Logistic Regression

STAT 37710 / CMSC 35300
Rebecca Willett and Yuxin Chen

Recap: loss and risk

Definition: Loss

Given the domain set $\mathcal{X}$, the label set $\mathcal{Y}$, a prediction function (also knowns as a *hypothesis*) $h : \mathcal{X} \to \mathcal{Y}$, and a loss function ℓ, the loss of an example (x, y) is $\ell(h(x), y)$.
e.g. *squared loss*: $\ell(\hat{y}, y) = \|y - \hat{y}\|^2$.

Definition: Bayes Risk

The quality of predictor h is measured by the expected loss, known as the Bayes risk:

$$R(h) := \mathbb{E}_{X,Y}\left[\ell(h(X), Y)\right].$$

Risk in classification

- Consider 0/1 loss function for classification:

$$\ell\big(h(x), y\big) = \begin{cases} 1 & y \neq h(x), \\ 0 & \text{otherwise} \end{cases}$$

The risk of a prediction function h is

$$R(h) = \mathbb{E}_{X,Y}[[Y \neq h(X)]]$$

Suppose we knew $P(X, Y)$, which h minimizes the risk?

Risk in classification

- Consider 0/1 loss function for classification:

$$\ell\big(h(x), y\big) = \begin{cases} 1 & y \neq h(x), \\ 0 & \text{otherwise} \end{cases}$$

The risk of a prediction function h is

$$R(h) = \mathbb{E}_{X,Y}[[Y \neq h(X)]]$$

Suppose we knew $P(X, Y)$, which h minimizes the risk?

$$\begin{aligned} h^*(x) &= \arg\min_{\hat{y}} \mathbb{E}_Y[[Y \neq \hat{y} \mid X = x]] \\ &= \arg\max_{\hat{y}} P(Y = \hat{y} \mid X = x) \end{aligned}$$

Bayes' optimal classifier

- Assuming the data is generated iid according to

$$(x_i, y_i) \sim P(X, Y)$$

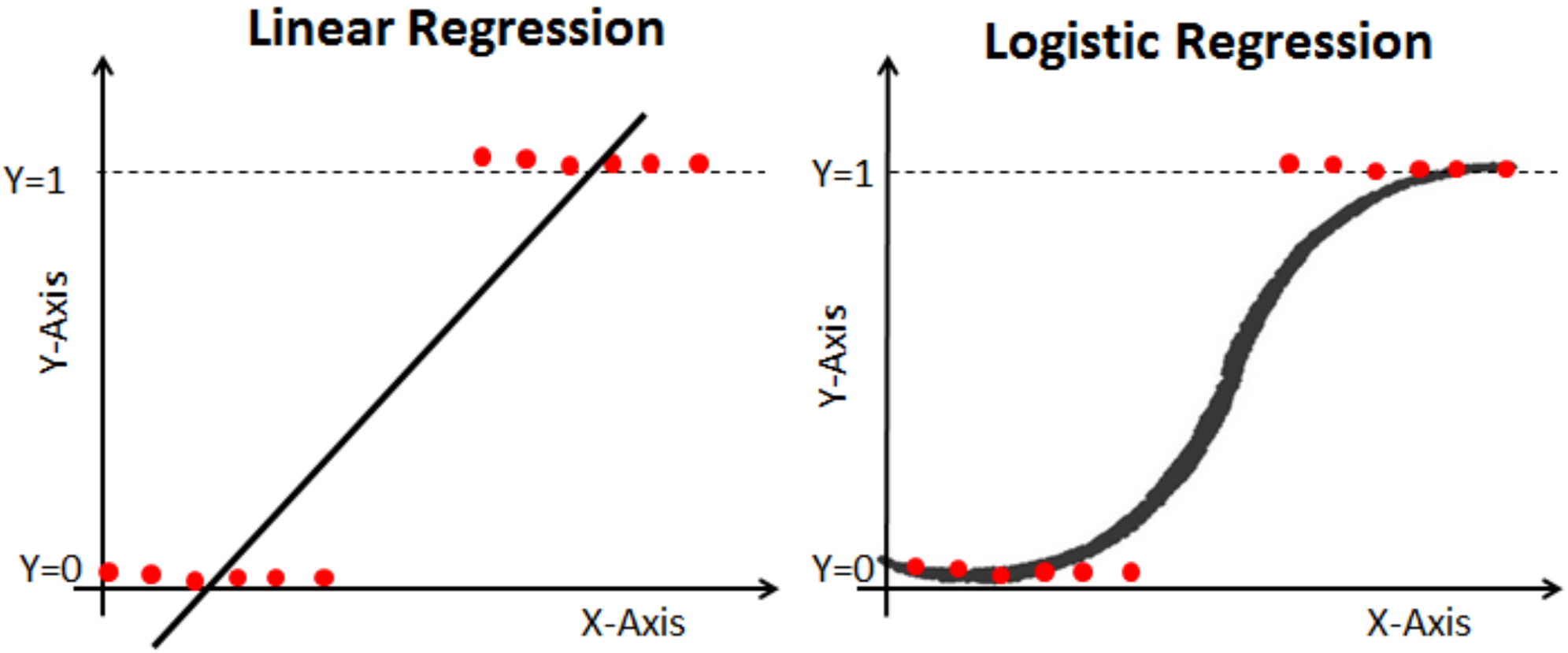
- The hypothesis h^* minimizing $R(h) = \mathbb{E}_{X,Y}[[Y \neq h(X)]]$ is given by the most probable class

