Machine Learning Lecture 6
Logistic Regression (Linear Classifiers)

Dr. Ioannis Patras

Slides thanks: Tim Hospedales

Course Context

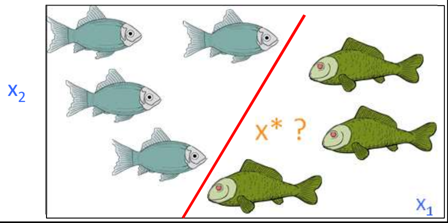
- Supervised Learning
  - (Linear) regression
  - (Linear) Classifiers and Logistic Regression
  - Neural Networks
- Unsupervised
  - Clustering
  - Density Estimation
  - HMMs

Supervised Learning

- Applications where the training data comprises examples of input vectors along with corresponding target vectors
- Regression: desired output consists of one or more continuous variables
- Classification: Desired output consists of a finite number of discrete categories.

(Linear) Classifier

- Goal:
  - Take an input vector **x** and assign it to one of K discrete classes y.
- Assume an partition of the feature space: y=f(x)
- Given examples (x_i,y_i), which may be noisy
- Learn f(x), to enable prediction of y* given new point x*. It should generalise well to new x*
  - E.g., x_1: Fish Weight, x_2=Fish Length, y=Fish Species.



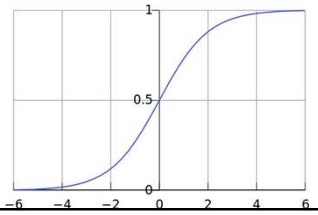
Overview

- Binary Classification – Logistic Regression
- Multiclass Classification – Max Entropy
- Extensions

Linear Versus Logistic Regression

- Linear & Logistic Regression use different representation/model assumptions.
- Linear Regression $f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} \in [-\infty, \infty]$
- Logistic Regression $f_{\mathbf{w}}(\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{w}^T\mathbf{x}})} \in [0,1]$

$\sigma(\mathbf{x})$



$\mathbf{x}$

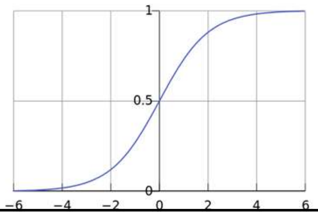
Sigmoid aka Logistic Function $\sigma(\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{x}})}$

Linear Versus Logistic Regression

- Think about this Logistic function.
  - What if x -> -/+ infinity? (Assume w=1)
    - x controls range of σ(x) in [0,1]
  - What if x=2 and we increase/decrease w magnitude?
    - w Controls slope of $f_w(x)$

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

$f_w(\mathbf{x})$



$\mathbf{x}$

Sigmoid aka Logistic Function $\sigma(\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{x}})}$

Linear Versus Logistic Regression

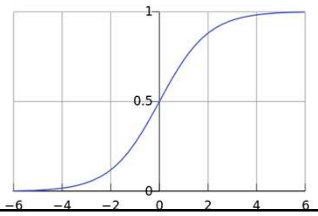
- Linear & Logistic Regression use different representation/model assumptions.
- Linear Regression $f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$

Range: +inf to - inf

- Logistic Regression $f_{\mathbf{w}}(\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{w}^T\mathbf{x}})}$

Range: 0 to 1

$f_w(\mathbf{x})$



$\mathbf{x}$

Sigmoid aka Logistic Function $\sigma(\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{x}})}$

Logistic Regression: What does it mean for modeling classes?

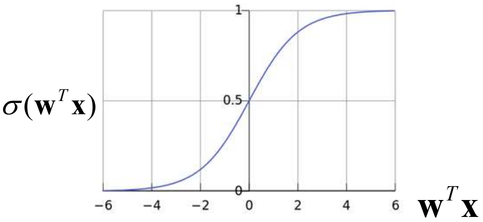
- With assumption

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

- Probability Model:

$$p(y \mid \mathbf{x}, \mathbf{w}) = Bernoulli(y \mid \sigma(\mathbf{w}^T\mathbf{x}))$$

  – => Probability of class y is $\sigma(\mathbf{w}^T\mathbf{x})$



Logistic Regression

$$p(y \mid \mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{w}^T\mathbf{x}})}$$

Range: 0 to 1

P(Class 1|**x**) $= \sigma(\mathbf{w}^T\mathbf{x})$

P(Class 0|**x**) $= 1 - \sigma(\mathbf{w}^T\mathbf{x})$

$\Rightarrow$

$\sigma(\mathbf{w}^T\mathbf{x}) > 0.5$: Class 1

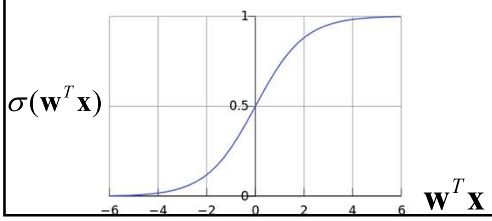
$\sigma(\mathbf{w}^T\mathbf{x}) < 0.5$: Class 0

$\Rightarrow$

$\mathbf{w}^T\mathbf{x}>0$

$\mathbf{w}^T\mathbf{x}<0$

$\mathbf{w}^T\mathbf{x}=0$



What Kind of Decision Boundary Does LR Have?

