COMPUTER SCIENCE

INDIANA UNIVERSITY

School of Informatics and Computing
Bloomington

Generalized linear models and logistic regression

Reminders/Comments

- Write your name on the assignment to make it easier for AIs
  - assignments must be written in some kind of editor, not by hand
- Assignment 1 marks are released today
- Thought questions 3 for
  - Chapter 5: Generalized Linear Models
  - Chapter 6: Linear classification
- Changed deadlines for thought questions
- Note: will have to balance completing project and last two assignments, so consider getting started early

Thought question

- I assume we don't use Newton's method because the requirements are more strict (smoothness of the function). Are there are any other reasons? From what I understand, Newton's method is much more efficient than Gradient descent.
  - In many cases, smoothness is not the issue for us
  - Rather, computing the Hessian is expensive
  - First-order (just gradient) is O(dn)
  - Second-order (newton with Hessian) is O(d^3 + d^2n)
  - e.g. for d = 10 features, n = 100 samples, first-order = 1000 and second order = 11000; this gets significantly worse for larger d
  - **Compromise**: quasi-Newton methods that keep a small d x m approximation (m < d) to the true d x d Hessian and avoid expensive inverses with a clever order of operations, giving O(dmn)

Thought question

- In Linear Regression we use Least Squared sum. Why do use squares? Why not the least sum of absolute values of the errors?
  - There are multiple answers to this question
  - First, we assumed that p(y l x) is Gaussian distributed —> forming the maximum likelihood optimization results in the squared error
  - However, we did not have to make this assumption; we could have assumed p(y l x) is Laplace distribution, for which the maximum likelihood optimization would give the sum of absolute errors (i.e., l1)
  - We did not do this because
    - (a) there is not a closed form solution for the minimization of the sum of absolute errors, but there is for the squared error
    - (b) using gradient descent with the absolute values is more problematic

