Lecture Notes for

Machine Learning in Python

⟦👨‍🏫, 👨‍💻, 🐍, 👨‍🔬⟧

Professor Eric Larson

Logistic Regression

Class Logistics and Agenda

- Logistics
  - A2: Images due soon!
  - Grading update
  - **Reminder**: Stay up to date with the quizzes!
- Agenda
  - Finish Image Town Hall
  - Logistic Regression
    - Solving and Programming

Class Overview, by topic

Table Data Visualization

Numpy, Pandas, Seaborn

Overviews with some in-depth discussion

Dimension Reduction and Image Processing

Scikit-learn, Scikit Image,

Intuition only, Some mathematics

Linear and Logistic Regression

Numpy, Recreate API for Scikit-learn

Detailed mathematics for simple optimization
intuition for advanced optimization

Neural Networks and Back Prop.

Numpy

Detailed mathematics for NN operations

Wide and Deep Networks | **Convolutional Networks** | **Recurrent Networks**

Keras, Tensorflow

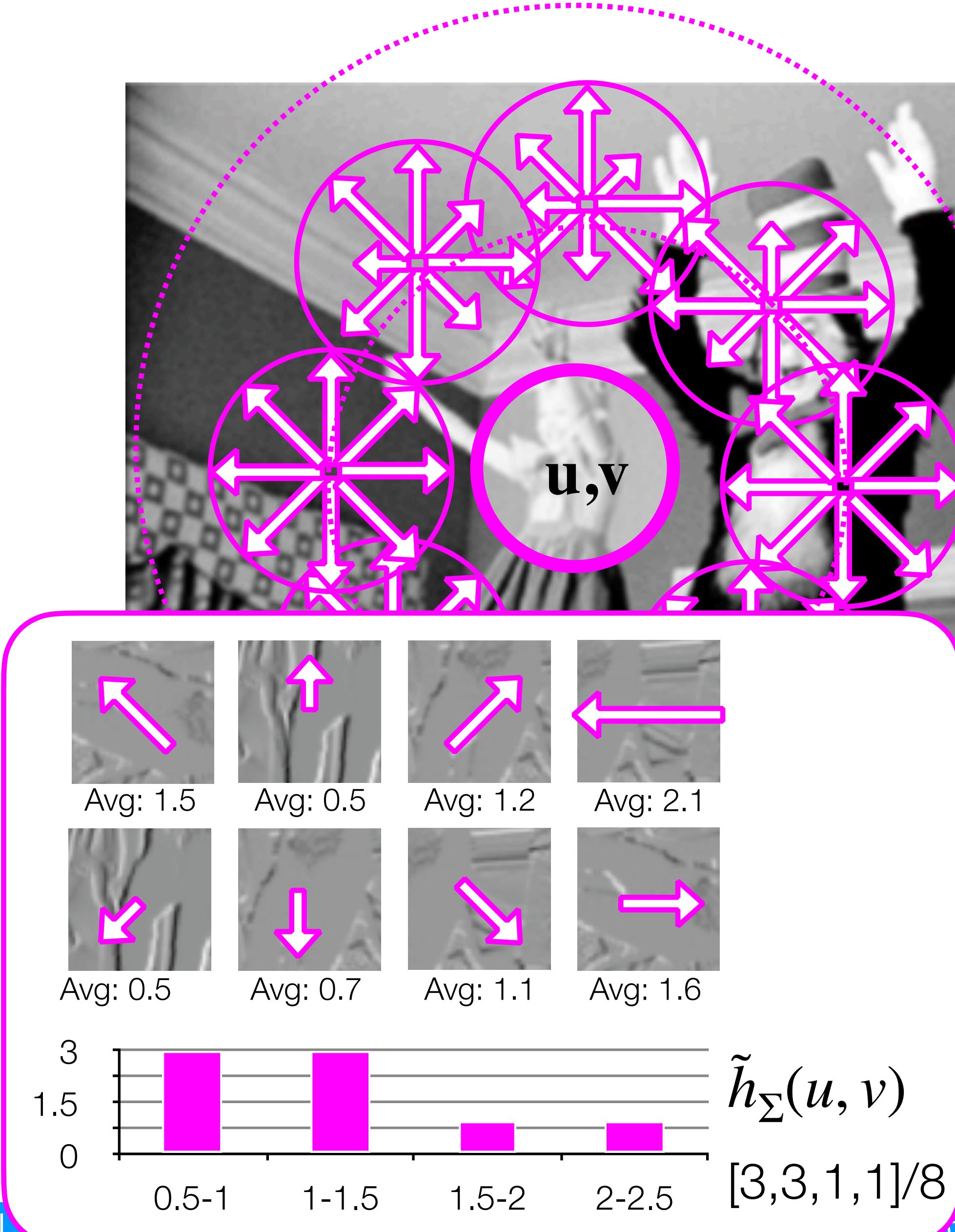
Intuition, Detailed implement.

