Lecture Notes for

Machine Learning in Python

⟦👨‍🏫, 👨‍💻, 🐍, 👨‍🔬⟧

Professor Eric Larson

Logistic Regression

Class Logistics and Agenda

- Logistics
  - A2: Images due soon!
  - Grading discussion
  - **Reminder**: Stay up to date with the quizzes! (both for the canvas and flipped modules)
- Agenda
  - Finish Image Town Hall (if needed)
  - 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**

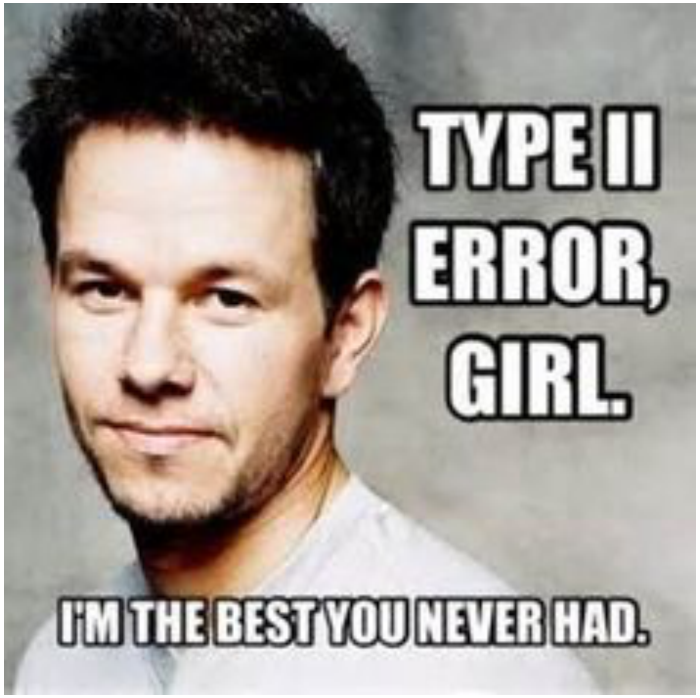
Keras, Tensorflow
Intuition, Detailed implement.