- Boundary between two classes. Contour where:
  - p(Class 1|**x**)=p(Class 0|**x**)=0.5.

$$1/(1+e^{-\mathbf{w}^T\mathbf{x}}) = 0.5$$

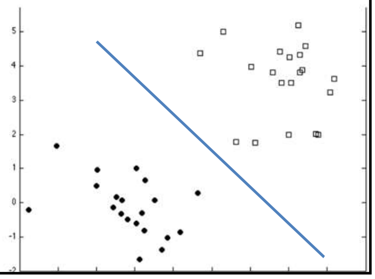
$$2 = 1+e^{-\mathbf{w}^T\mathbf{x}}$$

$$0 = \mathbf{w}^T\mathbf{x}$$

=> It's a straight line!

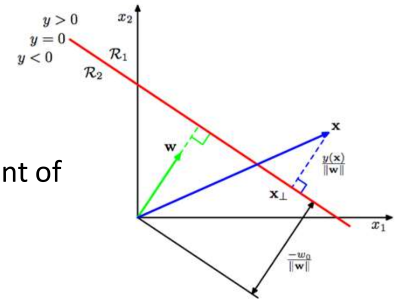
$$f_{\mathbf{w}}(\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{w}^T\mathbf{x}})}$$

Parameters **w** specify exactly which straight line we try to separate the data with

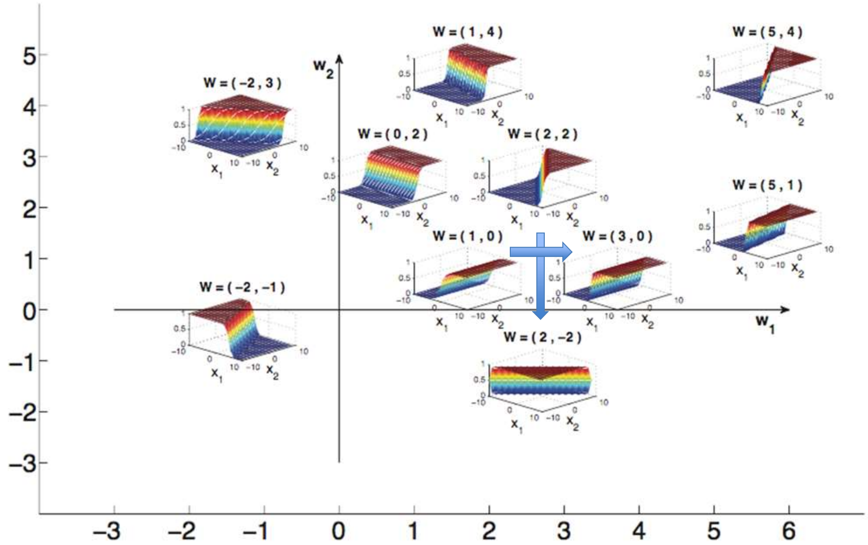


Unpacking the decision boundary.

- Really we should be doing: $\sigma(\mathbf{w}^T\mathbf{x}+w_0) = \frac{1}{(1+e^{-\mathbf{w}^T\mathbf{x}+w_0})}$
  - Recall in linear regression, we used the trick **x'**:=[1,**x**]
  - Now **w'** automatically includes a [w_0,**w**].
  - So $\mathbf{w}'^T\mathbf{x}'=\mathbf{w}^T\mathbf{x}+\mathbf{w}_0$
- The decision boundary (red) is perpendicular to the vector **w**.
  - Weight w_0 controls the displacement of the line from origin.



Unpacking the prediction surface.



Choosing the objective function

- For Linear Regression, we had:

$$f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} \quad \Rightarrow \quad E(\mathbf{w}) = \sum_i \left(y_i - \mathbf{w}^T\mathbf{x}_i\right)^2$$

- For Logistic Regression?

$$f_{\mathbf{w}}(\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{w}^T\mathbf{x}})} \quad \Rightarrow \quad ??$$

Choosing the objective function

- For Linear Regression, we had:

Square deviation of prediction from label

$$f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} \quad \Rightarrow \quad E(\mathbf{w}) = \sum_i \left(y_i - \mathbf{w}^T\mathbf{x}_i\right)^2$$

- For Logistic Regression?

Recall y_i: {0,1}

$$p(Y \mid X, \mathbf{w}) = \prod_i p(y=1 \mid \mathbf{x}_i)^{y_i}(1 - p(y=1 \mid \mathbf{x}_i))^{(1-y_i)}$$

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

Probability model **w** would have got each point correct

Cost: Negative Log Likelihood

$$E(\mathbf{w}) = -\sum_i y_i \log p(y=1 \mid \mathbf{x}_i) + (1-y_i)\log(1-p(y=1 \mid \mathbf{x}_i))$$

Learning Logistic Regression

- Find the weights **w** that minimize the cost function for data {**y**,X } or {(y_i,$\mathbf{x}_i$)}:

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

$$E(\mathbf{w}) = -\sum_i y_i \log p(y=1 \mid \mathbf{x}_i) + (1-y_i)\log(1-p(y=1 \mid \mathbf{x}_i))$$