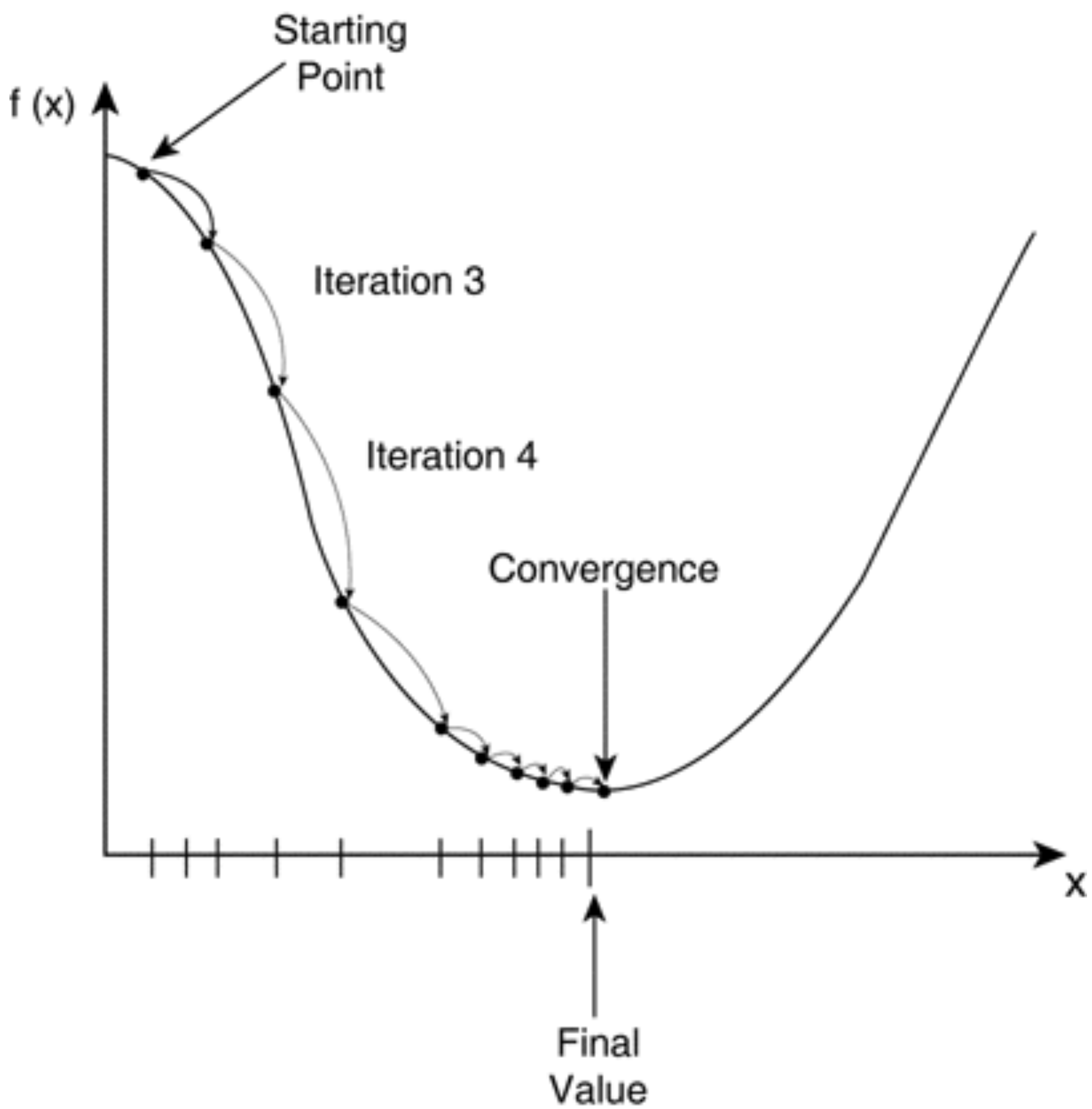
Clarification about assumptions

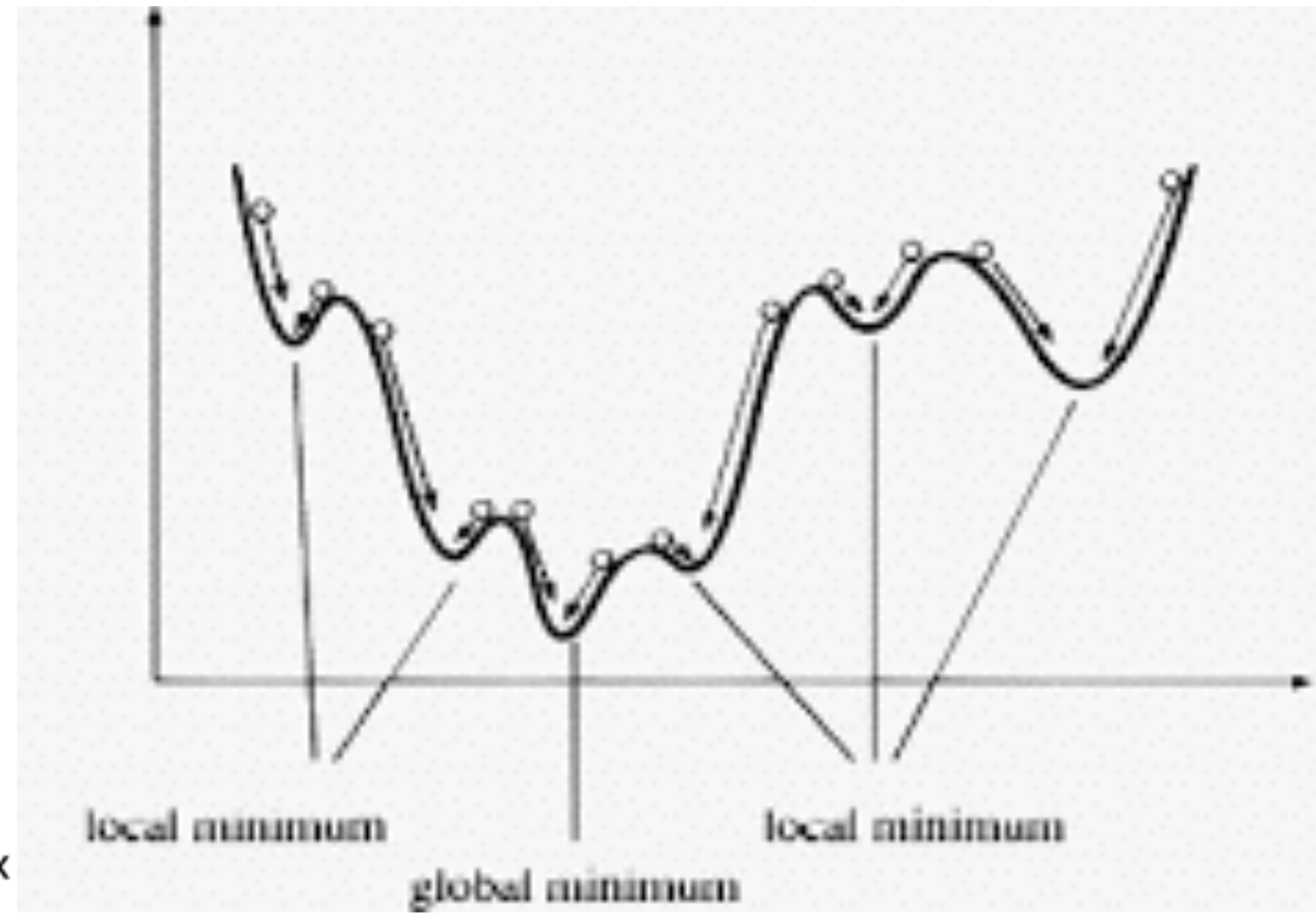
- We have made distributional assumptions for modeling
- In Chapter 2, assumed distributions on variables
  - e.g. commute time variable X was Gamma distribution
  - e.g. multivariate Gaussian distribution on a collection of features
  - these can be pretty strong assumptions on many (complex) variables in a dynamical system
- For conditional distributions, predicting E[y | x] and making distributional assumption on noise
  - intuitively, in most cases, this is not that strong of an assumption, as it is not unreasonable to assume noise simple, but variables complex
  - E[y|x] not a complete picture of p(y | x), but still useful for prediction
  - this is one of the reasons we can gain so much by improving features

