Logistic Regression

CSC 461: Machine Learning

Fall 2020

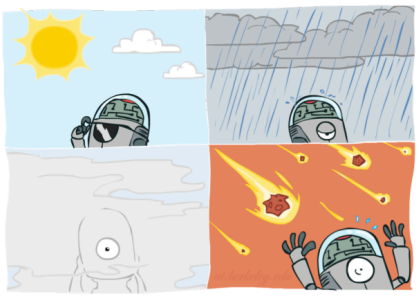
Prof. Marco Alvarez
University of Rhode Island

So far …

- Linear methods for classification
  - ✓ discrete outputs (perceptron)
- What if probabilistic outputs are needed?

Review of basic probability

Probability distributions



$P(W)$

W	P
sun	0.6
rain	0.1
fog	0.3
meteor	0.0

Random variable, domain, and probabilities

https://inst.eecs.berkeley.edu/~cs188/fa19/assets/slides/lec13.pdf

Joint distribution

- A distribution over a **set of random variables**
- Specifies probabilities for each **outcome**
- **Normalized**: sum to 1
- **Event** is a set of outcomes

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

P(hot)? .5

P(sun)? .6

P(hot,rain)? .1

Marginal distribution

$P(X, Y)$

X	Y	P
+x	+y	0.2
+x	-y	0.3
-x	+y	0.4
-x	-y	0.1

$P(x) = \sum_y P(x, y)$

$P(y) = \sum_x P(x, y)$

$P(X)$

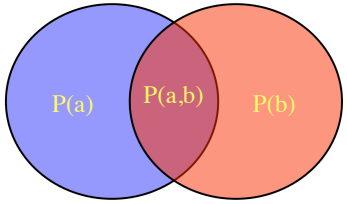
X	P
+x	
-x	

$P(Y)$

Y	P
+y	
-y	

probability distribution of a subset of (marginal) random variables

Conditional probabilities



$$P(a|b) = \frac{P(a, b)}{P(b)}$$

$P(X, Y)$

X	Y	P
+x	+y	0.2
+x	-y	0.3
-x	+y	0.4
-x	-y	0.1

P(+x|+y)?.2/.6 P(-x|+y)?.4/.6 P(-y|+x)?.3/.5

Conditional distribution

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

SELECT the joint probabilities matching the evidence

$P(c, W)$

T	W	P
cold	sun	0.2
cold	rain	0.3

NORMALIZE the selection (make it sum to one)

$P(W|T = c)$

W	P
sun	0.4
rain	0.6

$P(X, Y)$

X	Y	P
+x	+y	0.2
+x	-y	0.3
-x	+y	0.4
-x	-y	0.1

SELECT the joint probabilities matching the evidence

NORMALIZE the selection (make it sum to one)

Logistic regression

Basics

$$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(n)}, y^{(n)})\}$$

$$\mathbf{x}^{(i)} \in \mathbb{R}^d, y^{(i)} \in \{-1, +1\}$$

Logistic regression

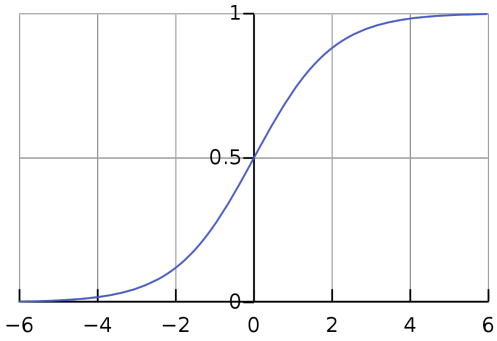
- Binary classification
  - ✓ similar to the perceptron but uses a **logistic function** (type of sigmoid function, S-shaped)
  - ✓ models **probability** of output in terms of input
- It is considered a **linear classifier**
  - ✓ even though the 'activation' function is non-linear
- It is a **discriminative model**
  - ✓ models decision boundary directly, $P(y \mid \mathbf{x})$ in this case

Big picture

- Define a hypothesis (classifier)
- Define a loss function
- Optimize with gradient descent
- Predict the class with highest probability using the learned hypothesis

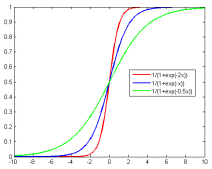
Logistic function

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



mapping $\mathbb{R}$ to [0,1]

continuous and differentiable



https://alliance.seas.upenn.edu/~cis520/wiki/index.php?n=Lectures.Logistic

Probabilistic interpretation

$$h_{\mathbf{w}}(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$$

(probability of class +1) $$P(y=+1 \mid \mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$$

