CSE 519: Data Science
Steven Skiena
Stony Brook University

Lecture 18: Logistic Regression and Classification

Classification Problems

Often we are given collections of examples *labeled* by class:

- male / female?
- democrat / republican?
- spam / non-spam (ham)?
- cancer / benign?

Classification assigns a label to an input record.

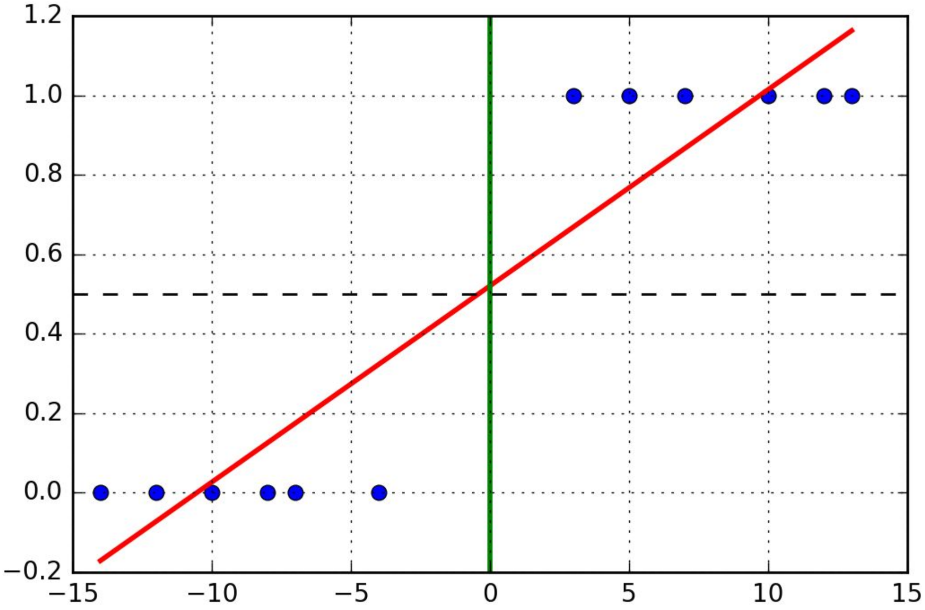
Regression for Classification

We *could* use linear regression to build from annotated examples by converting the class names to numbers:

- male=0 / female=1
- democrat=0 / republican=1
- spam=1 / non-spam=0
- cancer=1 / benign=0

Zero/one works for binary classifiers. By convention, the "positive" class gets 1 and the "negative" one 0.

Class Labels from Regression Lines

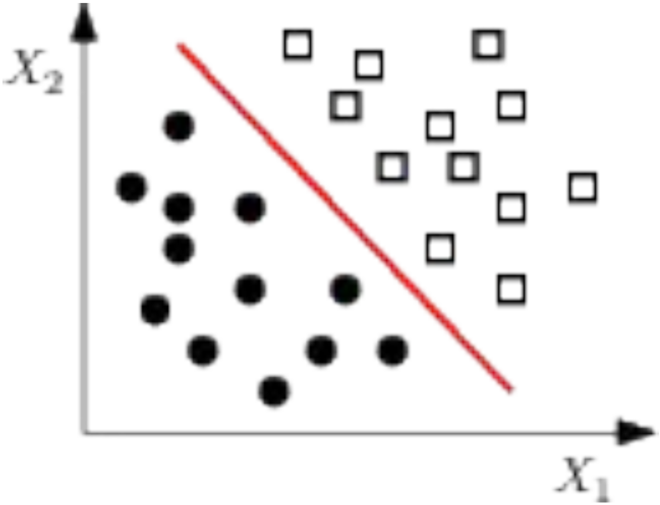


The regression line will cut through these classes, even through a separator exists.