Ethics in Language Models

ConceptNet
Case studies

Last week: Town Hall for Lab 2, Images

Logistic Regression

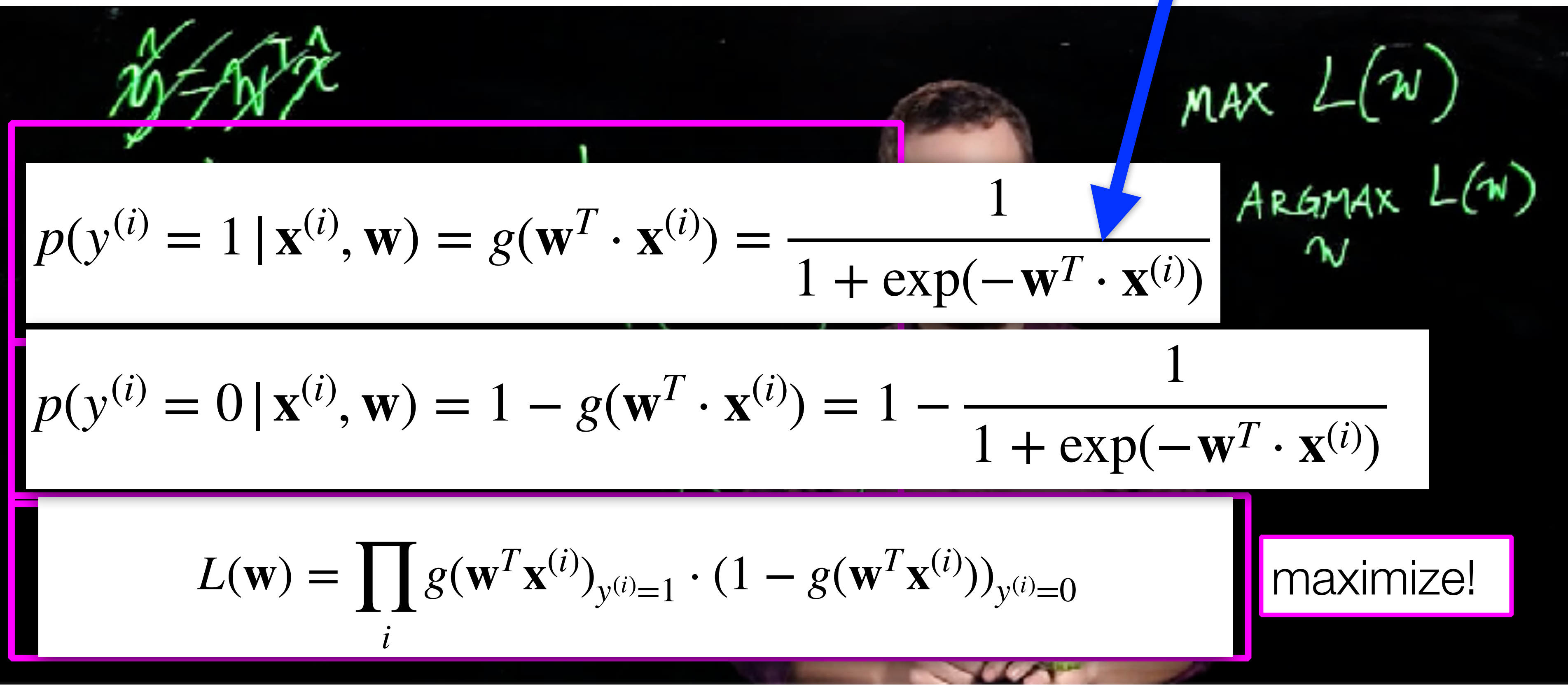


@researchmark

Setting Up Binary Logistic Regression

- From flipped lecture:

This notation assumes that $\mathbf{x}^{(i)}$ is a *column* vector, not *row*…



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

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

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

maximize!

where $g(.)$ is a sigmoid

Check on Understanding

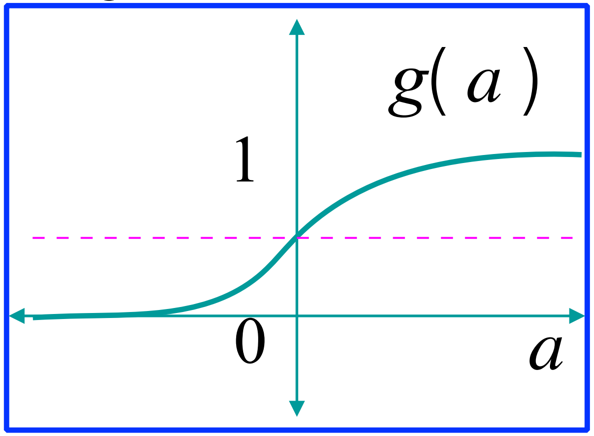
Updated Notation for Row Vector

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

$$\mathbf{X} = \begin{bmatrix} \leftarrow & \mathbf{x}^{(1)} & \rightarrow \\ \leftarrow & \mathbf{x}^{(2)} & \rightarrow \\ & \vdots & \\ \leftarrow & \mathbf{x}^{(M)} & \rightarrow \end{bmatrix} \qquad \mathbf{X} \cdot \mathbf{w} = \begin{bmatrix} \leftarrow & \mathbf{x}^{(1)} & \rightarrow \\ \leftarrow & \mathbf{x}^{(2)} & \rightarrow \\ & \vdots & \\ \leftarrow & \mathbf{x}^{(M)} & \rightarrow \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{(1)} \cdot \mathbf{w} \\ \mathbf{x}^{(2)} \cdot \mathbf{w} \\ \vdots \\ \mathbf{x}^{(M)} \cdot \mathbf{w} \end{bmatrix}$$

