Logistic Regression

Machine Learning
CS5824/ECE5424
Bert Huang
Virginia Tech

Outline

- Review conditional probability and classification
- Linear parameterization and logistic function
- Gradient descent
  - Other optimization methods
- Regularization

Classification and Conditional Probability

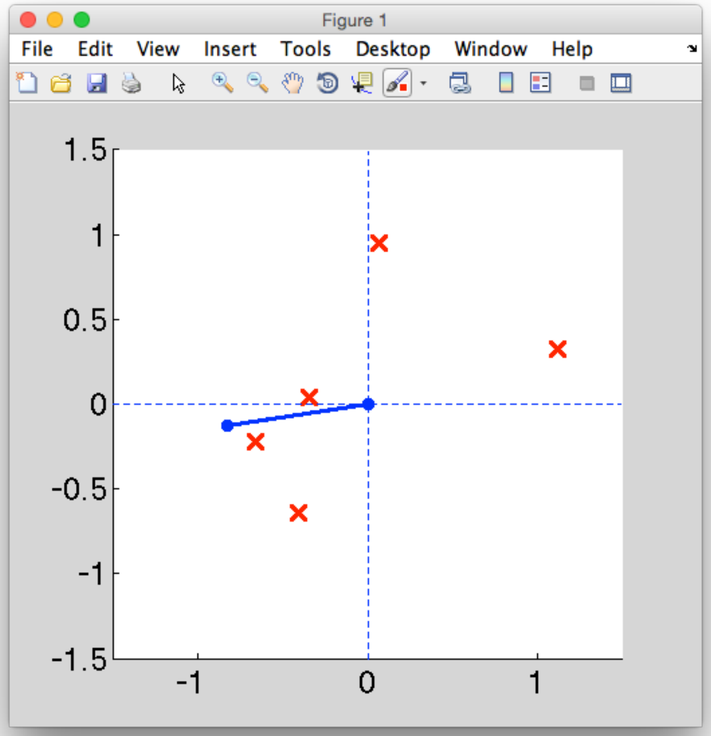
- Discriminative learning: maximize p(y|x)
- Naive Bayes: learn p(x|y) and p(y)
- Today: parameterize p(y|x) and directly learn p(y|x)

Parameterizing p(y|x)

$$p(y|x) := f$$

$$f : \mathbb{R}^d \rightarrow [0, 1]$$

$$f(x) := \frac{1}{1 + \exp(-w^\top x)}$$

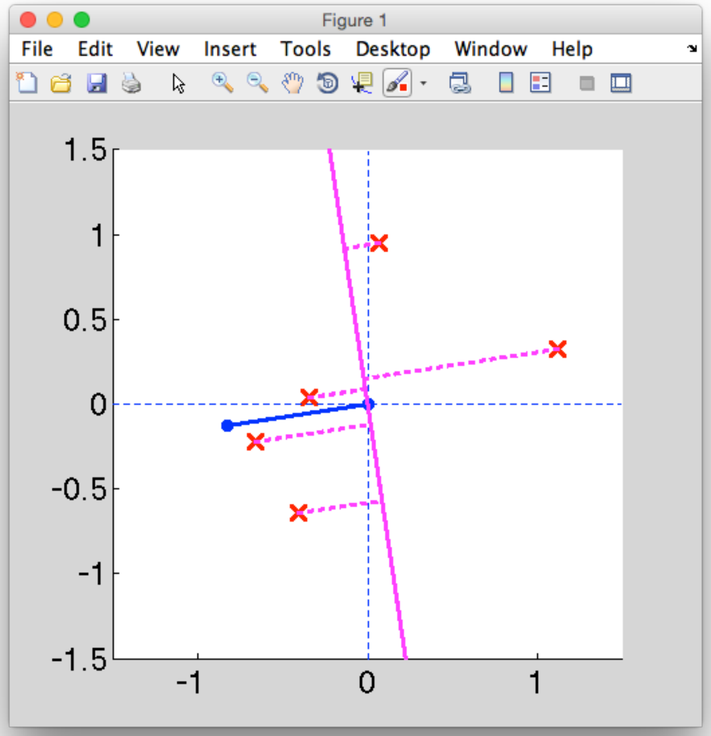


Parameterizing p(y|x)