  – Derivates wrt weights…

$$\frac{dE(\mathbf{w})}{d\mathbf{w}} = -\sum_i (\sigma(\mathbf{w}^T\mathbf{x}_i) - y_i)\mathbf{x}_i$$

$$\frac{dE(\mathbf{w})}{d\mathbf{w}} = -X^T(p(\mathbf{y} \mid X) - \mathbf{y})$$

Aside: Convex vs Closed Form

- Find the weights **w** to minimize the cost function for data {**y**,X} or {y_i,$\mathbf{x}_i$}:

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

$$E(\mathbf{w}) = -\sum_i y_i \log p(y=1|\mathbf{x}_i) + (1-y_i)\log(1-p(y=1|\mathbf{x}_i))$$

  – Derivates wrt weights…

$$\frac{dE(\mathbf{w})}{d\mathbf{w}} = -\sum_i (\sigma(\mathbf{w}^T\mathbf{x}_i) - y_i)\mathbf{x}_i$$

Note: This time (unlike linear regression), we can't re-arrange to solve for w! It's stuck inside the sigmoid.

No closed form solution. Only gradient.
However it is convex.
Remember what convex means?

Gradient will get the solution as unique minimum.

Learning Logistic Regression

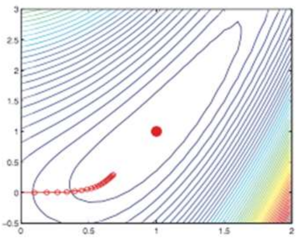
- Find the weights **w** that minimize the cost function for data {**y**,X} or {y_i,$\mathbf{x}_i$}:

$$E(\mathbf{w}) = -\sum_i y_i \log p(y=1|\mathbf{x}_i) + (1-y_i)\log(1-p(y=1|\mathbf{x}_i))$$

- Algorithm:
  – Repeat:

$$\mathbf{w}' := \mathbf{w} + \alpha\frac{E(\mathbf{w})}{d\mathbf{w}} \qquad \mathbf{w}' := \mathbf{w} - \alpha\sum_i (\sigma(\mathbf{w}^T\mathbf{x}_i) - y_i)\mathbf{x}_i$$

  Until convergence



Checkpoint

- Good practice:
  - Check you understand the dimensions of any algorithm/linear algebra expression.
- Questions:
  - If input data is d dimensions, how many dimensions is **w**? **x**?
  - How many dimensions is σ(**w**^{T}**x**)
  - How many dimensions is E(**w**)?
  - How many dimensions is dE(**w**)/d**w**?

$$p(y=1\,|\,\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{w}^T\mathbf{x}})}$$

$$E(\mathbf{w}) = -\sum_i y_i \log p(y=1\,|\,\mathbf{x}_i) + (1-y_i)\log(1-p(y=1\,|\,\mathbf{x}_i))$$

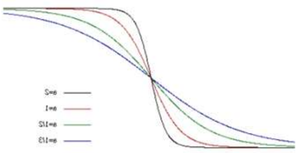
$$\frac{dE(\mathbf{w})}{d\mathbf{w}} = -\sum_i (\sigma(\mathbf{w}^T\mathbf{x}_i) - y_i)\mathbf{x}_i$$

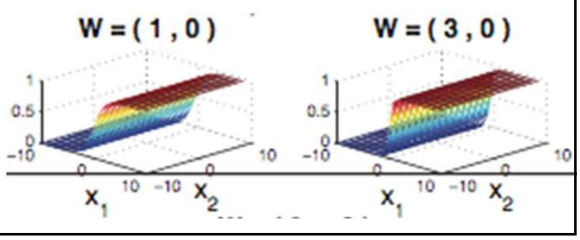
What about Confidence?

- Larger magnitude weights result to steeper sigmoid.

$$p(Y\,|\,X,\mathbf{w}) = \prod_i p(y=1\,|\,\mathbf{x}_i)^{y_i}(1-p(y=1\,|\,\mathbf{x}_i))^{(1-y_i)}$$

$$p(y=1\,|\,\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{w}^T\mathbf{x}})}$$





What about Confidence?

- Larger magnitude weights result to steeper sigmoid.

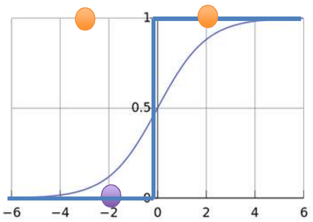
$$p(Y \mid X, \mathbf{w}) = \prod_i p(y=1 \mid \mathbf{x}_i)^{y_i} (1 - p(y=1 \mid \mathbf{x}_i))^{(1-y_i)}$$

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

  – If w=1. σ($\mathbf{w}^T\mathbf{x}_i$)=[0.05,0.9,0.9]
    - Probability of all data = 0.05*0.9*0.9 = 0.04

Learning won't increase weight

  – If w=inf. σ($\mathbf{w}^T\mathbf{x}_i$)=[0,1,1]



Probability of all data =0*1*1 =0.

Larger weights => more confident prediction.

What about Confidence?

- Larger magnitude weights result to steeper sigmoid.