Comments from last time

- Convexity of negative log likelihood of (many) exponential families
  - The negative log likelihood of many exponential families is convex, which is an important advantage of the maximum likelihood approach
  - We will focus on natural exponential family distributions (also called regular exponential family distributions)
- Why is convexity important?
  - e.g., why is (sigmoid(xw) - y)^2 not a good choice for binary classification?
  - we'll see that this Euclidean loss (squared loss) results in a non-convex function later

Convex versus nonconvex





Convex function

Non-convex function

How can we check convexity?

- Can check the definition of convexity

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

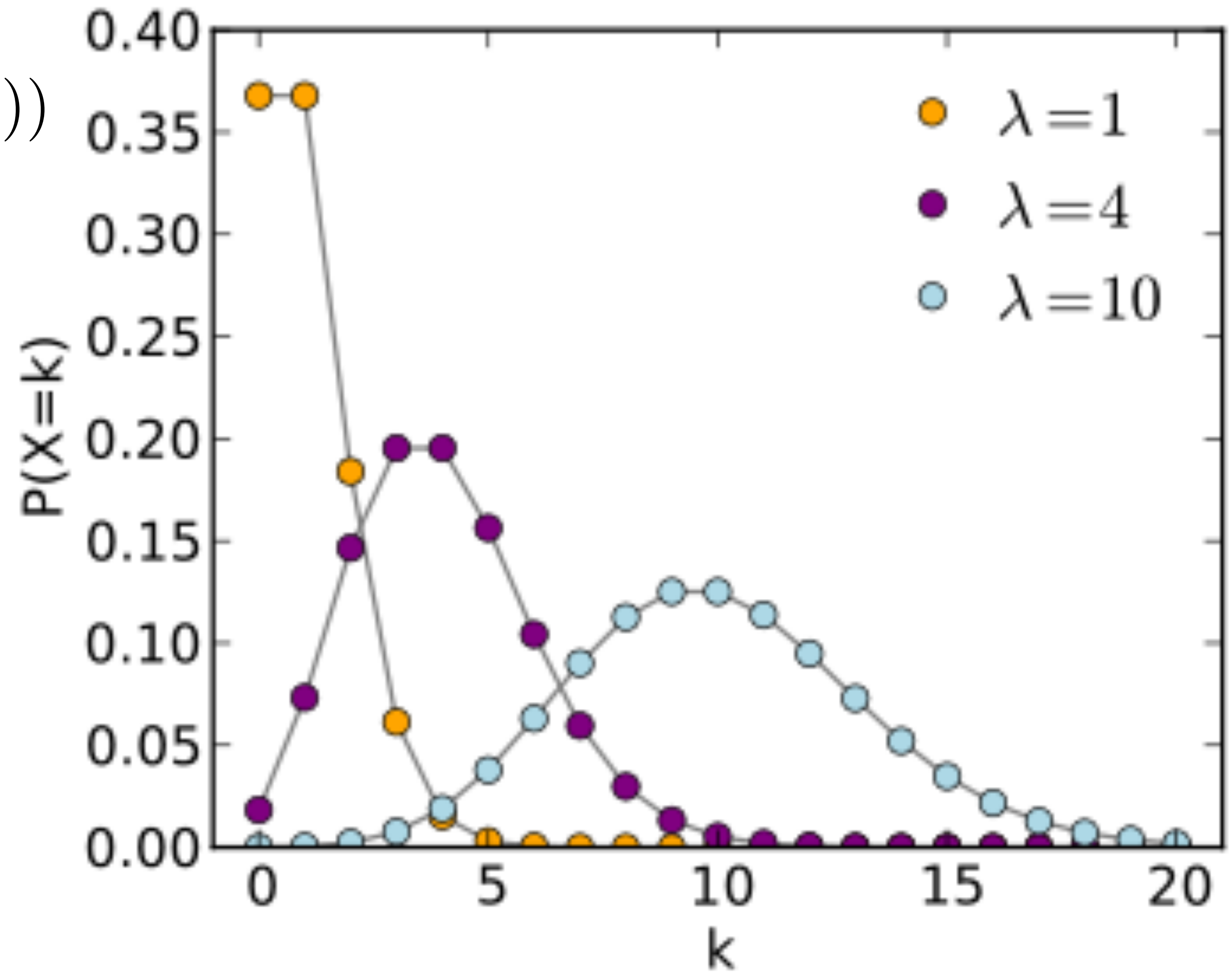
- Can check second derivative for scalar parameters (e.g. λ) and Hessian for multidimensional parameters (e.g., $\mathbf{w}$)
  - e.g., for linear regression (least-squares), the Hessian is $\mathbf{H} = \mathbf{X}^\top \mathbf{X}$ and so clearly positive semi-definite
  - e.g., for Poisson regression, the Hessian of the negative log-likelihood is $\mathbf{H} = \mathbf{X}^\top \mathbf{C} \mathbf{X}$ and so clearly positive semi-definite

- Note: for Poisson regression, in notes used log likelihood, so function concave and Hessian was negative semi-definite

Poisson regression

$$p(y|\mathbf{x}) = \text{Poisson}(y|\lambda = \exp(\mathbf{x}^\top \mathbf{w}))$$

1. $\log(E[y|\mathbf{x}]) = \boldsymbol{\omega}^T \mathbf{x}$
2. $p(y|\mathbf{x}) = \text{Poisson}(\lambda)$



$$\begin{aligned} p(x|\boldsymbol{\theta}) &= \exp\left(\sum_{i=1}^{m} \theta_i t_i(x) - a(\boldsymbol{\theta}) + b(x)\right) \\ &= \exp\left(\boldsymbol{\theta}^T \boldsymbol{t}(x) - a(\boldsymbol{\theta}) + b(x)\right), \end{aligned}$$

where $\boldsymbol{t}(x) = (t_1(x), t_2(x), \ldots, t_m(x))$.

$$p(y|\lambda) = \exp(y \log \lambda - \lambda - \log y!)$$

$$\theta = \log \lambda, t(y) = y, a(\theta) = \exp^{\theta}, b(y) = -\log y!$$

GLMs

$$\nabla a(\theta) = g(\theta) \text{ where } g(\theta) = E[t(x)], g = f^{-1}$$

e.g. for conditional distribution on y:

$$\theta = \mathbf{x}^\top \mathbf{w}$$

$$a(\theta) = \exp(\theta)$$

$$g(\theta) = \exp(\theta) = \exp(\mathbf{x}^\top \mathbf{w}) = E[y|\mathbf{x}]$$

$$\begin{aligned} ll(\boldsymbol{\theta}) &= \log \prod_{i=1}^{n} e^{\boldsymbol{\theta}^T \boldsymbol{t}(x_i) - a(\boldsymbol{\theta}) + b(x_i)} \\ &= \sum_{i=1}^{n} \boldsymbol{\theta}^T \boldsymbol{t}(x_i) - n \cdot a(\boldsymbol{\theta}) + \sum_{i=1}^{n} b(x_i). \end{aligned}$$