Adding very +/- examples shifts the line but hurts the boundary

Decision Boundaries

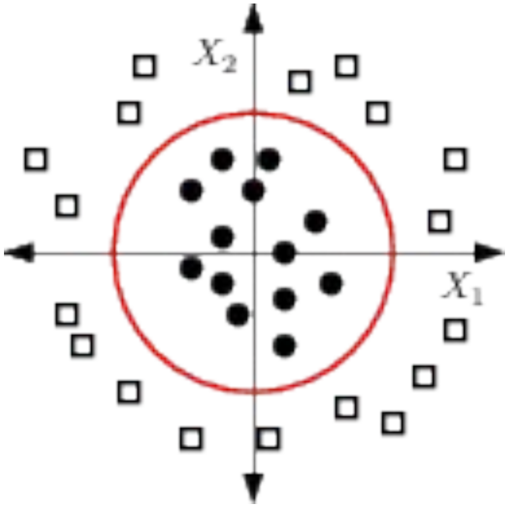
Ideally, our two classes will be well-separated in feature space, so a line can partition them.



Logistic regression is a method to find the best separating line for a given training set.

Non-Linear Decision Boundaries

Logistic regression can find non-linear boundaries if seeded with non-linear features.



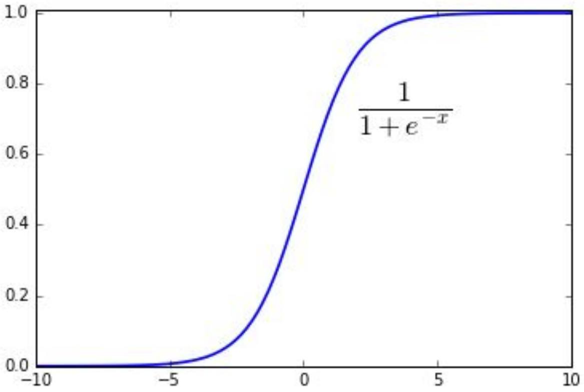
To get separating circles, we need to explicitly add features like x^2 and x1 * x2.

The Logit Function

We want a way to take a real variable and convert it to a probability:

$$f(x) = \frac{1}{1 + e^{-cx}}$$

- f(0) = ½
- f(infinity) = 1
- f(-infinity) = 0



Logit can give the probability of x being in a particular class.

Logit for Classification

To extend logit to m variables, and set the threshold and steepness parameters, fit

$$h(x, w) = w_0 + \sum_{i=1}^{m-1} w_i \cdot x_i$$

Plug into logit to get a classifier:

$$f(x) = \frac{1}{1 + e^{-h(x,w)}}$$

Scoring for Logistic Regression

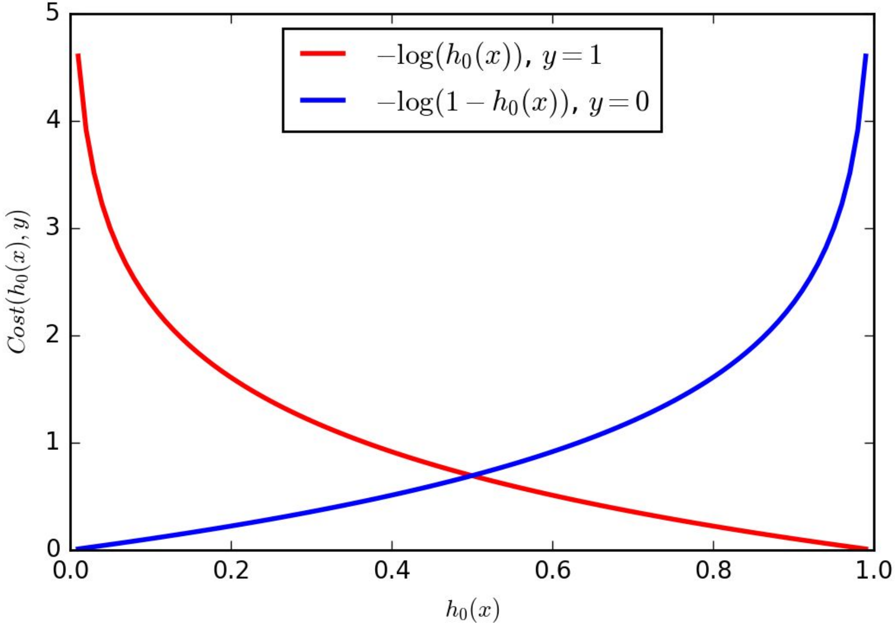
Assume each of our n training examples are labeled as to being in either class 0 or 1.