Ethics in Language Models

ConceptNet

Case studies

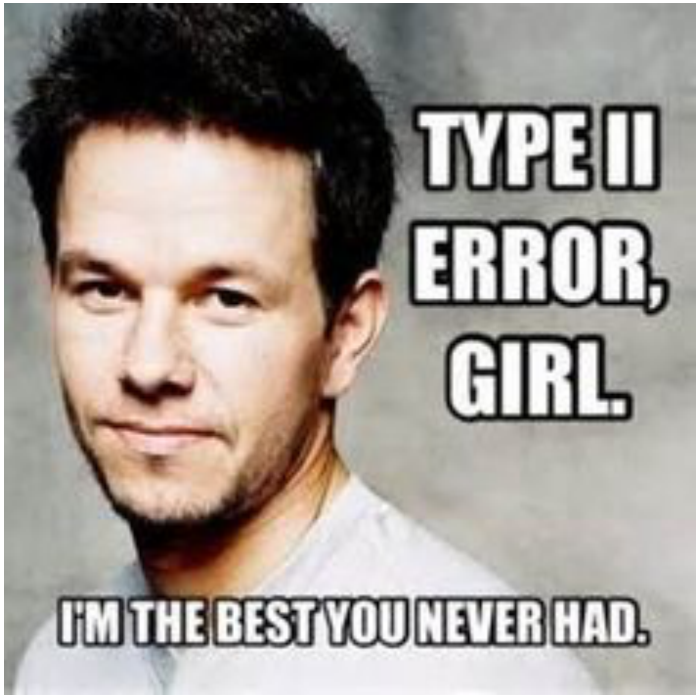
Lat Time: DAISY



1. Select u,v pixel location in image and radius
2. Take histogram of average gradient magnitudes in circle for each orientation $\tilde{h}_{\Sigma}(u, v)$
3. Select circles in a ring, R
4. For each circle on the ring, take another histogram $\tilde{h}_{\Sigma}(\mathbf{l}_{O}(u, v, R_1))$
5. Repeat for more rings
6. Save all histograms as “descriptors” $[\tilde{h}_{\Sigma}(\cdot), \tilde{h}_{\Sigma}(\mathbf{l}_1(\cdot, R_1)), \tilde{h}_{\Sigma}(\mathbf{l}_2(\cdot, R_1)) \dots]$
7. Can concatenate descriptors as “feature” vector at that pixel location