Note: this is a generic x,
here we intend y
not the features x

GLM log-likelihood

$$\begin{aligned} p(x|\boldsymbol{\theta}) &= \exp\left(\sum_{i=1}^{m} \theta_i t_i(x) - a(\boldsymbol{\theta}) + b(x)\right) \\ &= \exp\left(\boldsymbol{\theta}^T \boldsymbol{t}(x) - a(\boldsymbol{\theta}) + b(x)\right), \end{aligned}$$

This is a generic x, here we intend y not the features x

Final log-likelihood and gradient in terms of w

$$\begin{aligned} ll(\mathbf{w}) &= \log \prod_{i=1}^{n} e^{\boldsymbol{\theta}^T \boldsymbol{t}(x_i) - a(\boldsymbol{\theta}) + b(x_i)} \\ &= \sum_i \sum_m \theta_m t_m(x_i) - n \cdot a(\boldsymbol{\theta}) + \sum_i b(x_i) \\ &= \sum_i ll_i(\mathbf{w}) \end{aligned}$$

$$\frac{\partial ll_i(\mathbf{w})}{\partial w_j} = \sum_m \frac{\partial \theta_m}{\partial w_j} t_m(x_i) - \frac{\partial a(\boldsymbol{\theta})}{\partial w_j}$$

Exercise: what do each of these terms look like for Poisson regression?

$$p(y|\lambda) = \exp(y \log \lambda - \lambda - \log y!)$$

$$\theta = \log \lambda, t(y) = y, a(\theta) = \exp^{\theta}, b(y) = -\log y!$$

Logistic regression

1. $\text{logit}(E[y|\mathbf{x}]) = \boldsymbol{\omega}^T\mathbf{x}$
2. $p(y|\mathbf{x}) = \text{Bernoulli}(\alpha)$

where $\text{logit}(x) = \ln\frac{x}{1-x}$, $y \in \{0,1\}$, and $\alpha \in (0,1)$

$$\alpha = p(y=1|\mathbf{x})$$

$$f(\mathbf{w}^\top\mathbf{x}) = \text{logit}(\mathbf{w}^\top\mathbf{x})$$

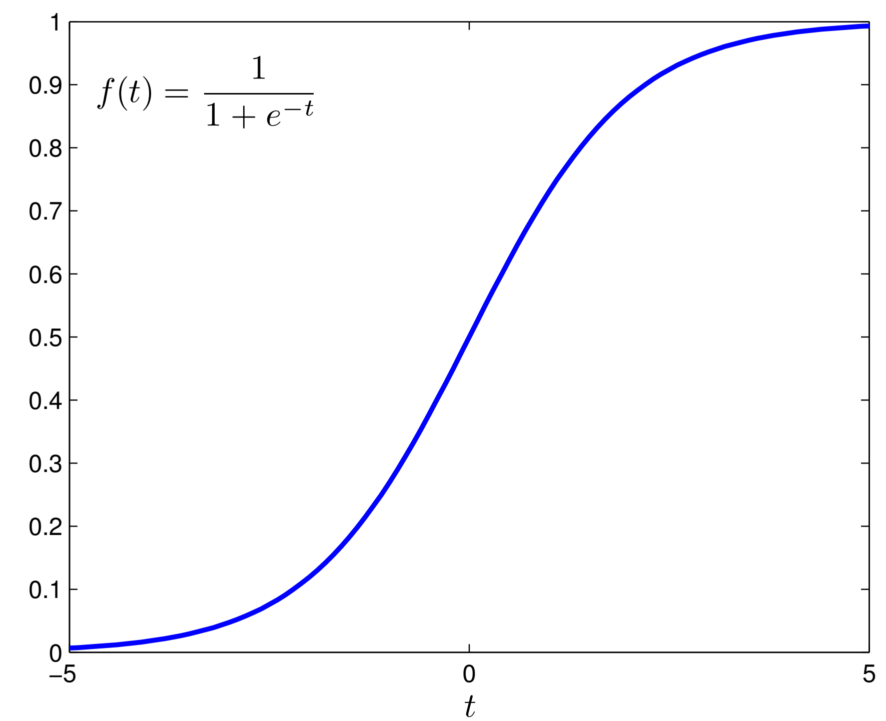
$$g(\mathbf{w}^\top\mathbf{x}) = f^{-1}(\mathbf{w}^\top\mathbf{x})$$