We seek to identify the coefficients w that give high probability on positive (class 1) items, and low probability on negative (class 0) items.

We need a cost function to value a probability p for an item of class c.

Costs for Positive / Negative Cases

We want zero error for the best probability, and increasing cost as the class prediction gets more and more wrong.



Logarithms of Probabilities

Observe that log(1) = 0. Thus it gives proper cost for correct predictions of positive items.

Observe that $\log(0) \to -\infty$ The cost function:

$$cost(x_i, 1) = -\log(f(x_i))$$

For negative instances, the cost function is:

$$cost(x_i, 0) = -\log(1 - f(x_i))$$

Cost/Loss Function

We can use the zero/one labels as indicator variables to compute the loss function:

$$J(w) = \frac{1}{n}\sum_{i=1}^{n} cost(f(x_i, w), y_i)$$

$$= -\frac{1}{n}[\sum_{i=1}^{n} y_i \log f(x_i, w) + (1 - y_i) \log(1 - f(x_i, w))$$

This works because the class indicator variables are 0 and 1.

Logistic Regression via Gradient Descent

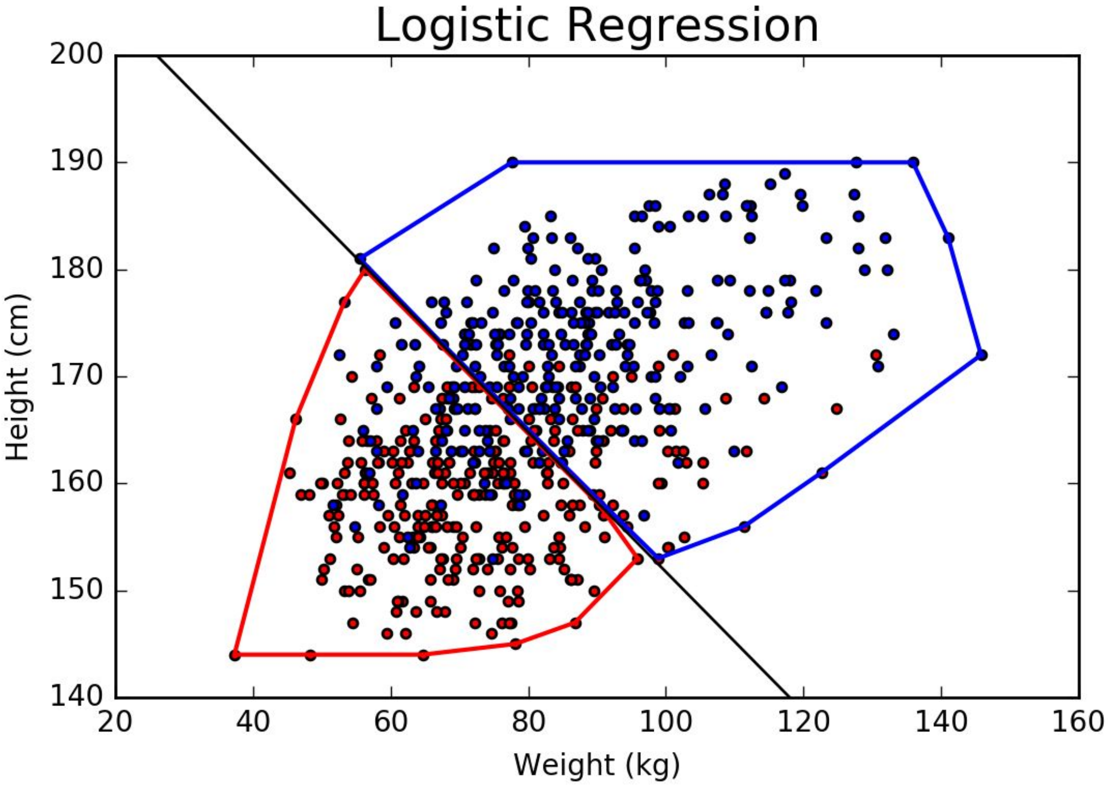
The loss function here is convex, so we can find the parameters which best fit it by gradient descent.