$$h^*(x) = \arg\max_y P(Y = y \mid X = x)$$

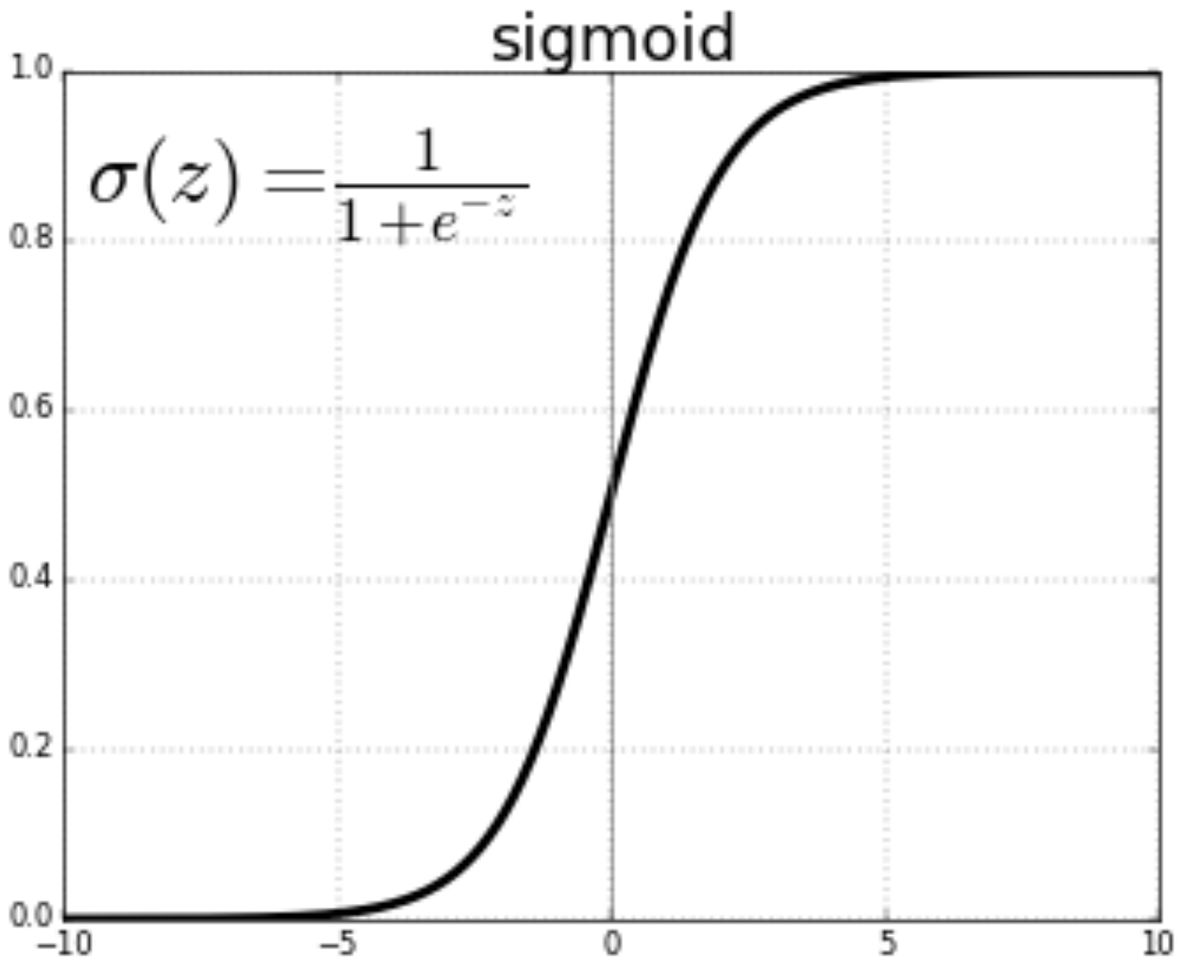
- This (in practice unattainable) hypothesis is called the Bayes' optimal predictor for the 0/1 loss. Thus, natural approach is again to estimate $P(Y|X)$