$$= \text{sigmoid}(\mathbf{w}^\top\mathbf{x})$$

$$= E[y|\mathbf{x}]$$

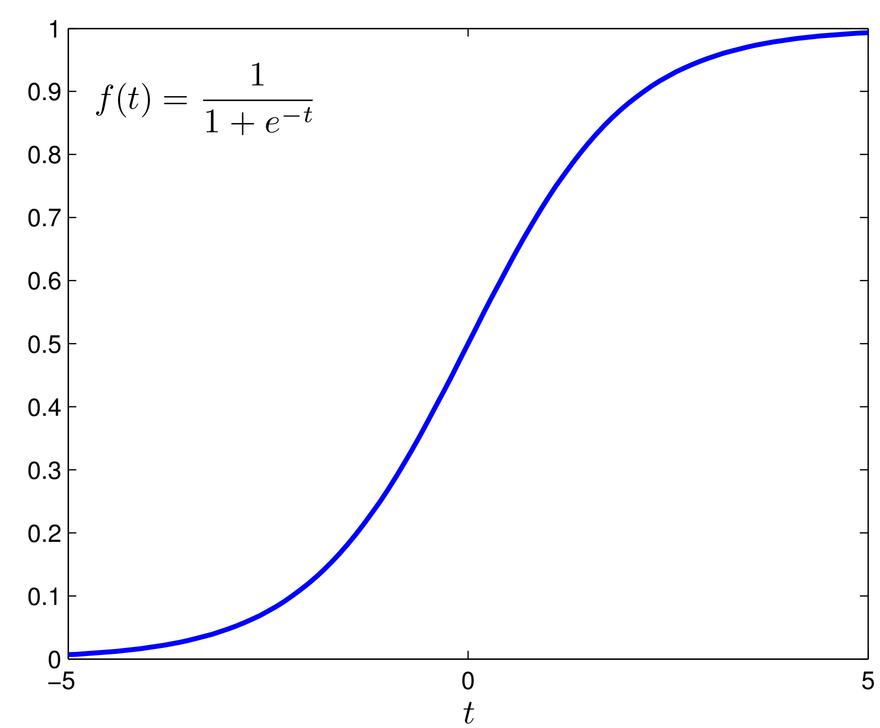
$$E[y|\mathbf{x}] = \frac{1}{1+e^{-\boldsymbol{\omega}^T\mathbf{x}}}$$

$$p(y|\mathbf{x}) = \left(\frac{1}{1+e^{-\omega^T\mathbf{x}}}\right)^y \left(1-\frac{1}{1+e^{-\omega^T\mathbf{x}}}\right)^{1-y}.$$