Town Hall for Lab 2, Images

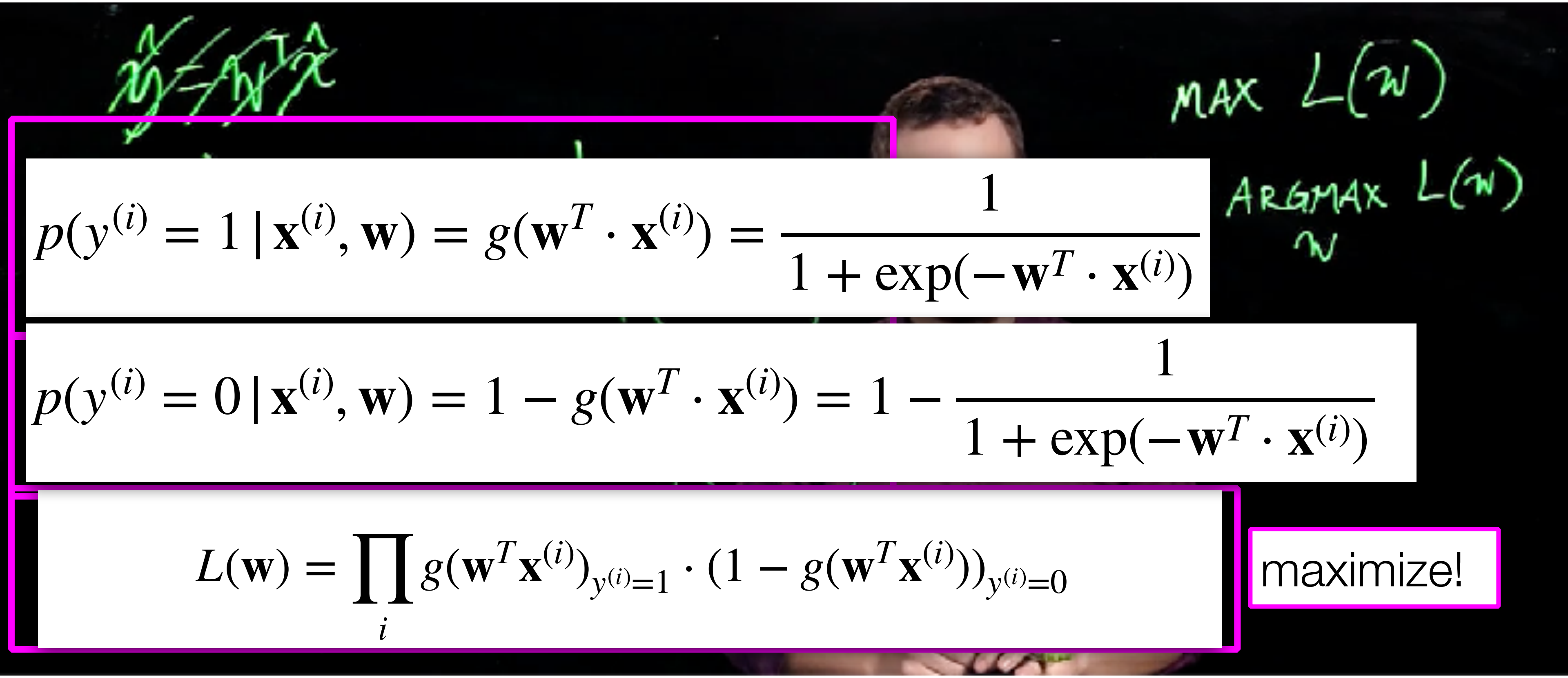
Logistic Regression



@researchmark

Setting Up Binary Logistic Regression

- From flipped lecture:



$$p(y^{(i)} = 1 \,|\, \mathbf{x}^{(i)}, \mathbf{w}) = g(\mathbf{w}^T \cdot \mathbf{x}^{(i)}) = \frac{1}{1 + \exp(-\mathbf{w}^T \cdot \mathbf{x}^{(i)})}$$

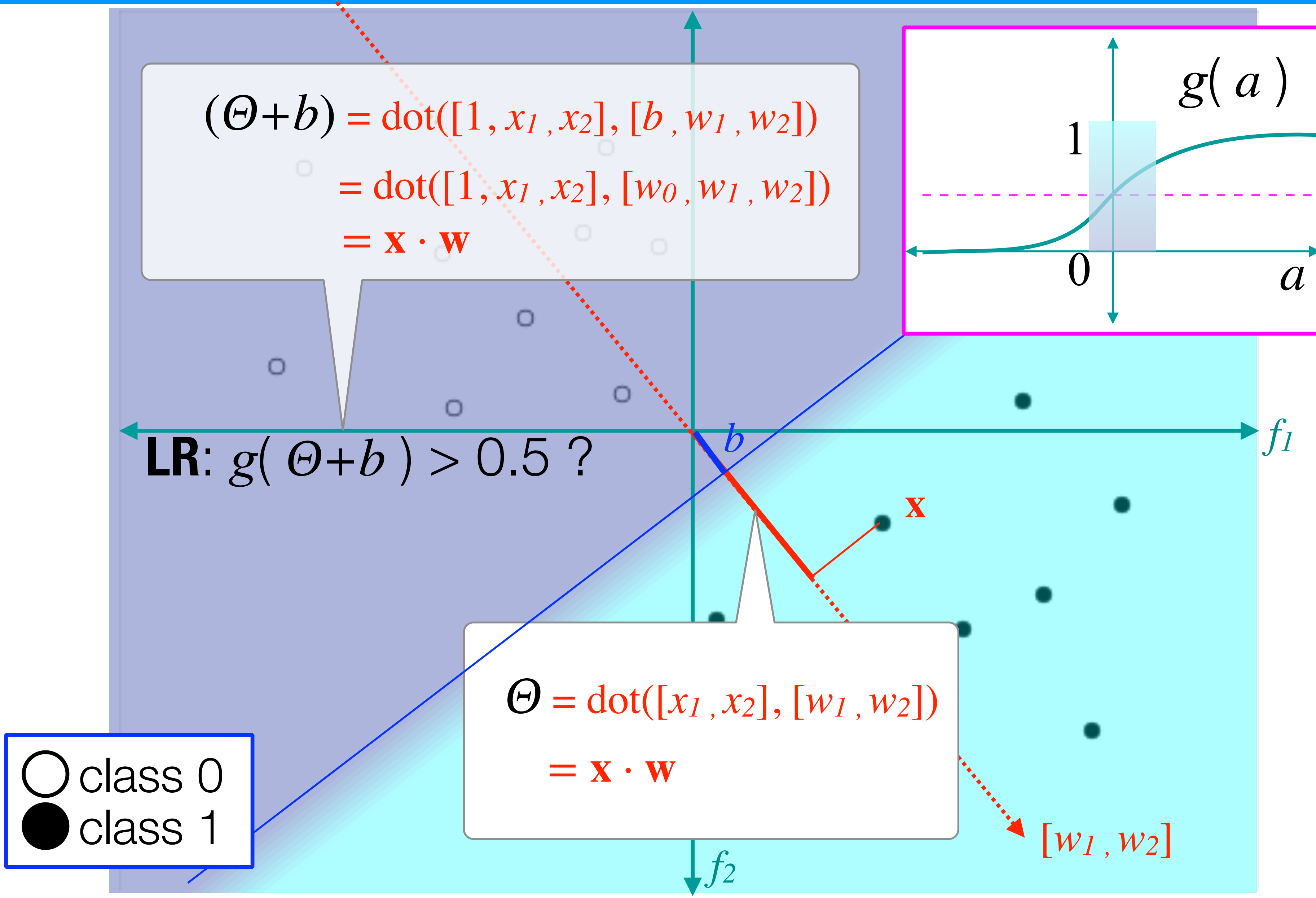
$$p(y^{(i)} = 0 \,|\, \mathbf{x}^{(i)}, \mathbf{w}) = 1 - g(\mathbf{w}^T \cdot \mathbf{x}^{(i)}) = 1 - \frac{1}{1 + \exp(-\mathbf{w}^T \cdot \mathbf{x}^{(i)})}$$