Logistic regression

- Use regression model for the class probability



Link function for logistic regression



- Link function

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

Logistic regression

- Logistic regression (a classification method) replaces the assumption of Gaussian noise (squared loss) by independently, but not identically distributed Bernoulli noise:

$$P(y \mid x, w) = \mathsf{Bernoulli}(y; \sigma(w^\top x))$$

Parameter estimation for logistic regression

- (Generalized) method of moments estimation
- Maximum likelihood estimation
- Maximum a posteriori estimation

Method of moments estimation for Logistic Regression

- Consider the following function

$$\phi(x, y) = \left(y - \sigma\left(w^\top x\right)\right) x$$

Method of moments estimation for Logistic Regression

- Consider the following function

$$\phi(x, y) = \left(y - \sigma\left(w^\top x\right)\right) x$$

- Apply moment mapping to get the theoretical moments

$$M(w) = \mathbb{E}_{X,Y}\left[\left(Y - \sigma\left(w^\top X\right)\right) X\right]$$

$$= \mathbb{E}_X\Big[\underbrace{\mathbb{E}_Y\left[Y - \sigma\left(w^\top X\right)\right]}_{=0}\Big|X\Big]=0$$

Method of moments estimation for Logistic Regression

- Consider the following function

$$\phi(x, y) = \left(y - \sigma\left(w^\top x\right)\right) x$$

- Apply moment mapping to get the theoretical moments

$$M(w) = \mathbb{E}_{X,Y}\left[\left(Y - \sigma\left(w^\top X\right)\right) X\right]$$
$$= \mathbb{E}_X\Big[\underbrace{\mathbb{E}_Y\left[Y - \sigma\left(w^\top X\right)\right]}_{=0}\Big|X\Big]=0$$

- Sample moments:

$$\hat{m} = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i, y_i) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sigma\left(w^\top x_i\right)\right)\cdot x_i$$

Method of moments estimation for Logistic Regression

- Consider the following function

$$\phi(x, y) = \left(y - \sigma\left(w^\top x\right)\right) x$$

- Apply moment mapping to get the theoretical moments

$$M(w) = \mathbb{E}_{X,Y}\left[\left(Y - \sigma\left(w^\top X\right)\right) X\right]$$
$$= \mathbb{E}_X\Big[\underbrace{\mathbb{E}_Y\left[Y - \sigma\left(w^\top X\right)\right]}_{=0}\Big|X\Big]=0$$

- Sample moments:

$$\hat{m} = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i, y_i) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sigma\left(w^\top x_i\right)\right)\cdot x_i$$

- Practically, minimize $(M(w) - \hat{m})^\top (M(w) - \hat{m})$ to get $\widehat{w}$

MLE for logistic regression

$$
\begin{aligned}
w^* \in \arg\max_w P(D \mid w) &= \arg\max_w \prod_{i=1}^{n} P(y_i \mid x_i, w) \\
&= \arg\max_w \sum_{i=1}^{n} \log P(y_i \mid x_i, w) \\
&= \arg\min_w \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i w^\top x_i\right)\right)
\end{aligned}
$$