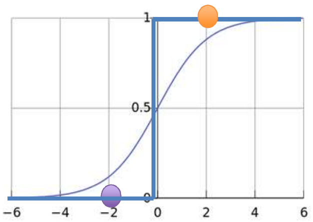
$$p(Y \mid X, \mathbf{w}) = \prod_i p(y=1 \mid \mathbf{x}_i)^{y_i} (1 - p(y=1 \mid \mathbf{x}_i))^{(1-y_i)}$$

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

  – If w=1. σ($\mathbf{w}^T\mathbf{x}_i$)=[0.9,0.9]
    - Probability of all data = 0.9*0.9 = 0.8

Learning will increase weight forever

  – If w=inf. σ($\mathbf{w}^T\mathbf{x}_i$)=[1,1]



Probability of all data =1*1 =1.

Larger weights => more confident prediction.

Regularized Logistic Regression

- To address overconfidence, we can add an L2 norm regularizer.

$$E(\mathbf{w}) = -\sum_i y_i \log p(y_i = 1 | \mathbf{x}_i) + (1 - y_i) \log(1 - p(y_i = 1 | \mathbf{x}_i)) + \lambda \mathbf{w}^T \mathbf{w}$$

Aside: What is "l2 norm" about?

- Roughly: A norm gives a scalar magnitude from a vector.
  - What are ways you can imagine to do this?

$$\|\mathbf{w}\|_2 = \sqrt{\sum_k w^2{}_k}$$

L2 norm
"Euclidean"

$$\|\mathbf{w}\|_1 = \sum_k |w_k|$$

L1 norm
"Manhattan"

$$\|\mathbf{w}\|_0 = \sum_k w_k \neq 0$$

L0 norm

Regularized Logistic Regression

- To address overconfidence, we can add a l2 regularizer.

$$E(\mathbf{w}) = -\sum_i y_i \log p(y_i = 1 | \mathbf{x}_i) + (1 - y_i) \log(1 - p(y_i = 1 | \mathbf{x}_i)) + \lambda \mathbf{w}^T \mathbf{w}$$

- Cost is scalar.
  – Need a scalar to quantify how costly a weight vector is.
  – Need a norm of the weight vector. $\lambda \sum_k w_k{}^2 = \lambda \mathbf{w}^T \mathbf{w}$

Regularized Logistic Regression

- To address overconfidence, we can add a l2 regularizer.
  – As before we saw this corresponds to zero mean Gaussian weight prior!

$$p(w | Y, X) \propto p(Y | X, \mathbf{w}) p(\mathbf{w})$$

$$\propto \prod_i p(y = 1 | \mathbf{x}_i)^{y_i} (1 - p(y = 1 | \mathbf{x}_i))^{(1 - y_i)} \exp(-\lambda \mathbf{w}^T \mathbf{w})$$

$$E(\mathbf{w}) = -\sum_i y_i \log p(y_i = 1 | \mathbf{x}_i) + (1 - y_i) \log(1 - p(y_i = 1 | \mathbf{x}_i)) + \lambda \mathbf{w}^T \mathbf{w}$$

Learning Regularized Logistic Regression

- Take derivatives of…
  - The Posterior rather than the Likelihood
  - =The augmented cost function

$$E(\mathbf{w}) = -\sum_i y_i \log p(y_i = 1 \mid \mathbf{x}_i) + (1 - y_i)\log(1 - p(y_i = 1 \mid \mathbf{x}_i)) + \lambda \mathbf{w}^T \mathbf{w}$$

$$\mathbf{w}' := \mathbf{w} + \alpha \frac{E(\mathbf{w})}{d\mathbf{w}} \qquad \mathbf{w}' := \mathbf{w} - \alpha \sum_i (\sigma(\mathbf{w}^T \mathbf{x}_i) - y_i)\mathbf{x}_i - \lambda \mathbf{w}$$

Summary

- Logistic Regression
  - Uses sigmoid function to predict probability in [0,1]
  - Weights specify the decision boundary.
  - Weight magnitudes specify its sharpness
- Cost as negative logprobability of correct assignment.
  - Using binary exponent trick.
- Differentiation wrt weights leads to convex gradient-based solution.
  - (But not closed form)
- Regularize to avoid overconfidence.

Linear Classifier Examples – 1
EECS Work ☺

- Undergraduate Project
  - **x**: Text of bills under debate in parliament
  - $y=f(\mathbf{x})=\mathbf{w}^T\mathbf{x}$: Bill is accepted into law or not after parliament debate
- Simple linear classifier can predict with ~85% accuracy!

Linear Classifier Examples – 2
EECS Work ☺

- Person re-identification
  - Take two images (windows) x_1, x_2
  - $F(x_1,x_2) = \mathbf{w}^T|x_1-x_2|$ if it's the same person or not.

Case Study – Feature Engineering
EECS Work @ ACCV 2014 ☺

- Forensic Sketch / Caricature
  - Take an image $\mathbf{x}$
  - $A_i(\mathbf{x}) = \mathbf{w}_i^T\mathbf{x}$ says if the face has an attribute i or not
    - E.g., big/small nose, fat/thin lips, long/short hair, male/female
    - Each $A_i(x)$ is a new feature of original image
  - $F(\mathbf{x}_1,\mathbf{x}_2)=\mathbf{w}^T|A(\mathbf{x}_1)-A(\mathbf{x}_2)|$ says if the same person or not.
- => The feature itself is another machine learning model

Forensic sketch

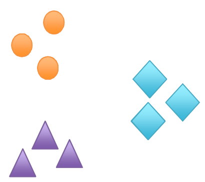
Caricature

Overview

- Binary Classification – Logistic Regression
- Multiclass Classification – Max Entropy
- Extensions

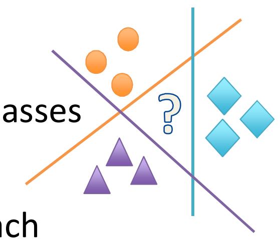
From Binary to Multiclass

- We know how to model (logistic function) and learn (gradient descent) binary classifiers.
  - What about K>2, multi-class problems?
  - Any ideas?