$$p(y|x) := f$$

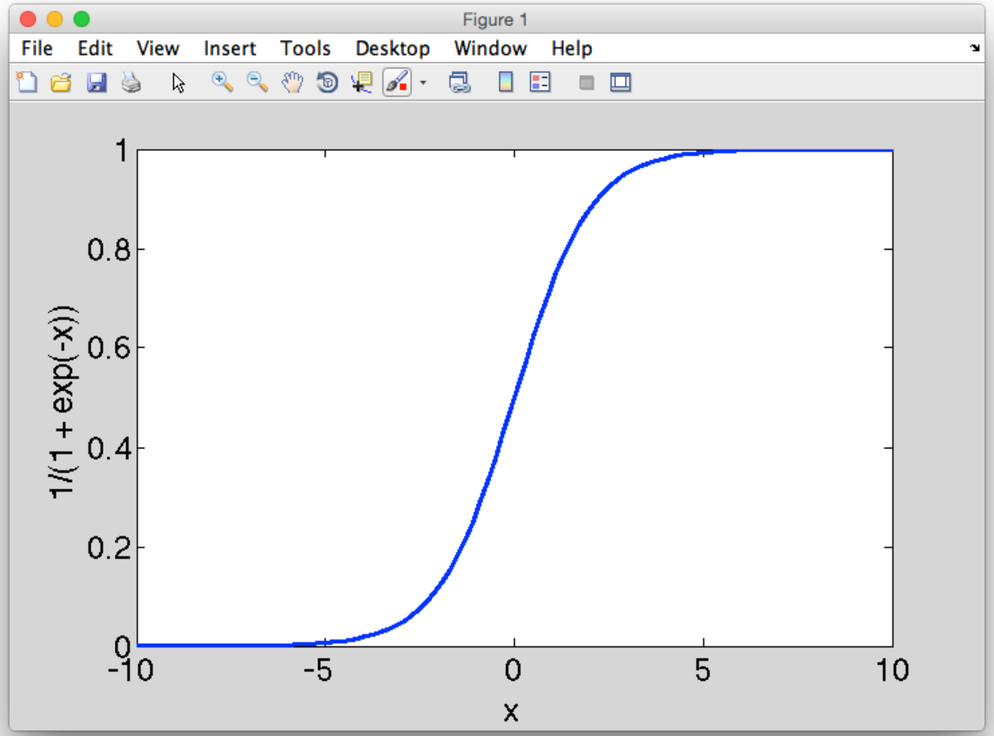
$$f : \mathbb{R}^d \rightarrow [0, 1]$$

$$f(x) := \frac{1}{1 + \exp(-w^\top x)}$$



Logistic Function

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



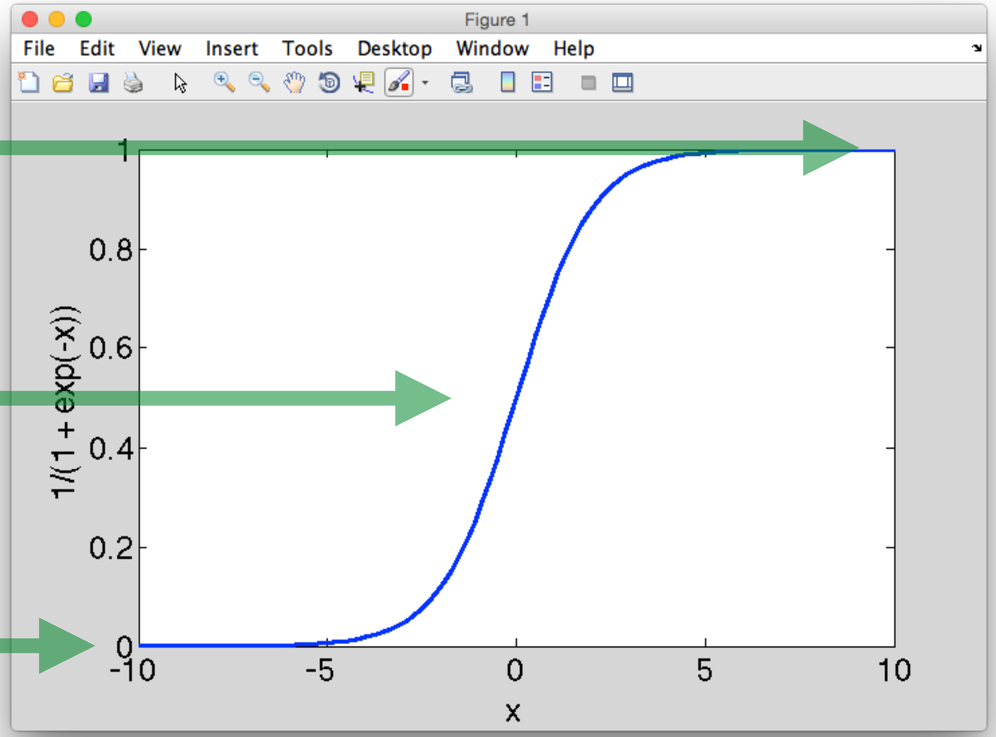
Logistic Function

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