$$L(\mathbf{w}) = \prod_i g(\mathbf{w}^T \mathbf{x}^{(i)})_{y^{(i)}=1} \cdot (1 - g(\mathbf{w}^T \mathbf{x}^{(i)}))_{y^{(i)}=0}$$

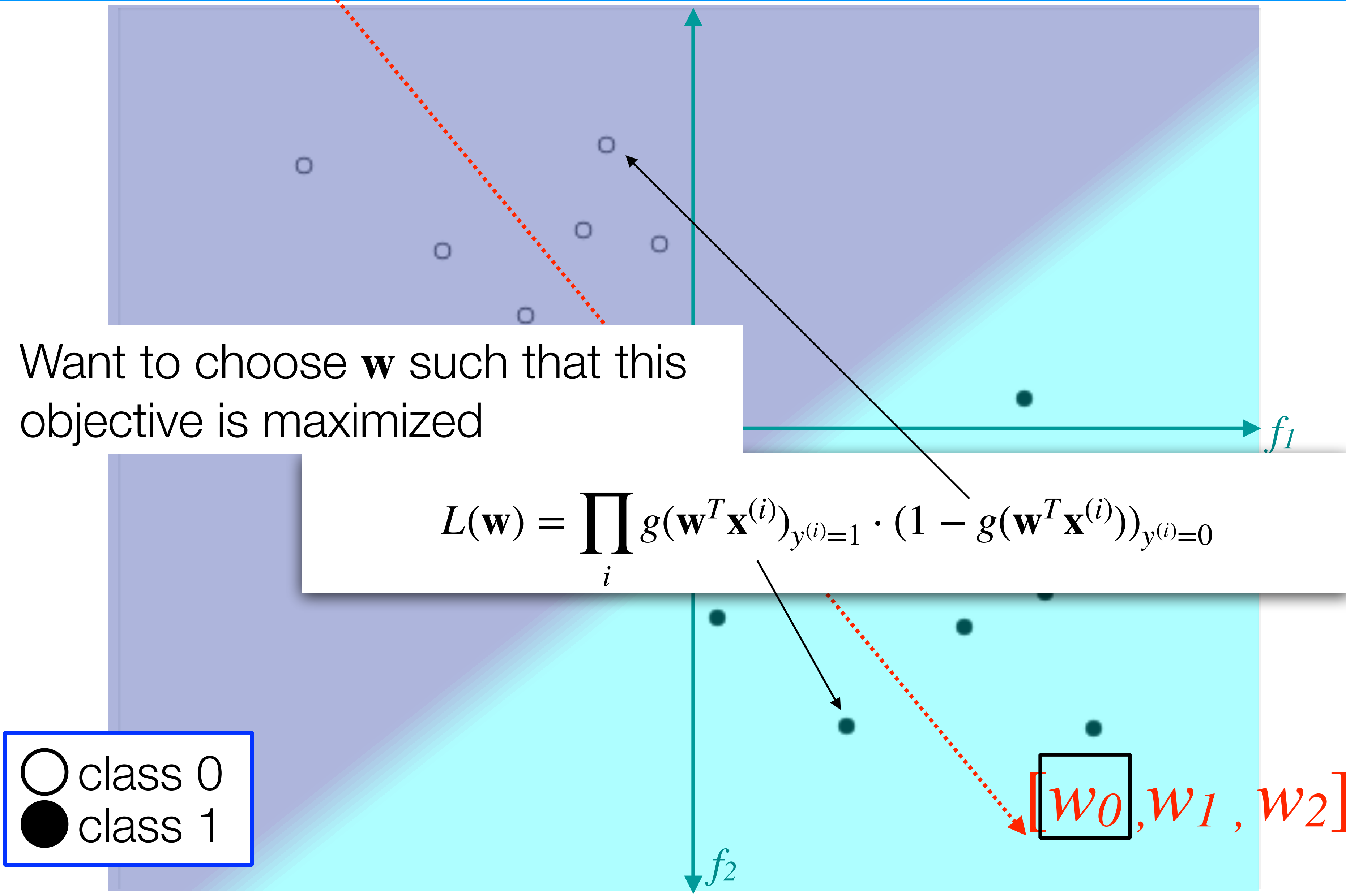
maximize!