Multiclass: 1 vs All

- Simple solution: 1-versus-All
  - Use K classifiers, each solving a two class problem of separating class k from all others.
- Issues:
  - Ambiguous regions
  - Gets expensive with many classes
- Next:
  - A 'natural' multiclass approach

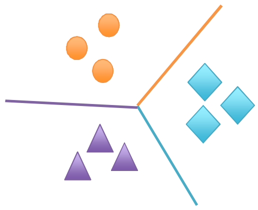


Multiclass: Multinomial logistic regression, MaxEnt, softmax.

- Before we had $p(y=1\,|\,\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{w}^T\mathbf{x}})}$
  - … a Bernoulli likelihood. $= Bernoulli(y\,|\,\sigma(\mathbf{w}^T\mathbf{x}))$
- What was the “dice” rather than “coin” distribution?
  - Multinomial. A parameter vector that adds up to 1.
  - Now y=1…K, rather than y=0,1.

$$p(y\,|\,\mathbf{x},\mathbf{W}) = \frac{\exp(\mathbf{w}^T{}_y\mathbf{x})}{\sum_{y'}\exp(\mathbf{w}^T{}_{y'}\mathbf{x})}$$

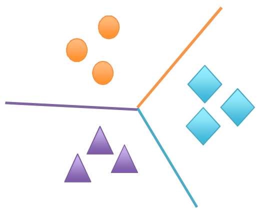
$$p(y\,|\,\mathbf{x},\mathbf{W}) = Multi(y\,|\,soft\max(\mathbf{w}_y^T\mathbf{x}))$$



Think about Softmax…

- What’s the range of exp($\mathbf{w}^T\mathbf{x}$) for x => -inf or +inf??
- What’s the range of: $\frac{\exp(\mathbf{w}^T{}_y\mathbf{x})}{\sum_{y'}\exp(\mathbf{w}^T{}_{y'}\mathbf{x})}$
- What if $\mathbf{x}$=k$\mathbf{w}_y$ for one y, and $\mathbf{x}$=-k$\mathbf{w}_{y'}$ for the others y’ != y?
  - And if k => inf?

$$p(y\,|\,\mathbf{x},\mathbf{W}) = \frac{\exp(\mathbf{w}^T{}_y\mathbf{x})}{\sum_{y'}\exp(\mathbf{w}^T{}_{y'}\mathbf{x})}$$



Relating Softmax to Logistic

- For Binary case K = 2.
  - What happens to Softmax?

$$p(y \mid \mathbf{x}, \mathbf{W}) = \frac{\exp(\mathbf{w}^T{}_y \mathbf{x})}{\sum_{y'} \exp(\mathbf{w}^T{}_{y'} \mathbf{x})}$$

$$p(y=1 \mid \mathbf{x}, \mathbf{W}) = \frac{\exp(\mathbf{w}_1^T \mathbf{x})}{\exp(\mathbf{w}_1^T \mathbf{x}) + \exp(\mathbf{w}_2^T \mathbf{x})} \qquad = \frac{1}{1 + \exp(\mathbf{w}_2^T \mathbf{x}) / \exp(\mathbf{w}_1^T \mathbf{x})}$$

$$= \frac{1}{1 + \exp(\mathbf{w}_2^T \mathbf{x} - \mathbf{w}_1^T \mathbf{x})} \qquad = \frac{1}{1 + \exp(-(\mathbf{w}_1 - \mathbf{w}_2)^T \mathbf{x})}$$

Same as Logistic From Before $\quad p(y=1 \mid \mathbf{x}) = \frac{1}{(1 + e^{-\mathbf{w}^T \mathbf{x}})}$

Evaluating a Multiclass Classifier

- Quantify The Goodness of a MaxEnt Classifier?
  - Again, probability that all data assigned correctly.

$$p(Y \mid X, W) = \prod_i^N \prod_c^K p(c \mid \mathbf{x}_i, W)^{Y_{ic}}$$

  - Where we let Y_{ic} be a sparse binary 1-of-N matrix encoding the labels. I.e., If y_i=c, then Y(i,c)=1.
  - Dimensions of Y and W?
    - Y is a N x K matrix of labels.
    - Now W is a KxD matrix stacking D dims of weights for each of K classifiers w_y.

