabilities using a linear function

$$w_o + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d.$$

The probability of $y = 1$:

- Increases as the linear function grows.
- Is 50% when this linear function is zero.

As before, add another feature $x_o \equiv 1$ so we can forget w_o.
Now linear function is $w \cdot x$.

A linear model for conditional probability estimation

Classify and return probabilities using a linear function

$$w_o + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d.$$

The probability of $y = 1$:

- Increases as the linear function grows.
- Is 50% when this linear function is zero.

As before, add another feature $x_o \equiv 1$ so we can forget w_o.
Now linear function is $w \cdot x$.

- $w \cdot x = 0 \implies \Pr(y = 1|x) = 1/2.$
- $w \cdot x \uparrow \infty \implies \Pr(y = 1|x) \uparrow 1.$
- $w \cdot x \downarrow -\infty \implies \Pr(y = 1|x) \downarrow 0.$

A linear model for conditional probability estimation

Classify and return probabilities using a linear function

$$w_o + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d.$$

The probability of $y = 1$:

- Increases as the linear function grows.
- Is 50% when this linear function is zero.

As before, add another feature $x_o \equiv 1$ so we can forget w_o.
Now linear function is $w \cdot x$.

- $w \cdot x = 0 \implies \Pr(y = 1|x) = 1/2.$
- $w \cdot x \uparrow \infty \implies \Pr(y = 1|x) \uparrow 1.$
- $w \cdot x \downarrow -\infty \implies \Pr(y = 1|x) \downarrow 0.$

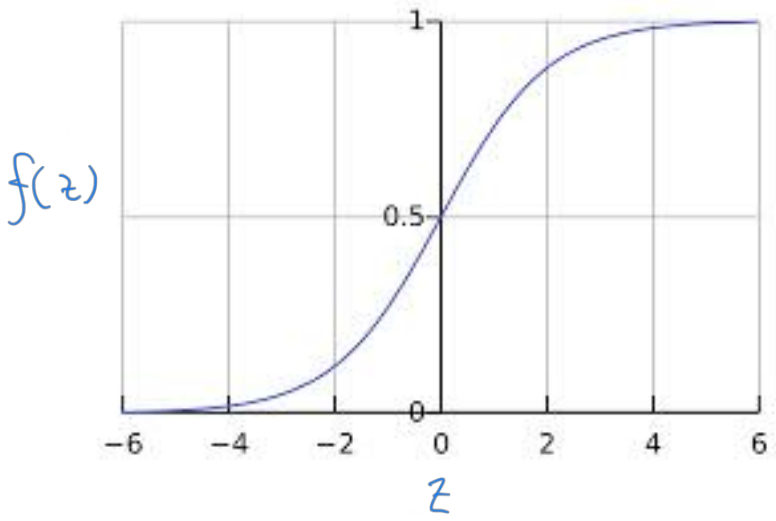
How can we convert $w \cdot x$ into a probability?

The logistic regression model

$$\Pr(y = 1|x) = \frac{1}{1 + e^{-w \cdot x}}.$$

$f(z) =$

The squashing function $1/(1 + e^{-z})$:



The conditional probability function

For some vector w, we have

$$\Pr(y = 1|x) = \frac{1}{1 + e^{-w \cdot x}}.$$

Therefore,

$$\Pr(y = -1 \mid x) \;=\; 1 - \frac{1}{1 + e^{-w \cdot x}} \;=\; \frac{e^{-w \cdot x}}{1 + e^{-w \cdot x}} \;=\; \frac{1}{1 + e^{w \cdot x}}$$

The conditional probability function

For some vector w, we have

$$\Pr(y = 1|x) = \frac{1}{1 + e^{-w \cdot x}}.$$

Therefore,

$$\Pr(y = -1 \mid x) \;=\; 1 - \frac{1}{1 + e^{-w \cdot x}} \;=\; \frac{e^{-w \cdot x}}{1 + e^{-w \cdot x}} \;=\; \frac{1}{1 + e^{w \cdot x}}$$