$$\lim_{x \to \infty} \sigma(x) = \lim_{x \to \infty} \frac{1}{1 + \exp(-x)} = \frac{1}{1} = 1.0$$

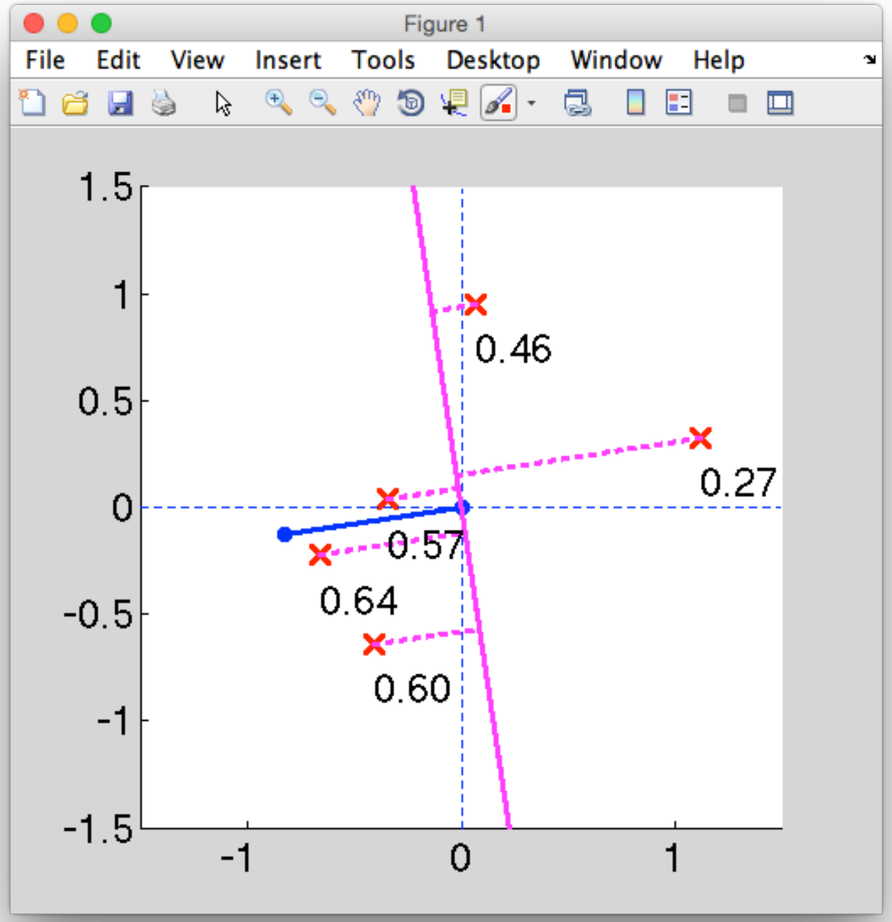
$$\sigma(0) = \frac{1}{1 + \exp(-0)} = \frac{1}{1 + 1} = 0.5$$

$$\lim_{x \to -\infty} \sigma(x) = \lim_{x \to -\infty} \frac{1}{1 + \exp(-x)} = 0.0$$