$$P(y=-1 \mid \mathbf{x}) = 1 - P(y=+1 \mid \mathbf{x})$$

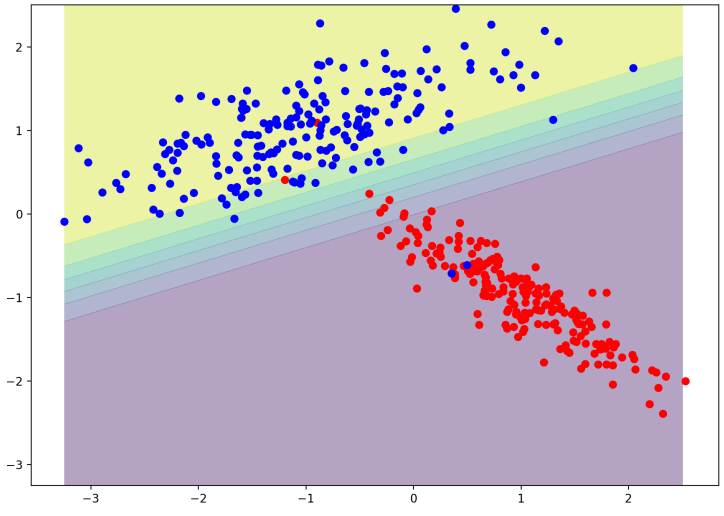
(probability of class -1) $$P(y=-1 \mid \mathbf{x}) = \sigma(-\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{\mathbf{w}^T\mathbf{x}}}$$

Decision boundary

$$P(y=+1 \mid \mathbf{x}) = P(y=-1 \mid \mathbf{x}) = 0.5$$

Logistic regression has a linear decision boundary $\mathbf{w}^T\mathbf{x} = 0$

Linear decision boundary



Loss function

$$L(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}}\right)$$

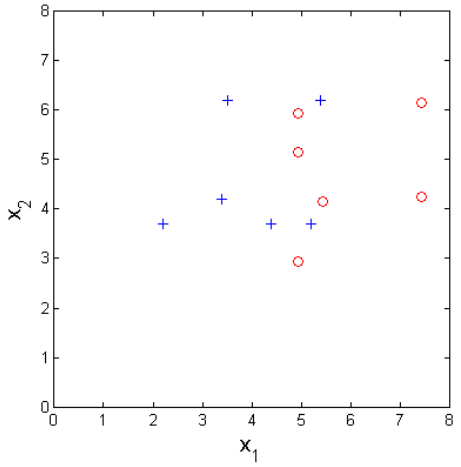
Logistic regression

Cross-entropy loss

will be derived later … when covering MLE

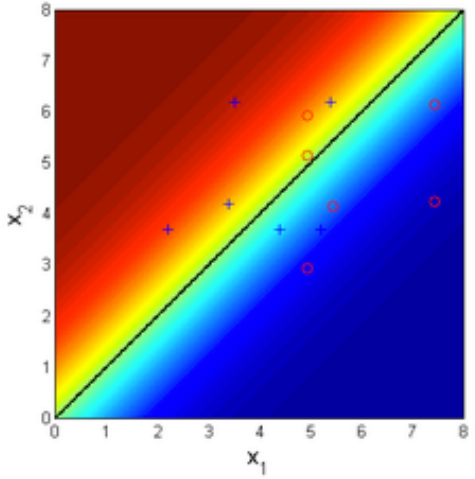
- no closed-form solution, but loss is convex
- can use gradient descent (or stochastic) or second-order methods

Example: 2d dataset



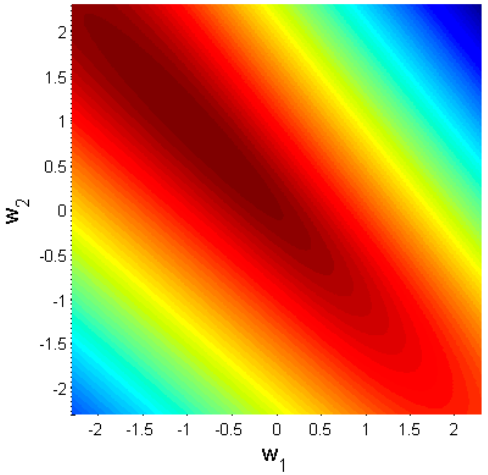
https://alliance.seas.upenn.edu/~cis520/wiki/index.php?n=Lectures.Logistic

Solution



https://alliance.seas.upenn.edu/~cis520/wiki/index.php?n=Lectures.Logistic

Example: loss function



plot shows contour lines in the space of parameters w_1 and w_2, w_0 is omitted

https://alliance.seas.upenn.edu/~cis520/wiki/index.php?n=Lectures.Logistic

Gradient

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = \left[\frac{\partial L(\mathbf{w})}{\partial w_0}, \ldots, \frac{\partial L(\mathbf{w})}{\partial w_d} \right]$$

$$\frac{\partial L(\mathbf{w})}{\partial w_j} = \frac{\partial}{\partial w_j} \frac{1}{n} \sum_{i=1}^{n} \log \left(1 + e^{-y^{(i)} \mathbf{w}^T \mathbf{x}^{(i)}} \right)$$

$$= -\frac{1}{n} \sum_{i=1}^{n} \sigma(-\mathbf{w}^T \mathbf{x}^{(i)}) y^{(i)} x_j^{(i)}$$

How to classify new data?

- Once the final hypothesis $h_{\mathbf{w}}(\mathbf{x})$ is known …
  - ✓ output the label of the most probable class
- Assign label $+1$ to input instance $\mathbf{x}$, if $p(+1 \mid \mathbf{x}) > 0.5$ and label -1 otherwise

Final remarks

- Simple classifier with probabilistic outputs
- Loss function is convex and can be trained with GD methods (no closed-form)
- Robust to overfitting
- Offers interpretability to weights (feature importance)
- However, decision boundary is still linear