MLE for logistic regression

$$w^* \in \arg\max_w P(D \mid w) = \arg\max_w \prod_{i=1}^{n} P(y_i \mid x_i, w)$$

$$= \arg\max_w \sum_{i=1}^{n} \log P(y_i \mid x_i, w)$$

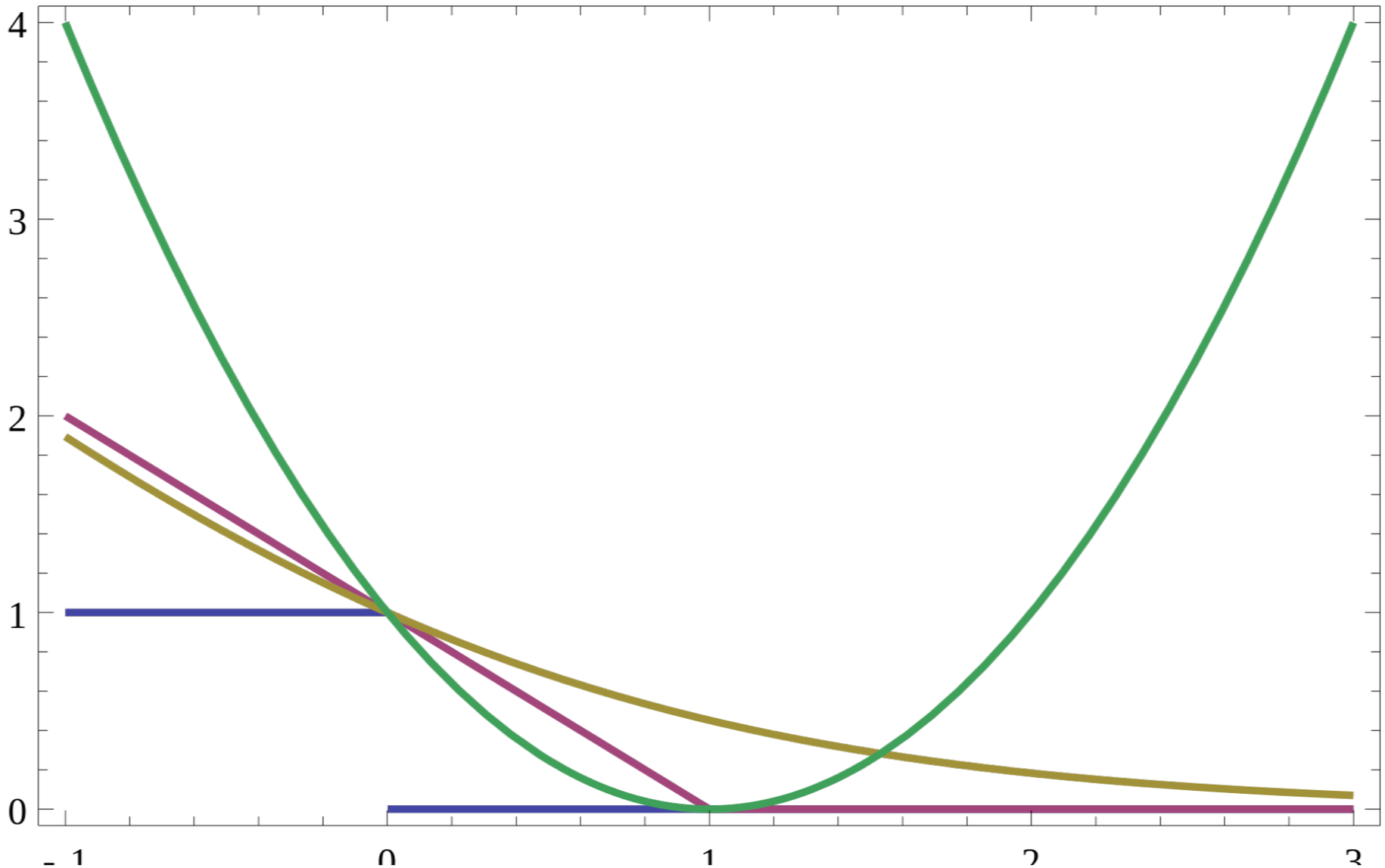
$$= \arg\min_w \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i w^\top x_i\right)\right)$$

- Negative log likelihood (=objective) function is given by

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

- The logistic loss is convex
  - optimization with (stochastic) gradient descent

Logistic loss (log loss)