From Features to Probability

$$f(x) := \frac{1}{1 + \exp(-w^\top x)}$$



Likelihood Function

$$y \in \{-1, +1\} \qquad p_w(y|x) := \begin{cases} (1 + \exp(-w^\top x))^{-1} & \text{if } y = +1 \\ (1 + \exp(+w^\top x))^{-1} & \text{if } y = -1 \end{cases}$$

$$1 - (1 + \exp(-w^\top x))^{-1}$$

$$= 1 - \frac{1}{1 + \exp(-w^\top x)} \qquad = \frac{1 + \exp(-w^\top x)}{1 + \exp(-w^\top x)} - \frac{1}{1 + \exp(-w^\top x)}$$

$$= \frac{\exp(-w^\top x)}{1 + \exp(-w^\top x)} \qquad = \left(\frac{1 + \exp(-w^\top x)}{\exp(-w^\top x)} \right)^{-1}$$

$$= \left(\frac{1}{\exp(-w^\top x)} + \frac{\exp(-w^\top x)}{\exp(-w^\top x)} \right)^{-1} \qquad = (1 + \exp(w^\top x))^{-1}$$

Likelihood Function

$$y \in \{-1, +1\}$$

$$p_w(y|x) := \begin{cases} (1 + \exp(-w^\top x))^{-1} & \text{if } y = +1 \\ (1 + \exp(+w^\top x))^{-1} & \text{if } y = -1 \end{cases}$$

$$= (1 + \exp(-y w^\top x))^{-1}$$

$$L(w) = \prod_{i=1}^{n} (1 + \exp(-y_i w^\top x_i))^{-1}$$

$$\log L(w) = -\sum_{i=1}^{n} \log(1 + \exp(-y_i w^\top x_i))$$

Likelihood Function

$$\log L(w) = -\sum_{i=1}^{n} \log(1 + \exp(-y_i w^\top x_i))$$

$$\hat{w} \leftarrow \operatorname*{argmin}_w \sum_{i=1}^{n} \log(1 + \exp(-y_i w^\top x_i)) \quad := \text{nll}(w)$$

negative log likelihood

$$\nabla_w \text{nll} = -\sum_{i=1}^{n} \left(1 - \frac{1}{1 + \exp(-y_i w^\top x_i)}\right) y_i x_i$$

$$\hat{w} \leftarrow \operatorname*{argmin}_w \sum_{i=1}^{n} \log(1 + \exp(-y_i w^\top x_i)) \;\; := \text{nll}(w)$$

$$\nabla_w \text{nll} = \sum_{i=1}^{n} \left(\frac{1}{1 + \exp(-y_i w^\top x_i)} \times \nabla_w \left(1 + \exp(-y_i w^\top x_i)\right) \right)$$