0	0	1	0	0	0	…	0
0	0	0	1	0	0	…	0
0	0	1	0	0	0	…	0

Evaluating a Multiclass Classifier

- Quantify The Goodness of a MaxEnt Classifier?
  - Again, probability that all data assigned correctly.

$$p(Y|X,W) = \prod_i^N \prod_c p(c|\mathbf{x}_i, \mathbf{w}_y)^{Y_{ic}}$$

  - To specify a cost, take logs as usual.

$$E(\mathbf{W}) = \sum_i^N \sum_c Y_{ic} \log p(c|\mathbf{x}_i, W) \qquad p(c|\mathbf{x},\mathbf{W}) = \frac{\exp(\mathbf{w}^T{}_c\mathbf{x})}{\sum_{c'} \exp(\mathbf{w}^T{}_{c'}\mathbf{x})}$$

$$E(\mathbf{W}) = \sum_i^N \left[\left(\sum_c Y_{ic} \mathbf{w}_c{}^T \mathbf{x}_i \right) - \log \left(\sum_{c'} \exp(\mathbf{w}_{c'}{}^T \mathbf{x}_i) \right) \right]$$

Learning MaxEnt

- Differentiate the cost wrt W.

$$E(\mathbf{W}) = \sum_i^N \sum_c Y_{ic} \log p(c|\mathbf{x}_i, W) \qquad p(c|\mathbf{x},\mathbf{W}) = \frac{\exp(\mathbf{w}^T{}_c\mathbf{x})}{\sum_{c'} \exp(\mathbf{w}^T{}_{c'}\mathbf{x})}$$

$$E(\mathbf{W}) = \sum_i^N \left[\left(\sum_c Y_{ic} \mathbf{w}_c{}^T \mathbf{x}_i \right) - \log \left(\sum_{c'} \exp(\mathbf{w}_{c'}{}^T \mathbf{x}_i) \right) \right]$$

$$\frac{dE(\mathbf{W})}{d\mathbf{w}_c} = \sum_i^N (p(c|\mathbf{x}, W) - Y_{ic})\mathbf{x}_i$$

Intuition:
If x_i supposed to be class c:
- Push w_c towards/away from x as appropriate

(Differentiating the matrix one vector at a time)

Summary

- Multi-class classification:
  - Simple indirect solution 1-vs-All of Logistic regressions
  - Direct solution: Use softmax.
- Cost function: Probability of correct assignment.
  - Use binary exponentiation trick again to express succinctly.
- Can learn with gradient as before.
- Same form as LR in the case where K=2.

Overview

- Binary Classification – Logistic Regression
- Multiclass Classification – Max Entropy
- Extensions
  - Non-linear
  - Sparse regularizers
  - Multi-label

Non-linear trick

- As with Linear Regression, we can use a fixed non-linear (e.g., polynomial) transformation.
  – Result: Non-linear in feature-space. Linear wrt w

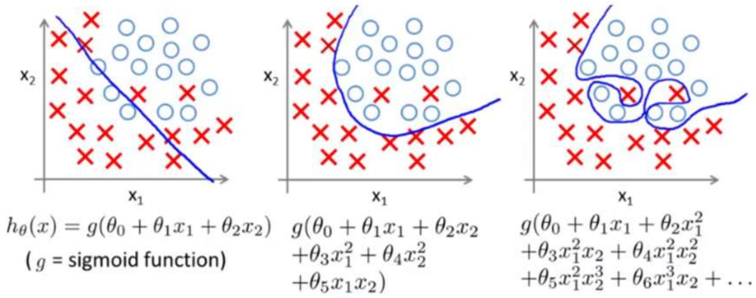
$$p(y=1|\mathbf{x}) = \sigma(\mathbf{w}^T\phi(\mathbf{x})) = \frac{1}{(1+e^{-\mathbf{w}^T\phi(\mathbf{x})})}$$

$$\phi(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_2^2, x_1x_2]^T$$

$$p(y=1|\mathbf{x}) = \sigma(w_0 + w_1x_1 + w_2x_1{}^2 + w_3x_2 + w_5x^2{}_2 + w_5x_1x_2)$$

Non-linear trick

- Using non-linear logistic regression this way, can result to over/underfitting again.

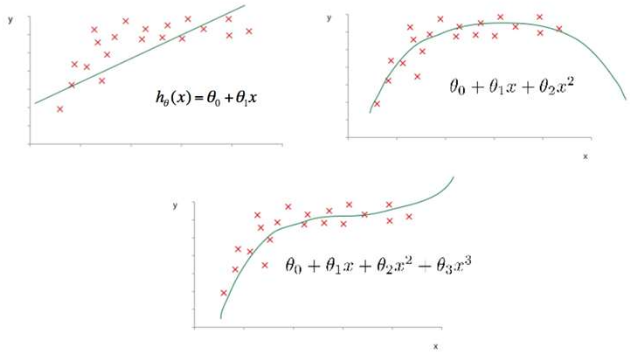


Underfitting

Overfitting

Non-linear trick

- Contrast non-linear we saw in regression….