Log loss is shown in brown

Gradient for logistic regression

- Loss for data point (x, y)

$$\ell(h_w(x), y) = \log\left(1 + \exp\left(-yw^\top x\right)\right)$$

Gradient for logistic regression

- Loss for data point (x, y)

$$\ell(h_w(x), y) = \log\left(1 + \exp\left(-yw^\top x\right)\right)$$

- Gradient

$$\begin{aligned}
\nabla_w \ell(h_w(x), y) &= \frac{1}{1 + \exp(-yw^\top x)} \cdot \exp\left(-yw^\top x\right) \cdot (-yx) \\
&= \frac{\exp\left(-yw^\top x\right)}{1 + \exp\left(-yw^\top x\right)} \cdot (-yx) \\
&= \frac{1}{1 + \exp\left(yw^\top x\right)} \cdot (-yx)
\end{aligned}$$

Optimization: logistic regression

Algorithm: SGD for logistic regression

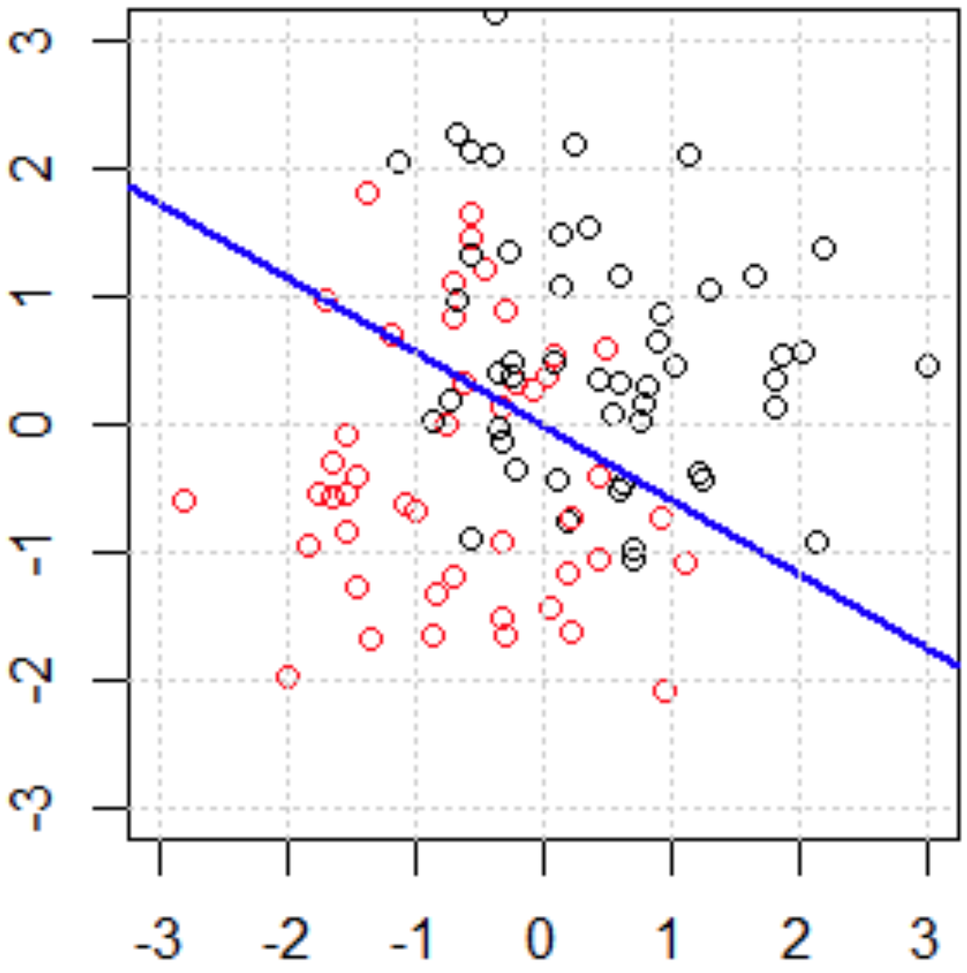
- Initialize w
- for $t = 1, 2, \ldots$
  - Pick data point (x, y) uniformly at random from data D
  - Compute probability of misclassification with current model