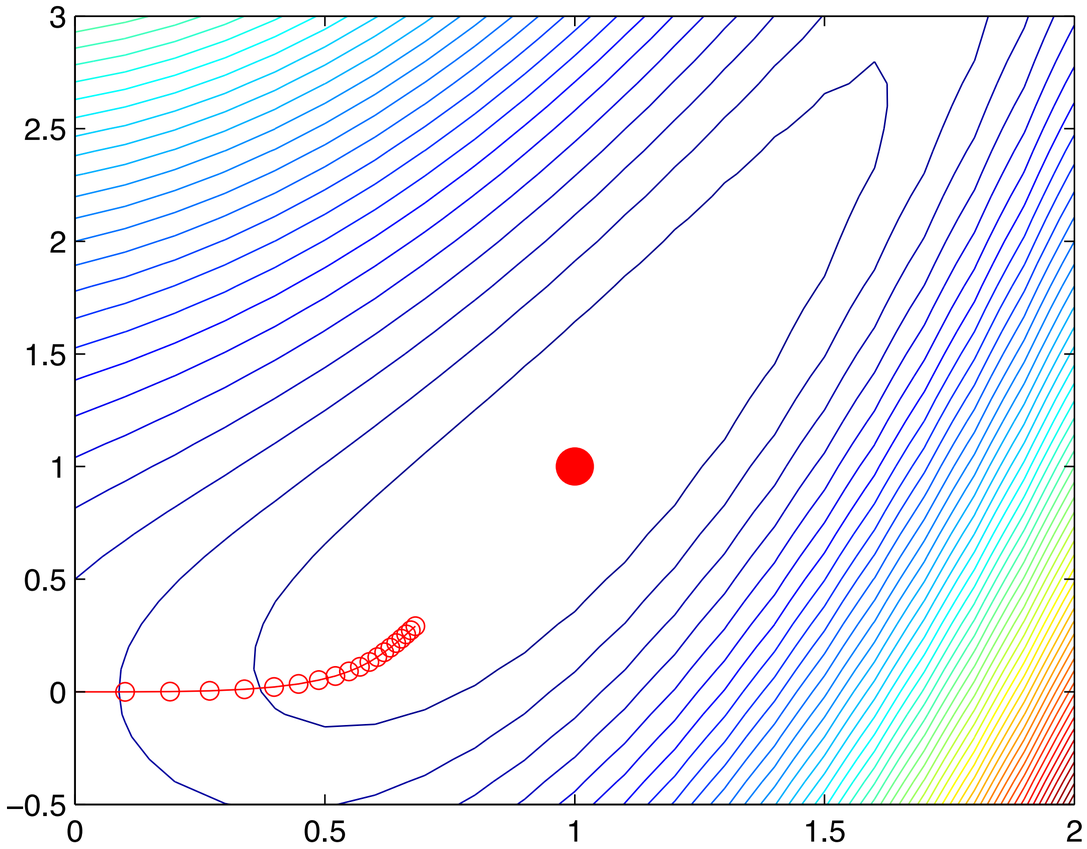
$$0 - \exp(-y_i w^\top x_i) y_i x_i$$

$$\nabla_w \text{nll} = \sum_{i=1}^{n} \left(\frac{-\exp(-y_i w^\top x_i) y_i x_i}{1 + \exp(-y_i w^\top x_i)} \right)$$

$$= \sum_{i=1}^{n} - \left(\frac{\exp(-y_i w^\top x_i)}{1 + \exp(-y_i w^\top x_i)} \right) y_i x_i = - \sum_{i=1}^{n} \left(\frac{-1 + 1 + \exp(-y_i w^\top x_i)}{1 + \exp(-y_i w^\top x_i)} \right) y_i x_i$$

$$\nabla_w \text{nll} = - \sum_{i=1}^{n} \left(1 - \frac{1}{1 + \exp(-y_i w^\top x_i)} \right) y_i x_i = - \sum_{i=1}^{n} (1 - p_w(y_i | x_i))\, y_i x_i := g$$