So we can write

$$\Pr_w(y \mid x) \;=\; \frac{1}{1 + e^{-y(w \cdot x)}}$$

Choosing w

The maximum-likelihood principle: given a data set

$$(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, 1\},$$

pick the $w \in \mathbb{R}^d$ that maximizes

$$\prod_{i=1}^{n} \Pr_w(y^{(i)} \mid x^{(i)}).$$

Choosing w

The maximum-likelihood principle: given a data set

$$(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, 1\},$$

pick the $w \in \mathbb{R}^d$ that maximizes

$$\prod_{i=1}^{n} \Pr_w(y^{(i)} \mid x^{(i)}).$$

Easier to work with sums, so take log to get **loss function**

$$L(w) \;=\; -\sum_{i=1}^{n} \ln \Pr_w(y^{(i)} \mid x^{(i)}) \;=\; \sum_{i=1}^{n} \ln(1 + e^{-y^{(i)}(w \cdot x^{(i)})})$$

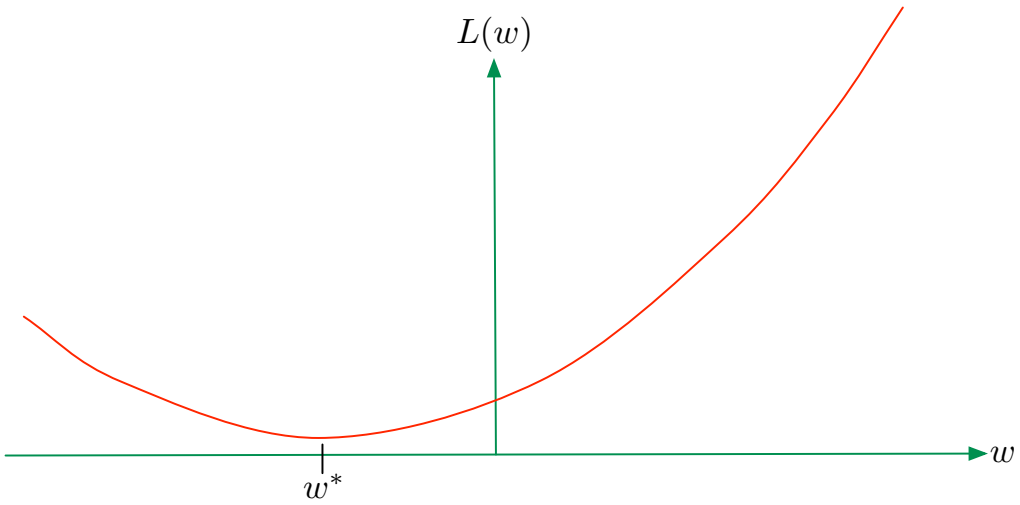
Our goal is to minimize $L(w)$.