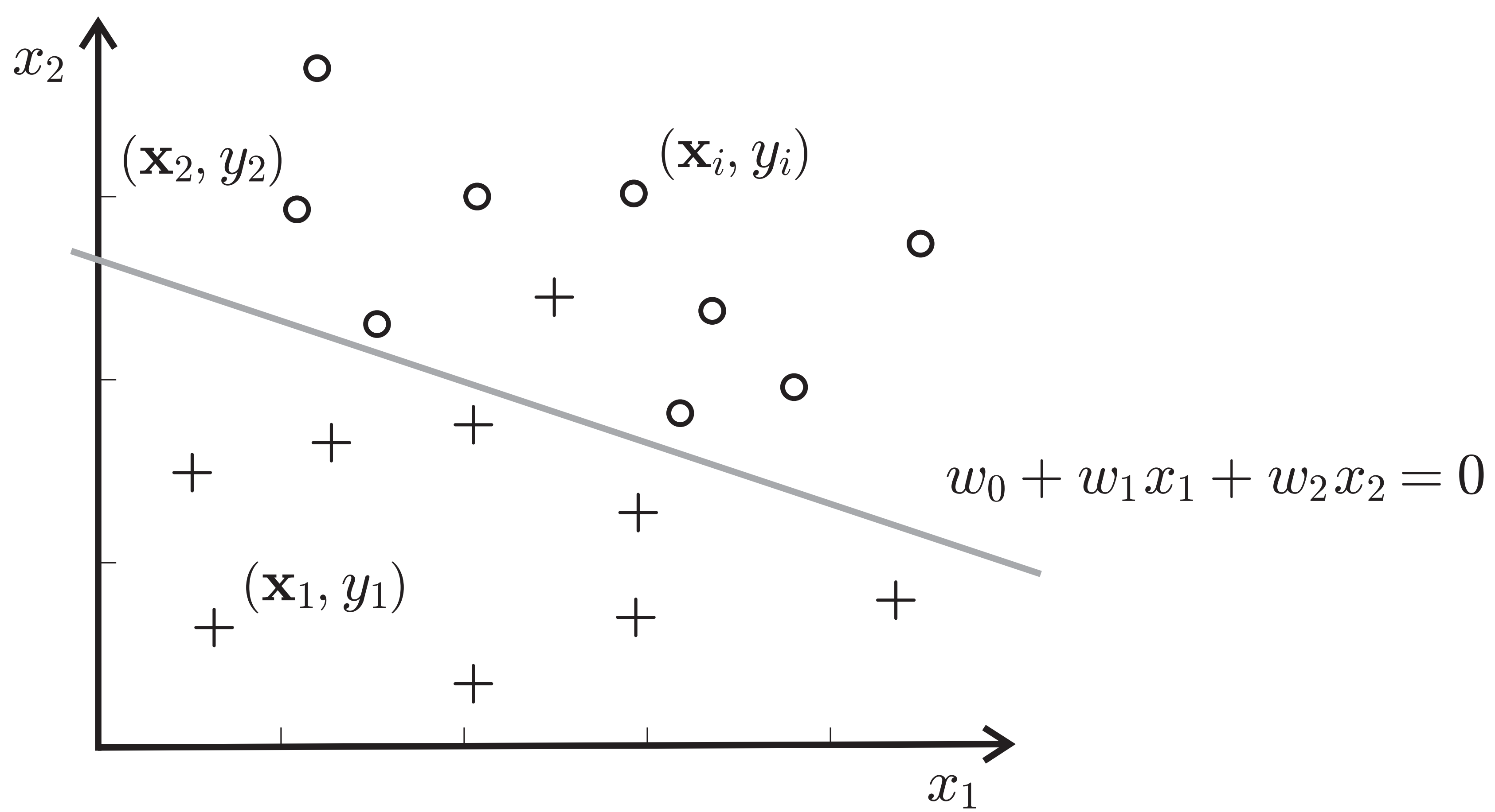
Prediction with logistic regression

- So far, we have used the prediction g(xw)
  - eg., xw for linear regression, exp(xw) for Poisson regression
- For binary classification, want to output 0 or 1, rather than the probability value p(y=1 | x) = sigmoid(xw)
- Sigmoid has few values xw mapped close to 0.5; most values somewhat larger than 0 are mapped close to 0 (and vice versa for 1)
- Decision threshold:
  - sigmoid(xw) < 0.5 is class 0
  - sigmoid(xw) > 0.5 is class 1



Logistic regression is a linear classifier

- Hyperplane $\mathbf{w}^\top \mathbf{x} = 0$ separates the two classes
  - P(y=1 | x, w) > 0.5 only when $\mathbf{w}^\top \mathbf{x} \geq 0.$
  - P(y=0 | x, w) > 0.5 only when P(y=1 | x, w) < 0.5, which happens when $\mathbf{w}^\top \mathbf{x} < 0$



Whiteboard

- Logistic regression
  - maximum likelihood
  - stochastic optimization
- Next class:
  - issues with minimizing Euclidean distance for sigmoid
  - multiclass with multinomial logistic regression
  - generative approach: naive Bayes