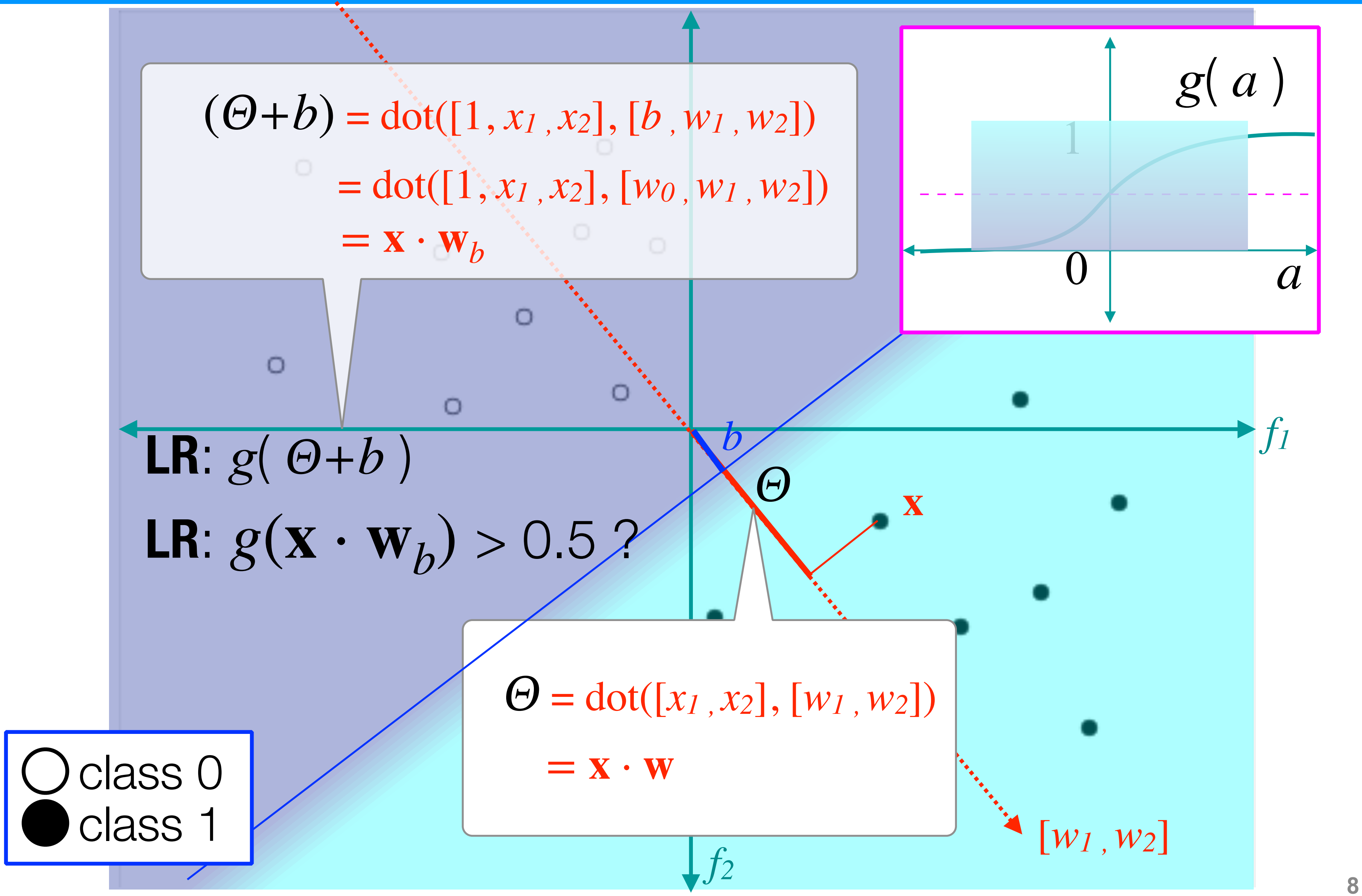
project onto $\mathbf{w}$

map projection to probability via sigmoid

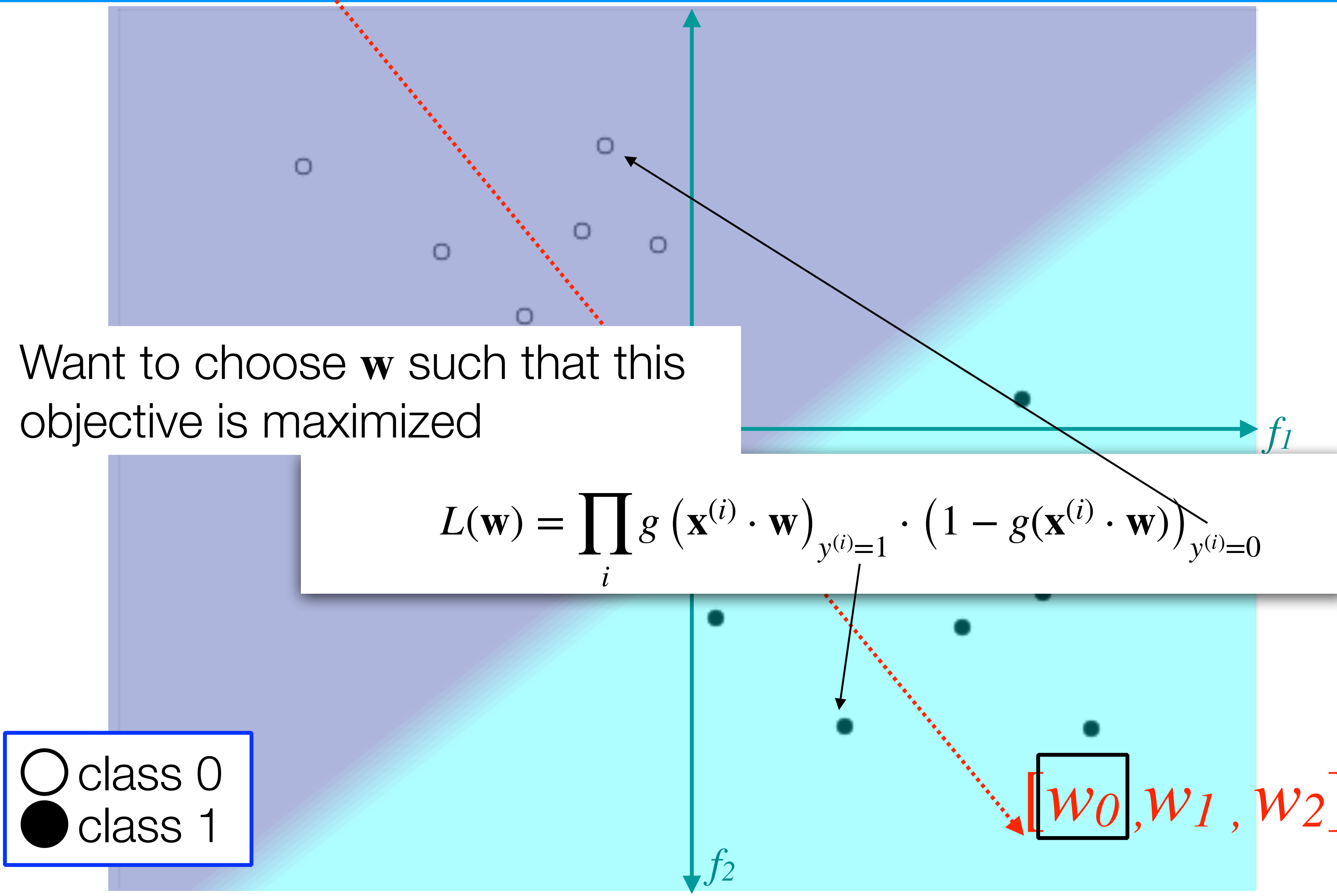


$$g\,(\mathbf{X} \cdot \mathbf{w}) = \begin{bmatrix} g(\mathbf{x}^{(1)} \cdot \mathbf{w}) \\ g(\mathbf{x}^{(2)} \cdot \mathbf{w}) \\ \vdots \\ g(\mathbf{x}^{(M)} \cdot \mathbf{w}) \end{bmatrix} = \begin{bmatrix} p(y^{(1)} = 1 \,|\, \mathbf{x}^{(1)}) \\ p(y^{(2)} = 1 \,|\, \mathbf{x}^{(2)}) \\ \vdots \\ p(y^{(M)} = 1 \,|\, \mathbf{x}^{(M)}) \end{bmatrix}$$

What do weights and intercept define?



Changing $\mathbf{w}$ alters probability



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

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

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?

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

Logistic Regression Optimization Procedure

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

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

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

- **Take** Gradient

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

- **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**

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

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

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

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

```
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