Convexity

- The bad news: no closed-form solution for w

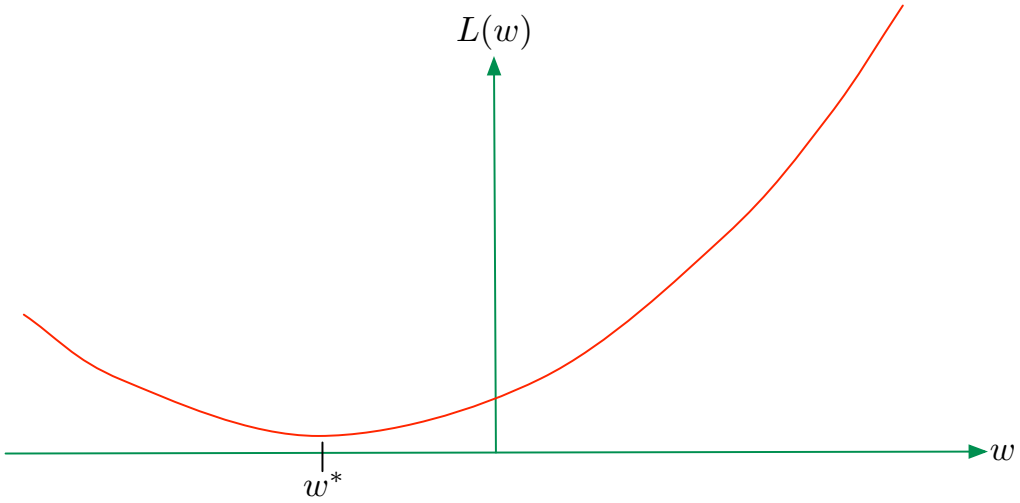
Convexity

- The bad news: no closed-form solution for w
- The good news: $L(w)$ is **convex** in w.



Convexity

- The bad news: no closed-form solution for w
- The good news: $L(w)$ is **convex** in w.



How to find the minimum of a convex function? By **local search.**

Gradient descent procedure for logistic regression

Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, 1\}$, find

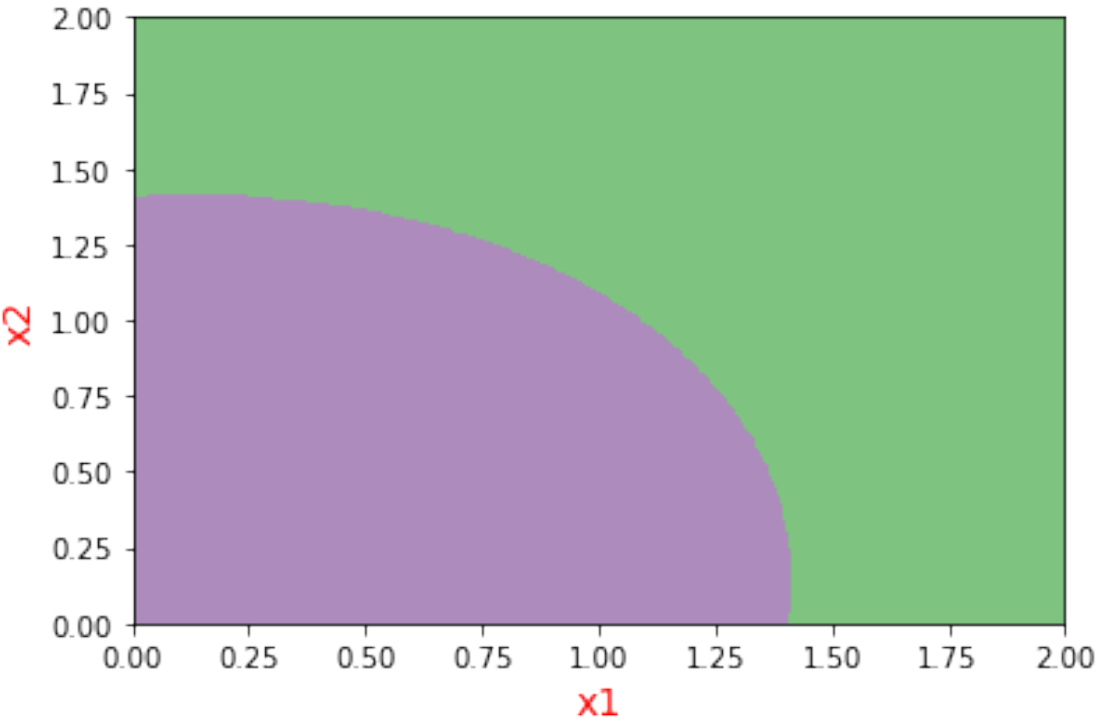
$$\arg\min_{w \in \mathbb{R}^d} L(w) \;=\; \sum_{i=1}^{n} \ln(1 + e^{-y^{(i)}(w \cdot x^{(i)})})$$

- Set $w_0 = 0$
- For $t = 0, 1, 2, \ldots$, until convergence:

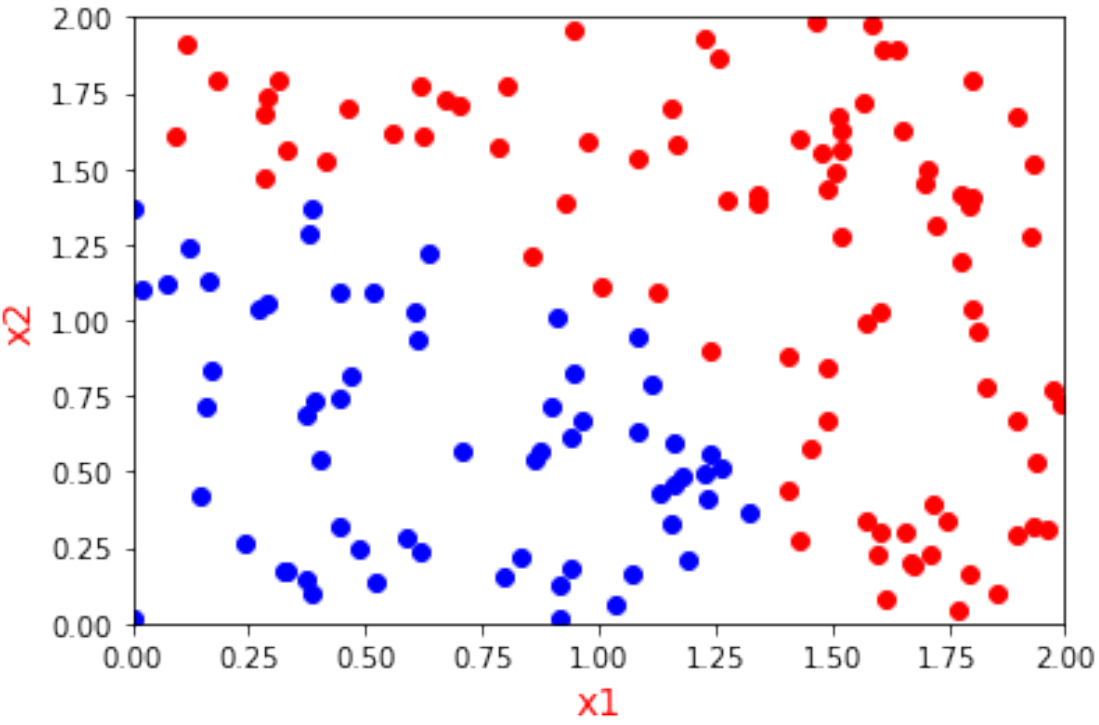
$$w_{t+1} \;=\; w_t \;+\; \eta_t \sum_{i=1}^{n} y^{(i)} x^{(i)} \underbrace{\Pr_{w_t}(-y^{(i)} | x^{(i)})}_{\text{doubt}_t(x^{(i)}, y^{(i)})},$$

where η_t is a step size that is fixed beforehand, or (for instance) chosen by line search to minimize $L(w_{t+1})$.