Overview

- Binary Classification – Logistic Regression
- Multiclass Classification – Max Entropy
- Extensions
  - Non-linear
  - Sparse regularizers
  - Multi-label

Regularization for Sparsity

- Common to use l2 regularization

$$E(\mathbf{w}) = -\sum_i y_i \log p(y_i = 1 | \mathbf{x}_i) + (1 - y_i)\log(1 - p(y_i = 1 | \mathbf{x}_i)) + \lambda \mathbf{w}^T \mathbf{w}$$

- …as for linear regression, l1 is also possible

$$E(\mathbf{w}) = -\sum_i y_i \log p(y_i = 1 | \mathbf{x}_i) + (1 - y_i)\log(1 - p(y_i = 1 | \mathbf{x}_i)) + \lambda \sum_d |w_d|$$

- This increases sparsity…

Regularization for Sparsity

- Common to use l2 regularization

$$E(\mathbf{w}) = -\sum_i y_i \log p(y_i = 1 | \mathbf{x}_i) + (1 - y_i)\log(1 - p(y_i = 1 | \mathbf{x}_i)) + \lambda \mathbf{w}^T \mathbf{w}$$

- Sometimes we want sparsity.
  - Delete irrelevant distractor features.
  - Save memory.
  - Gain domain insight.
- How? Remember those regularizers…
  - L0 is explicit about sparsity. Cost=# non-zero elements.

Can't differentiate wrt w

$$\|\mathbf{w}\|_2 = \sqrt{\sum_2 w^2{}_k} \qquad \|\mathbf{w}\|_1 = \sum_k |w_k| \qquad \|\mathbf{w}\|_0 = \sum_k w_k \neq 0$$

L1 Regularization

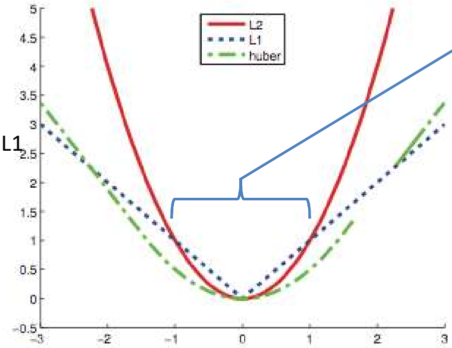
- L1 regularization increases sparsity…. Why?

This region:
L1 cost gradient remains constant approaching zero
L2 cost gradient approaches zero as w approaches zero

L2

Cost

L1

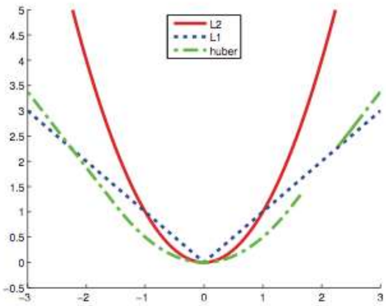


Weight Value

L1 Regularization

- But L1 regularisation still harder to Optimize. Why?
  - Absolute value has no derivative at zero (non-smooth corner)

$$\|\mathbf{w}\|_1 = \sum_k |w_k|$$



Overview

- Binary Classification – Logistic Regression
- Multiclass Classification – Max Entropy
- Extensions
  - Non-linear
  - Sparse regularizers
  - Multi-label
  - Large Scale
  - Interpretation

Multi-Label Classification

- Logistic Regression assumed data like $\{\mathbf{x}_i, y_i\}$.
  - And trained predictor: $$p(y=1\,|\,\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{(1+e^{-\mathbf{w}^T\mathbf{x}})}$$
- What about if you have $\{\mathbf{x}_i, \mathbf{y}_i\}$? (Vector of labels)
  - Classification analogy of multiple output linear regression.
  - When?
    - E.g., Webpage topic classifier for indexing. One web page can have multiple topics: Education+Government, Business+Finance, Business+Movie Industry..
    - Person can have multiple nationalities
    - Movie/music can span multiple genres.

Multilabel vs Multiclass

- Multiclass
  - Predict p(y|**x**).
  - y=1…K. Exclusive.
  - X has to have exactly one label.
- Multi-label
  - Predict p(**y**|**x**)
  - **y** is binary vector of size K.
  - If y_k=1, then **x** has label k.
  - If y_k=1 for all k. Then **x** has all labels.
  - If y_k=0 for all k. Then **x** has no labels.

Multilabel vs Multiclass

Multiclass

- One x has exactly one label
- Train data is scalar y for one **x**
- Prediction y=f(**x**) is a scalar
- y=1….K exclusive
- P(y|**x**) is a K-sized vector

Multilabel

- One x has many labels (0 to K)
- Train data is vector y for one x.
- Prediction **y**=f(**x**) is a K-vector
- y_k={0,1}
- P(**y**|**x**) is a K-sized vector.

How to do Multi-label Classification?

- Given data $\{\mathbf{y}_i,\mathbf{x}_i\}$.
  - Train an independent logistic regression classifier for each $p_k(y_k|\mathbf{x})$.
  - Stack up the results $p(\mathbf{y}|\mathbf{x})=[p_1(y_1|\mathbf{x}).....p_K(y_K|\mathbf{x})]$.
- This does not take into account label dependencies.