where $g(.)$ is a sigmoid

What do weights and intercept define?



Changing $\mathbf{w}$ alters probability



Want to choose $\mathbf{w}$ such that this objective is maximized

$$L(\mathbf{w}) = \prod_i g(\mathbf{w}^T\mathbf{x}^{(i)})_{y^{(i)}=1} \cdot (1 - g(\mathbf{w}^T\mathbf{x}^{(i)}))_{y^{(i)}=0}$$

How do you optimize iteratively?

- **Objective Function**: the function we want to minimize or maximize
- **Parameters**: what are the parameters of the model that we can change?
- **Update Formula**: what update "step"can we take for these parameters to optimize the objective function?

$$L(\mathbf{w}) = \prod_i g(\mathbf{w}^T\mathbf{x}^{(i)})_{y^{(i)}=1} \cdot (1 - g(\mathbf{w}^T\mathbf{x}^{(i)}))_{y^{(i)}=0}$$

Logistic Regression Optimization Procedure

$$L(\mathbf{w}) = \prod_i g(\mathbf{w}^T\mathbf{x}^{(i)})_{y^{(i)}=1} \cdot (1 - g(\mathbf{w}^T\mathbf{x}^{(i)}))_{y^{(i)}=0}$$

- Simplify $L(\mathbf{w})$ with **logarithm**, $l(\mathbf{w})$ (aka: negative bce)

$$l(\mathbf{w}) = \sum_i y^{(i)} \ln\left(g(\mathbf{w}^T\mathbf{x}^{(i)})\right) + (1 - y^{(i)}) \ln\left(1 - g(\mathbf{w}^T\mathbf{x}^{(i)})\right)$$

- **Take** Gradient

$$\frac{\partial}{\partial w_j} l(\mathbf{w}) = \sum_i \left(y^{(i)} - g(\mathbf{w}^T\mathbf{x}^{(i)})\right) x_j^{(i)}$$

- **Use** gradient to **update** equation for $\mathbf{w}$

  - Video Supplement (also on canvas):
    - https://www.youtube.com/watch?v=FGnoHdjFrJ8

Binary Solution for Update Equation

- Use gradient inside update equation for **w**

$$\frac{\partial}{\partial w_j} l(\mathbf{w}) = \sum_i \left(y^{(i)} - g(\mathbf{w}^T \mathbf{x}^{(i)})\right) x_j^{(i)}$$

$$\underbrace{w_j}_{\text{new value}} \leftarrow \underbrace{w_j}_{\text{old value}} + \underbrace{\frac{\eta}{M}}_{\text{step}} \underbrace{\sum_{i=1}^{M} (y^{(i)} - g(\mathbf{w}^T \mathbf{x}^{(i)})) x_j^{(i)}}_{\text{gradient}}$$

This updates each element of gradient, how to calculate for all elements of the gradient update?

$$\mathbf{w} \leftarrow \mathbf{w} + \frac{\eta}{M} \sum_{i=1}^{M} (y^{(i)} - g(\mathbf{w}^T \mathbf{x}^{(i)})) \cdot \mathbf{x}^{(i)}$$

Demo

`05. Logistic Regression.ipynb`

Programming
Vectorization
Regularization
Multi-class extension

Other Tutorials:

http://blog.yhat.com/posts/logistic-regression-python-rodeo.html

http://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html

For Next Lecture

- **Next time**: More gradient based optimization techniques for logistic regression