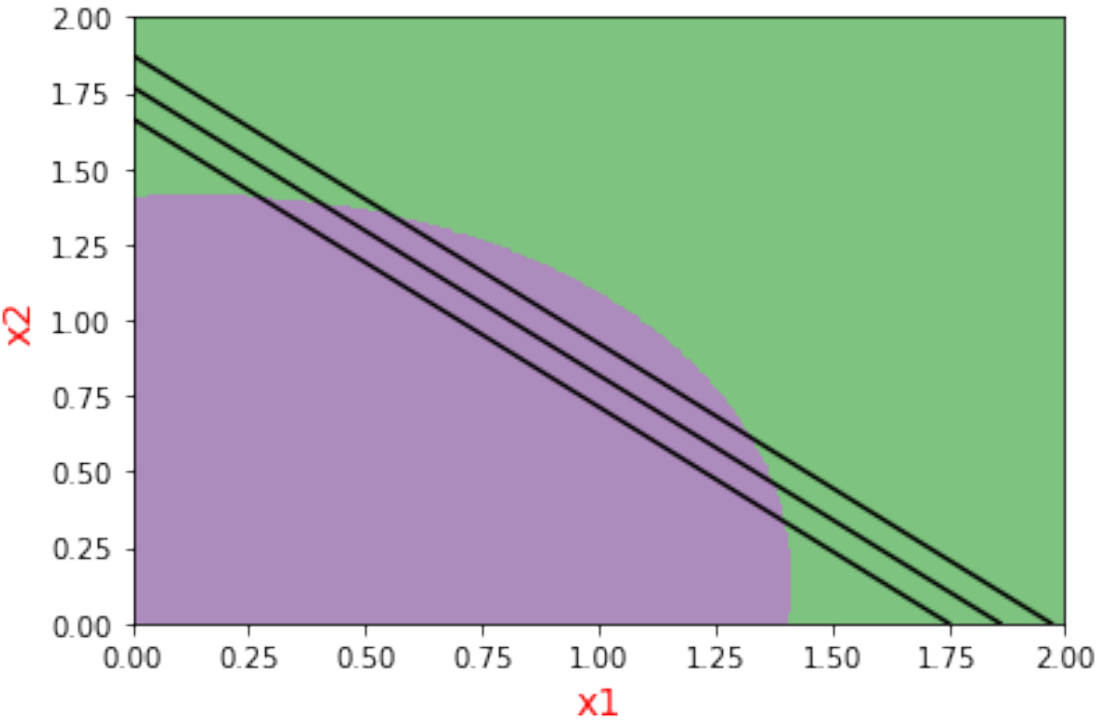
Toy example


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
x2
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
x1

Toy example


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
x2
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
x1

Toy example


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
x2
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
x1

Sentiment data

Data set: sentences from reviews on Amazon, Yelp, IMDB, each labeled as positive or negative.

Sentiment data

Data set: sentences from reviews on Amazon, Yelp, IMDB, each labeled as positive or negative.

- `Needless to say, I wasted my money.`
- `He was very impressed when going from the original battery to the extended battery.`

Sentiment data

Data set: sentences from reviews on Amazon, Yelp, IMDB, each labeled as positive or negative.

- `Needless to say, I wasted my money.`
- `He was very impressed when going from the original battery to the extended battery.`
- `I have to jiggle the plug to get it to line up right to get decent volume.`
- `Will order from them again!`

Sentiment data

Data set: sentences from reviews on Amazon, Yelp, IMDB, each labeled as positive or negative.

- `Needless to say, I wasted my money.`
- `He was very impressed when going from the original battery to the extended battery.`
- `I have to jiggle the plug to get it to line up right to get decent volume.`
- `Will order from them again!`

2500 training sentences, 500 test sentences

Handling text data

Bag-of-words: vectorial representation of text documents.

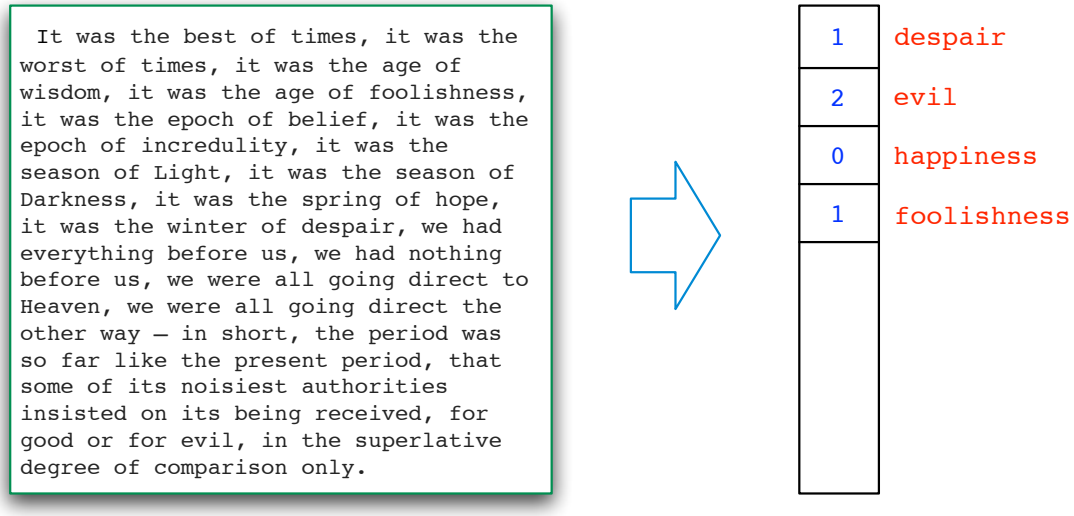
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way – in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.