Gradient Descent



$$w_{t+1} \rightarrow w_t - \eta_t g_t$$

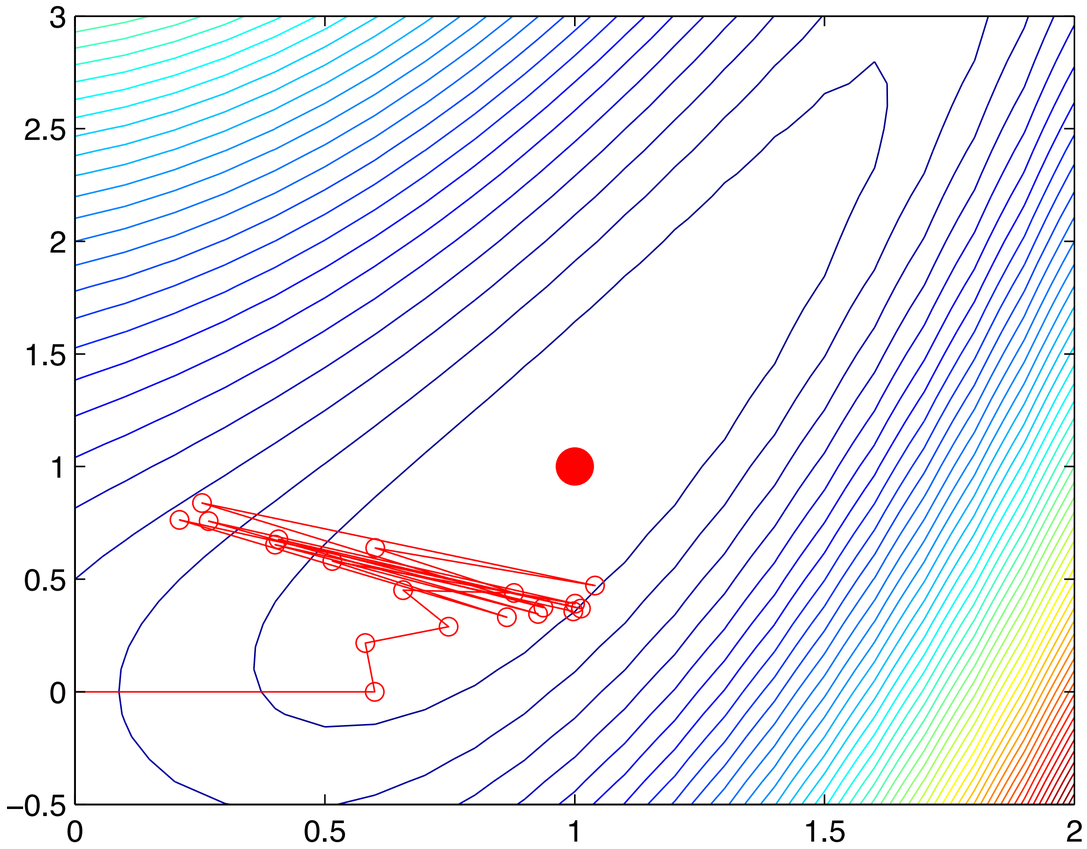
e.g., $\eta_t = \frac{1}{t}$

$\eta_t = \frac{1}{\sqrt{t}}$

$\eta_t = 1$

Image from Murphy Ch. 8

Gradient Descent



$$w_{t+1} \rightarrow w_t - \eta_t g_t$$

e.g., $\eta_t = \frac{1}{t}$

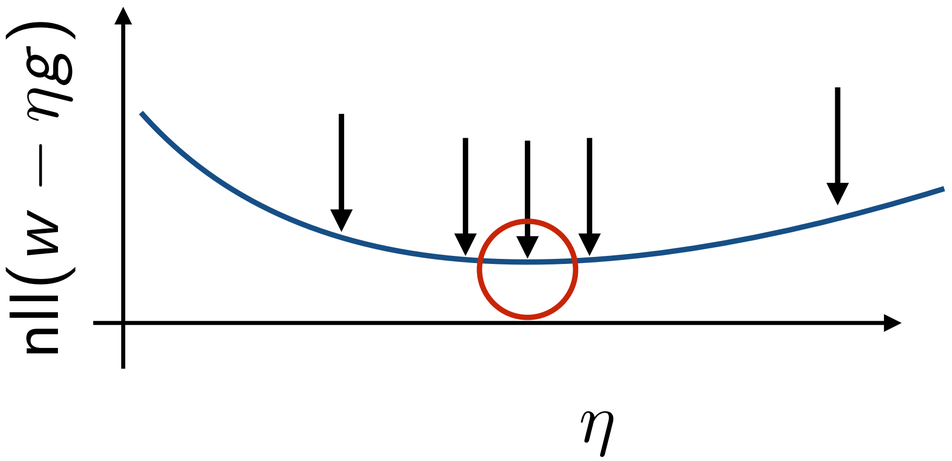
$\eta_t = \frac{1}{\sqrt{t}}$

$\eta_t = 1$

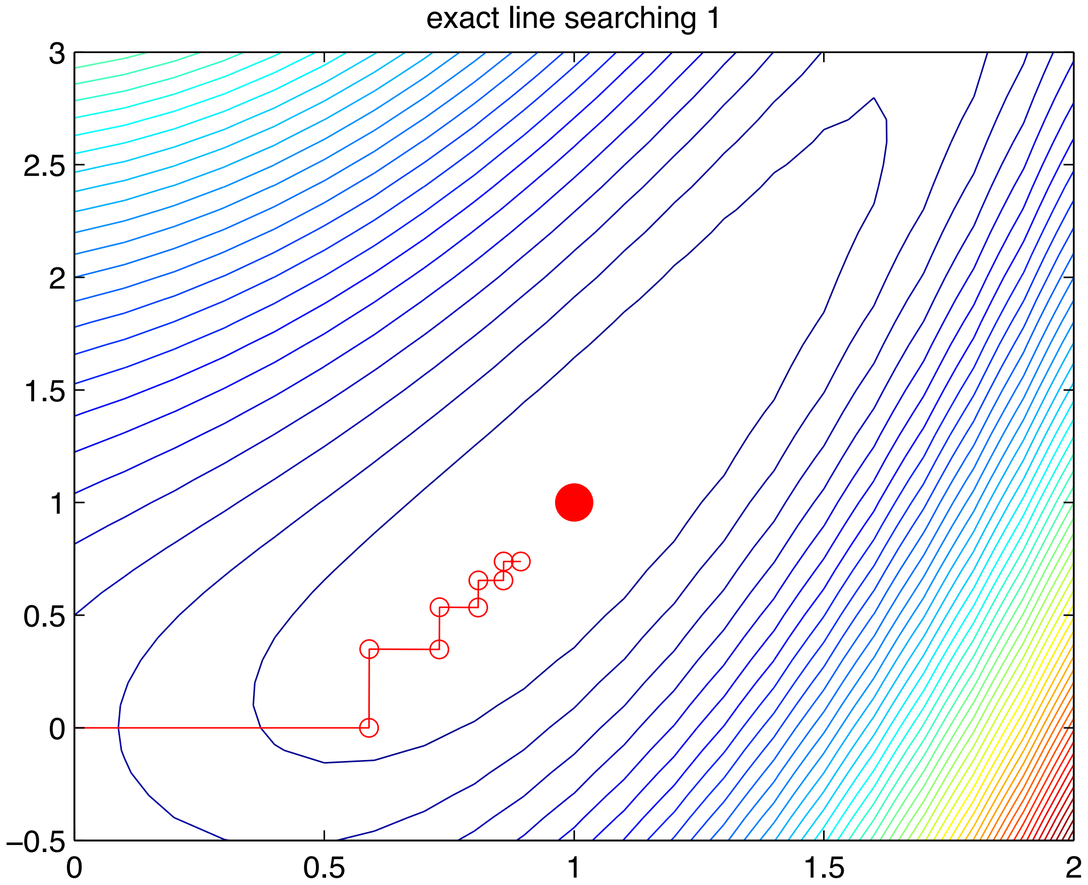
Image from Murphy Ch. 8

Line Search

$$\eta_t = \underset{\eta}{\text{argmin}} \ \text{nll}(w - \eta g)$$



Line Search