Multi-label Example

Overview

- Binary Classification – Logistic Regression
- Multiclass Classification – Max Entropy
- Extensions
  - Non-linear
  - Sparse regularizers
  - Multi-label
  - Large Scale
  - Interpretation

Large Scale

- As for linear regression, the strategy is to minimise the cost function

$$E(\mathbf{w}) = -\sum_i y_i \log p(y_i = 1 \mid \mathbf{x}_i) + (1 - y_i)\log(1 - p(y_i = 1 \mid \mathbf{x}_i)) + \lambda \mathbf{w}^T \mathbf{w}$$

- By iterating:

$$\mathbf{w}^{s+1} := \mathbf{w}^s - \alpha \sum_i^N (\sigma(\mathbf{w}^T \mathbf{x}_i) - y_i)\mathbf{x}_i - \lambda \mathbf{w}$$

- …. Means that you have to read database off disk into memory very many times.
  - Notice inner loop over i=1…N
    - … within an iterative loop over s

Large Scale

- Regular:
  - Iterate S times:

$$\mathbf{w}^{s+1} := \mathbf{w}^{s} - \alpha \sum_{i}^{N} (\sigma(\mathbf{w}^{T}\mathbf{x}_i) - y_i)\mathbf{x}_i - \lambda \mathbf{w}$$

- SGD with mini-batches:
  - Iterate M times:
  - Select a random subset B of total N examples
    - Iterate S times:

$$\mathbf{w}^{s+1} := \mathbf{w}^{s} - \alpha \sum_{i}^{B} (\sigma(\mathbf{w}^{T}\mathbf{x}_i) - y_i)\mathbf{x}_i - \lambda \mathbf{w}$$

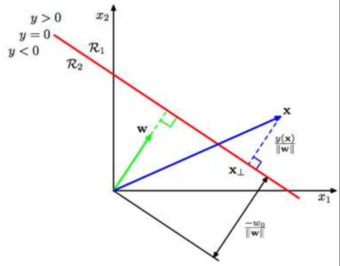
- Before: S memory reads of full database.
  - Now M*(B/N) reads. Can be << S.
  - E.g., M=N/B => Read once.

Overview

- Binary Classification – Logistic Regression
- Multiclass Classification – Max Entropy
- Extensions
  - Non-linear
  - Sparse regularizers
  - Multi-label
  - Large Scale
  - Interpretation

Interepreting Logistic Regression Outputs

- Once learned: $\mathbf{w} = \arg\min E(\mathbf{w})$
  - Recall each w_k is multiplied with $x_k$in $\sigma(\mathbf{w}^T\mathbf{x})$
  - => Positive w_k => Feature is indicator of class.
  - => Negative w_k => Contra-indicator.
  - If l1 regularizer killed w_k
    - Irrelevant to classification.



LR Interpretation Example

EECS Work ☺

theguardian

all opinion culture business lifestyle fashion environment tech travel ≡ browse all sections

Liberals swear more on Twitter than rightwingers, says study

Swearwords account for two of the top ten words used by left-leaning tweeters, while those on the right show a preference for "God" and "psalm"

An analysis of nearly a million tweets has found that liberals distinguish themselves from others online by letting rip with profanities.

While right-leaning tweeters displayed more caution in the language they used, the reds were more blue, as liberals peppered their 140-character missives with a hefty helping of "shits" and "fucks".

LR Interpretation Example

EECS Work ☺

Table 7. Logistic regression model with all predictors using data without outliers.

Predictors	Estimate	Standard Error	Z value	P value
(Intercept)	-0.32546	0.22806	-1.427	0.153555
1st person singular pronouns	0.08867	0.02054	4.316	1.59E-05***
1st person plural pronouns	-0.3435	0.09793	-3.507	0.000452***
Swear words	0.88577	0.21676	4.086	4.38E-05***
Positive emotion words	0.13363	0.02517	5.31	1.10E-07***
Negative emotion words	-0.12299	0.05362	-2.294	0.021806*
Anxiety words	0.85417	0.21939	3.893	9.88E-05***
Feeling words	-0.03407	0.14776	-0.231	0.817626
Tentative words	-0.11298	0.05405	-2.09	0.036593*
Certainty words	-0.03662	0.07287	-0.503	0.615252
Achievement words	-0.09091	0.05773	-1.575	0.115288
Religion words	-0.22651	0.1444	-1.569	0.116749
Death words	-0.28486	0.24512	-1.162	0.245182

Similarly, study from colleagues at MSR….


PNAS

Private traits and attribut
digital records of human b

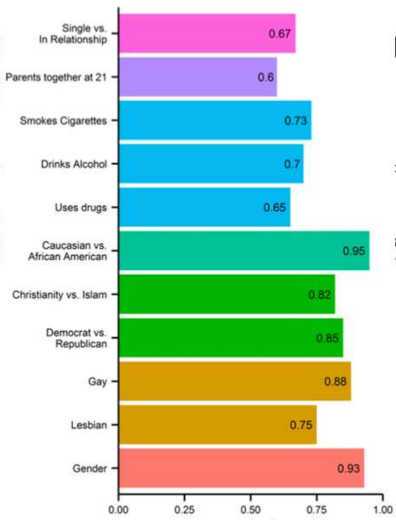
Michal Kosinski[a,1], David Stillwell[a], and Thore Graepel[b]

[a]Free School Lane, The Psychometrics Centre, University of Cambridge, Car
United Kingdom

Edited by Kenneth Wachter, University of California, Berkeley, CA, and ap

We show that easily accessible digital records of behavior, Faceboo

Facebook Likes (public by default)



Summary

- Non-linear extensions exist
  - Like linear regression, take fixed polynomial basis functions.
    - Then you would need to regularize for more than 'just' overconfidence.
- L2/L1 regularizer options also exist.
  - L2 is efficient.
  - L1 less efficient, but helps find sparsity.
- Multi-label versus Multi-class classification.
  - Multi-label: Simple solution by K-binary classifiers.