1	despair
2	evil
0	happiness
1	foolishness

Handling text data

Bag-of-words: vectorial representation of text documents.



- Fix V = some vocabulary.
- Treat each document as a vector of length $|V|$:

$$x = (x_1, x_2, \ldots, x_{|V|}),$$

where $x_i = \#$ of times the ith word appears in the document.

A logistic regression approach

Code positive as $+1$ and negative as -1.

$$\Pr_w(y \mid x) = \frac{1}{1 + e^{-y(w \cdot x)}}$$

Maximum-likelihood fit: Given training data

$$(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, 1\},$$

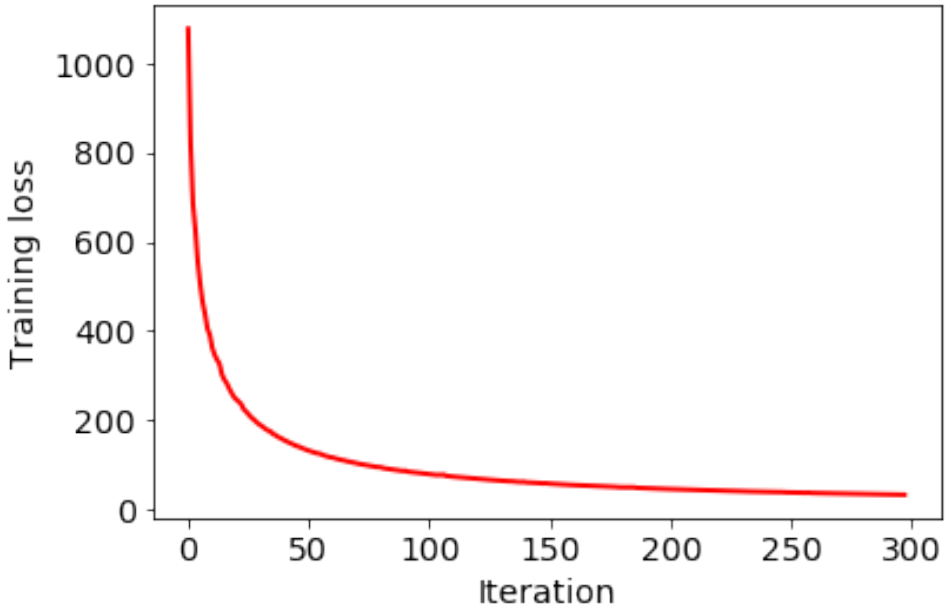
find the w that minimizes

$$L(w) = \sum_{i=1}^{n} \ln(1 + e^{-y^{(i)}(w \cdot x^{(i)})})$$

Convex problem: many solution methods, including gradient descent, stochastic gradient descent, Newton-Raphson, quasi-Newton, etc., will converge to the optimal w.