$$\eta_t = \underset{\eta}{\text{argmin}} \ \text{nll}(w - \eta g)$$

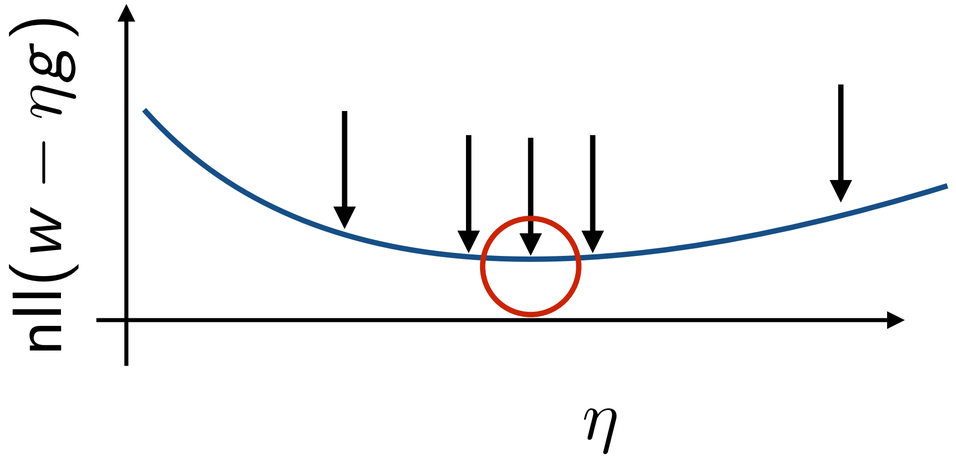
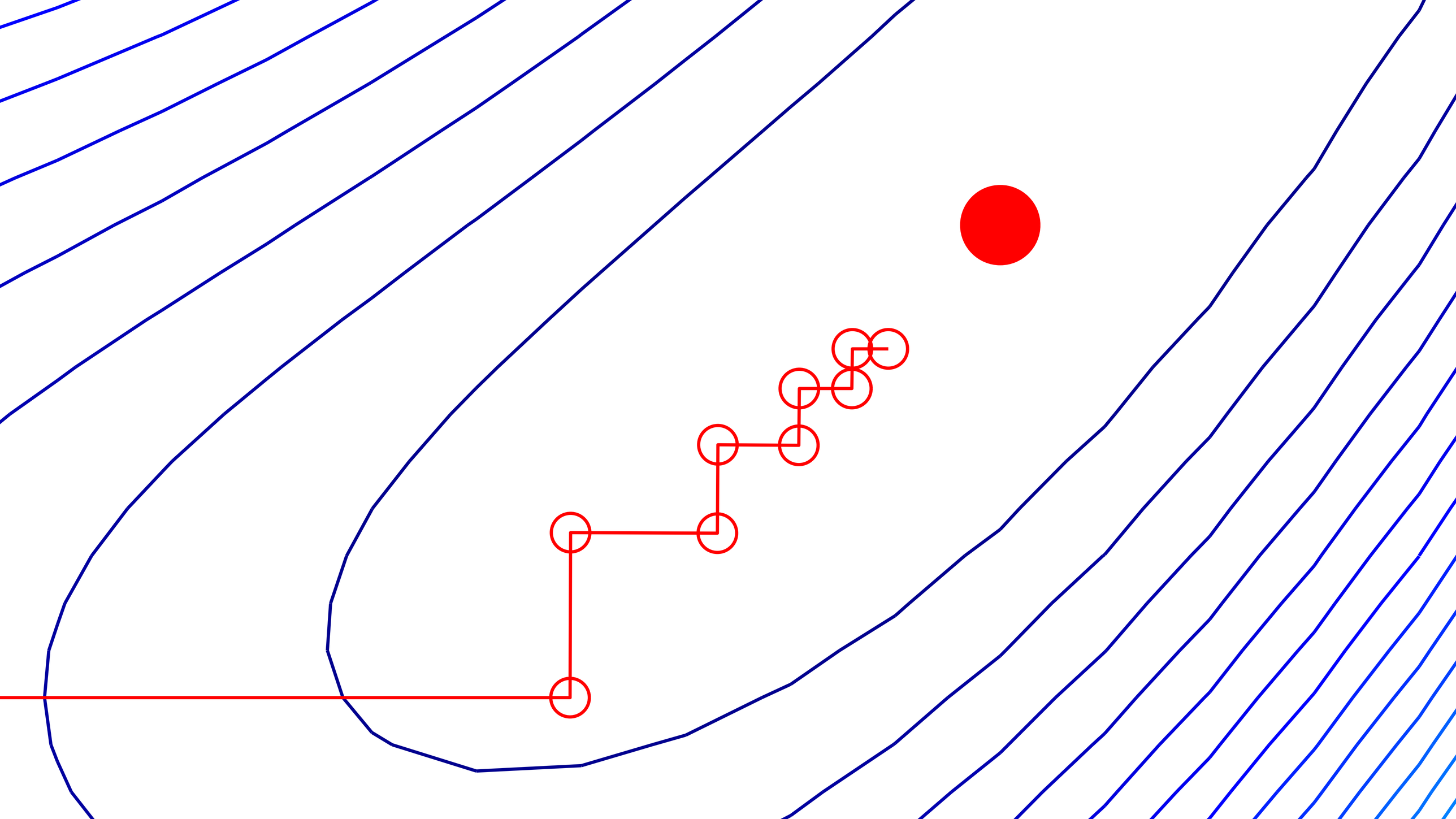


Image from Murphy Ch. 8



Momentum

- Heavy ball method

$$s_t \leftarrow -g_t + \beta_t s_{t-1}$$

$$w_t \leftarrow w_{t-1} + \eta_t s_t$$

- Momentum term retains “velocity” from previous steps
- Avoids sharp turns

Second-Order Optimization

- Newton's method
- Approximate function with quadratic and minimize quadratic exactly
- Requires computing Hessian (matrix of second derivatives)
- Various approximation methods (e.g., L-BFGS)

Regularization

$$L(w) = p(w) \prod_{i=1}^{n} (1 + \exp(-y_i w^\top x_i))^{-1}$$ posterior

$$p(w) = \mathcal{N}\left(0, \frac{1}{\lambda} \mathrm{I}\right)$$ prior

$$-\log L(w) = -\frac{\lambda}{2} w^\top w + \sum_{i=1}^{n} \log(1 + \exp(-y_i w^\top x_i))$$ neg log posterior (regularized nll)

$$\nabla_w \mathrm{nll} = -\lambda w - \sum_{i=1}^{n} \left(1 - \frac{1}{1 + \exp(-y_i w^\top x_i)}\right) y_i x_i$$ gradient

Summary

- Review conditional probability and classification
- Linear parameterization and logistic function
- Gradient descent
  - Other optimization methods
- Regularization