Thus we can find the best linear separator between two classes (linear in the possibly non-linear features).

Logistic Gender Classification

Red region:
229 w / 63 m

Blue region:
223 m / 65 w



Issues in Logistic Classification

- Balanced training class sizes
- Multi-class classification
- Hierarchical classification
- Partition functions

Balanced Training Classes

Consider the optimal separating line for grossly unbalanced class sizes, say 1 positive example vs. 1,000,000 negative examples.

The best scoring line from logistic regression will try to be very far from the big cluster instead of the midpoint between them.

Use equal numbers of pos and neg examples.

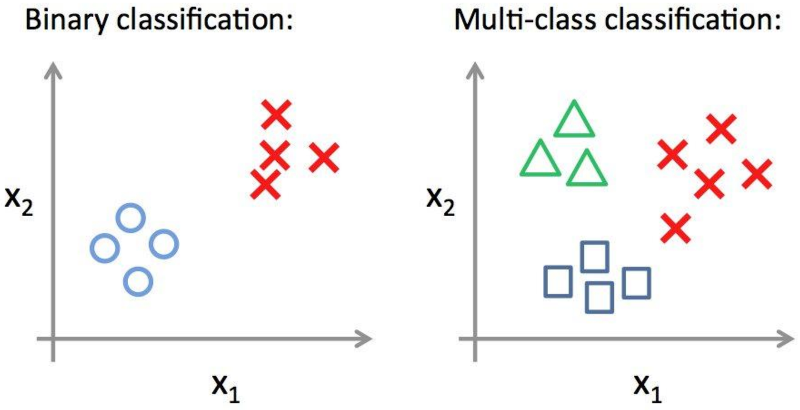
Ways to Balance Classes

- Work harder to find members of the minority class.
- Discard elements from the bigger class.
- Weigh the minority class more heavily, but beware of overfitting.
- Replicate members of the smaller class, ideally with random perturbation.

Multi-class Classification

Classification tasks are not always binary.

Is a given movie a comedy, drama, action, documentary, etc.?



Binary classification:

Multi-class classification:

Encoding Multi-Classes: Bad Idea

It is natural to represent multiple classes by distinct numbers: blond=0, brown=1, red=2.

But unless the ordering of the classes reflect an increasing scale, the numbering is meaningless.

Classes on a Likert scale are ordinal.