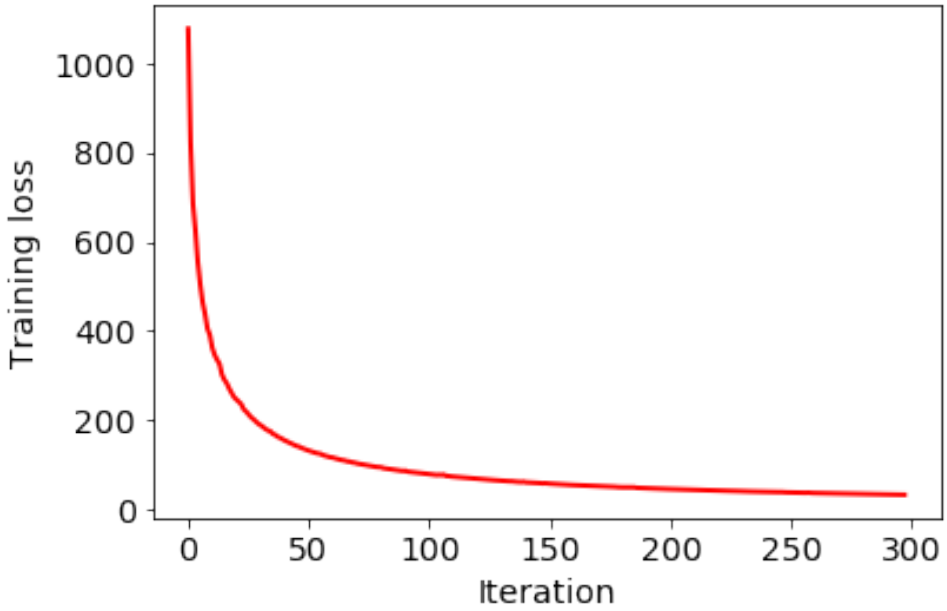
Local search in progress

Look at how loss function $L(w)$ changes over iterations of stochastic gradient descent.



Local search in progress

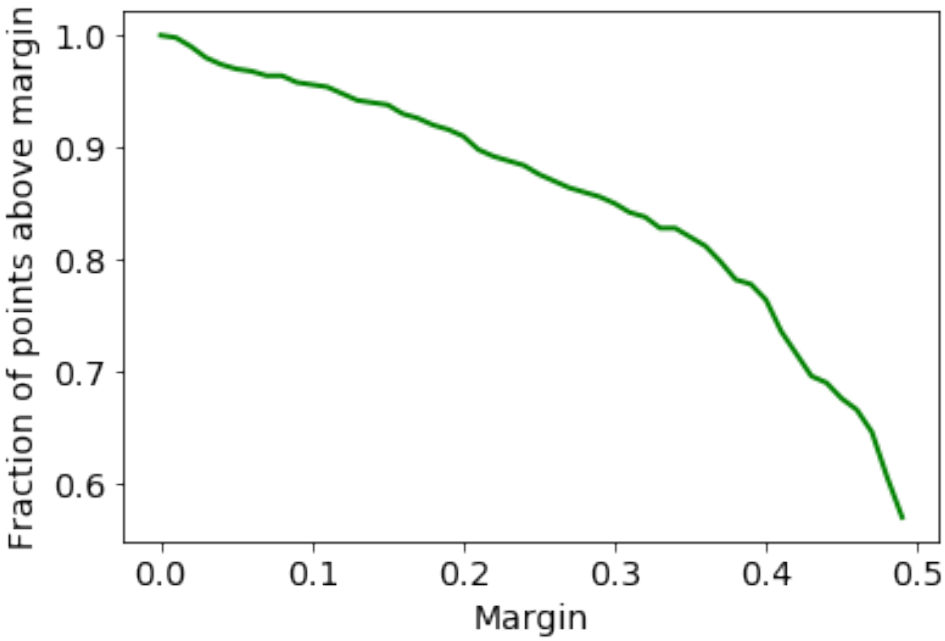
Look at how loss function $L(w)$ changes over iterations of stochastic gradient descent.



Final w: **test error** 0.21.

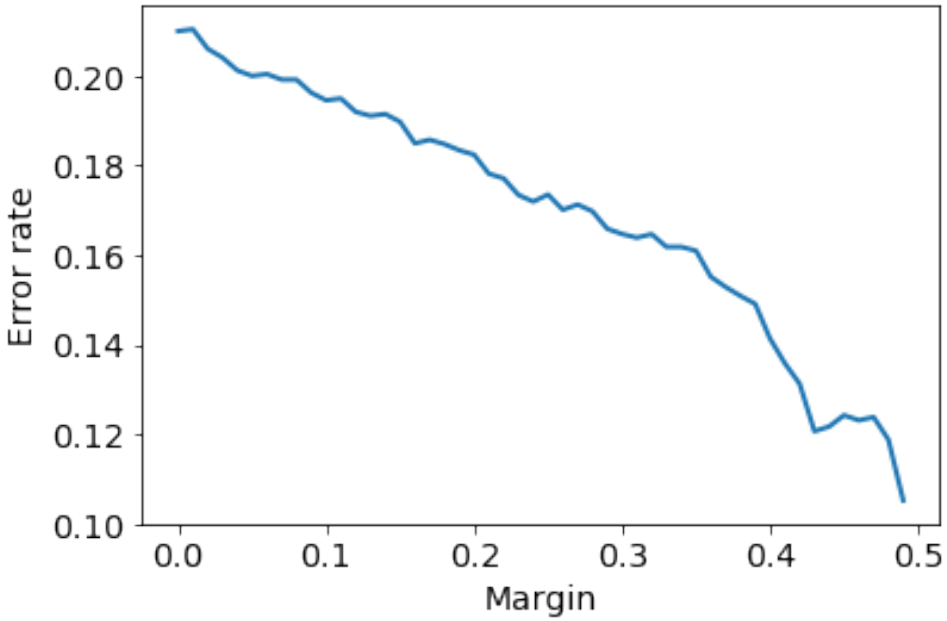
Margin and test error

$$\text{Margin on test pt } x = \left| \Pr_w(y = 1|x) - \frac{1}{2} \right|.$$



Margin and test error

$$\text{Margin on test pt } x = \left| \Pr_w(y = 1|x) - \frac{1}{2} \right|.$$



Some of the mistakes

Not much dialogue, not much music, the whole film was shot as elaborately and aesthetically like a sculpture. 1

This film highlights the fundamental flaws of the legal process, that it's not about discovering guilt or innocence, but rather, is about who presents better in court. 1

The last 15 minutes of movie are also not bad as well. 1

You need two hands to operate the screen. This software interface is decade old and cannot compete with new software designs. -1

If you plan to use this in a car forget about it. -1

If you look for authentic Thai food, go else where. -1

Waste your money on this game. 1

Interpreting the model

Words with the most positive coefficients

'sturdy', 'able', 'happy', 'disappoint', 'perfectly', 'remarkable', 'animation', 'recommendation', 'best', 'funny', 'restaurant', 'job', 'overly', 'cute', 'good', 'rocks', 'believable', 'brilliant', 'prompt', 'interesting', 'skimp', 'definitely', 'comfortable', 'amazing', 'tasty', 'wonderful', 'excellent', 'pleased', 'beautiful', 'fantastic', 'delicious', 'watch', 'soundtrack', 'predictable', 'nice', 'awesome', 'perfect', 'works', 'loved', 'enjoyed', 'love', 'great', 'happier', 'properly', 'liked', 'fun', 'screamy', 'masculine'

Words with the most negative coefficients

'disappointment', 'sucked', 'poor', 'aren', 'not', 'doesn', 'worst', 'average', 'garbage', 'bit', 'looking', 'avoid', 'roasted', 'broke', 'starter', 'disappointing', 'dont', 'waste', 'figure', 'why', 'sucks', 'slow', 'none', 'directing', 'stupid', 'lazy', 'unrecommended', 'unreliable', 'missing', 'awful', 'mad', 'hours', 'dirty', 'didn', 'probably', 'lame', 'sorry', 'horrible', 'fails', 'unfortunately', 'barking', 'bad', 'return', 'issues', 'rating', 'started', 'then', 'nothing', 'fair', 'pay'