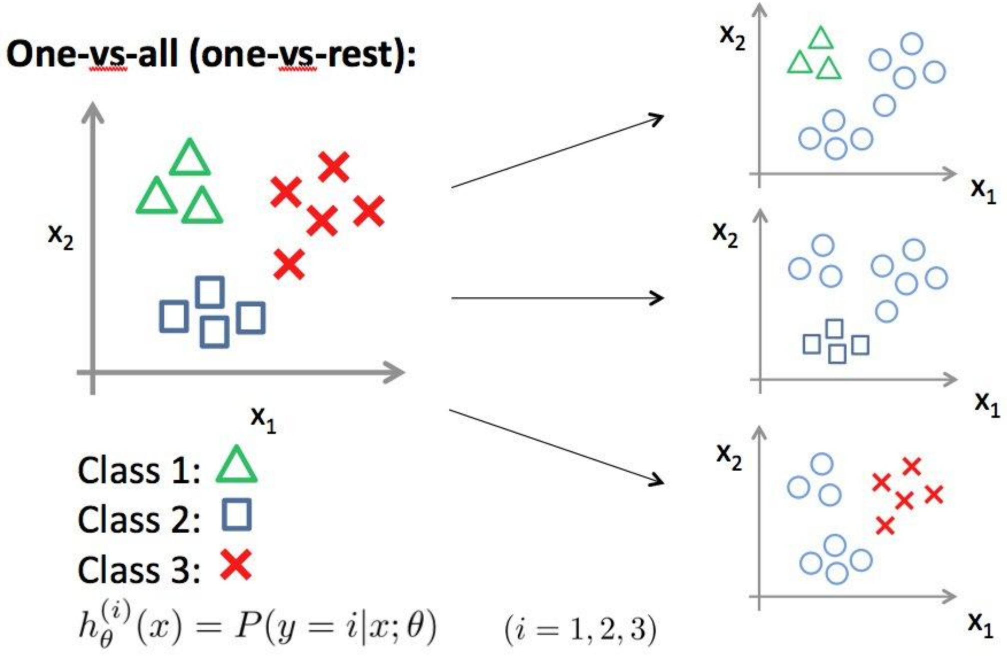
- Completely Agree
- Mostly Agree
- Slightly Agree
- Slightly Disagree
- Mostly Disagree
- Completely Disagree

4 stars - 3 stars - 2 stars - 1 star

One Versus All Classifiers

We can build multi-class classifiers by building multiple independent binary classifiers.

Select the class of highest probability as the predicted label.


One-vs-all (one-vs-rest):

Class 1: △
Class 2: □
Class 3: ✕

$h_\theta^{(i)}(x) = P(y = i|x;\theta) \qquad (i = 1, 2, 3)$

The Challenges of Many Classes

Multi-class classification gets much harder as the number of classes increase.

The monkey is right only *1/k* times for *k* classes.

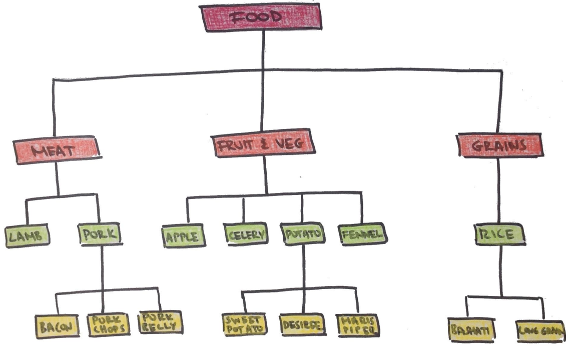
- With many available classes, certain mistakes are more acceptable than others.
- The actual population sizes of rare classes can easily become vanishingly small.

Hierarchical Classification

Grouping classes by similarity and building a taxonomy reduces the effective number of classes.

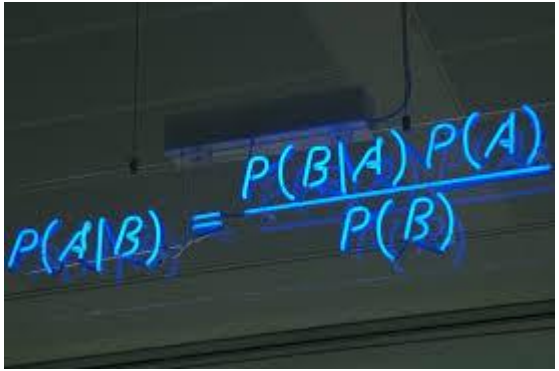
Imagenet has 27 categories (e.g. animal, appliance, food, furniture) on top of 21,841 subcategories.

Classify from top-down in tree.



Bayesian Priors

Having honest estimates for the probability distribution of classes is essential to avoid domination by rare classes ("rock stars")



Here A is the given class, and B is the event your classifier returned a guess of class A.

Partition Functions

There is no reason why independent binary classifiers yield probabilities which sum to 1.

To get real probabilities for Bayes theorem:

$$T = \sum_{i=1}^{c} F_i(x)$$

and *P(c)* = *F_c(x)* / *T*.