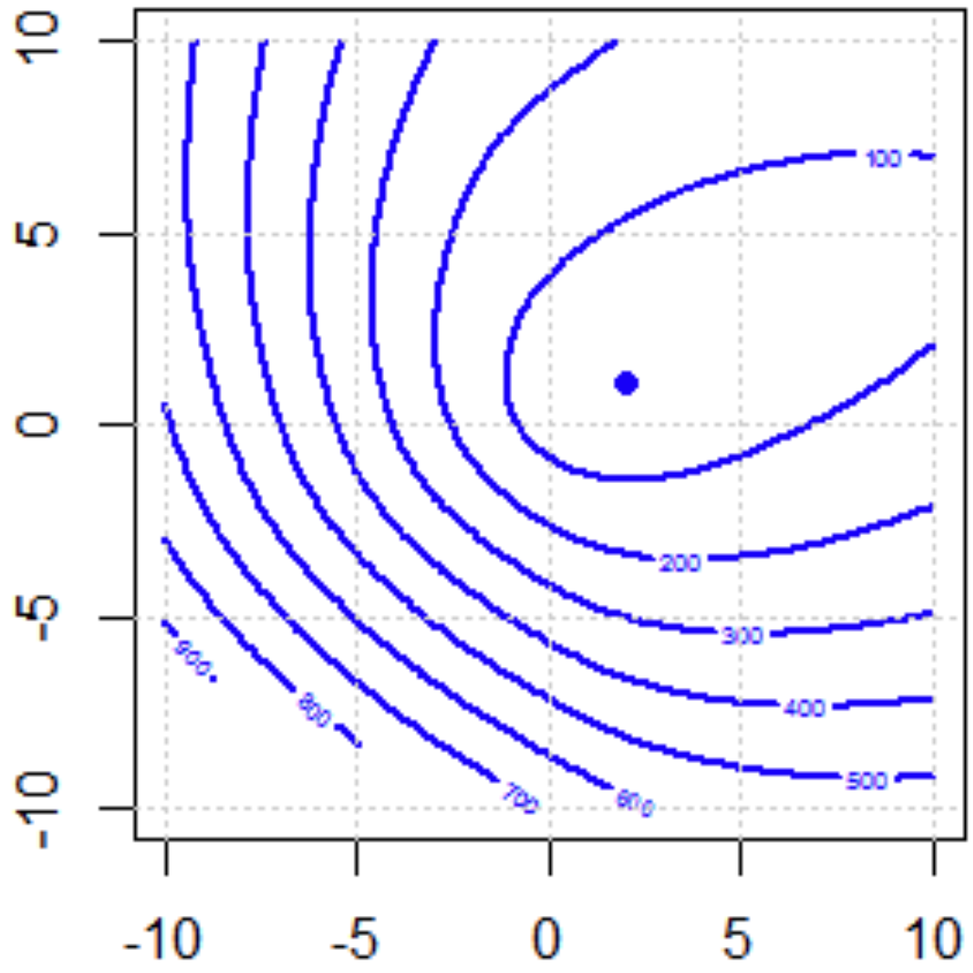
$$\hat{P}(Y = -y \mid w, x) = \frac{1}{1 + \exp(yw^\top x)}$$

- Take gradient step

$$w \leftarrow w + \eta_t y x \hat{P}(Y = -y \mid w, x)$$

Demo


3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3


10
5
0
-5
-10
100
200
300
400
500
600
700
800
900
-10
-5
0
5
10

Logistic regression and regularization

- Use regularizer to control model complexity
- Instead of solving MLE

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

estimate MAP/solve regularized problem

- L2 (Gaussian prior)

$$\min_w \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i w^\top x_i\right)\right) + \lambda \|w\|_2^2$$

- L1 (Laplace prior)

$$\min_w \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i w^\top x_i\right)\right) + \lambda \|w\|_1$$

Optimization: regularized logistic regression

Algorithm: SGD for L2-regularized logistic regression

- Initialize w
- for $t = 1, 2, \ldots$
  - Pick data point (x, y) uniformly at random from data D
  - Compute probability of misclassification with current model

$$\hat{P}(Y = -y \mid w, x) = \frac{1}{1 + \exp(yw^\top x)}$$

- Take gradient step

$$w \leftarrow w(1 - 2\lambda\eta_t) + \eta_t y x \hat{P}(Y = -y \mid w, x)$$

Regularized logistic regression

Learning

- Find optimal weights by minimizing logistic loss + regularizer

$$\hat{w} = \arg\min_w \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i w^\top x_i\right)\right) + \lambda \|w\|_2^2$$

$$= \arg\max_w P(w \mid x_1, \ldots, x_n, y_1, \ldots, y_n)$$

Classification

- Use conditional distribution

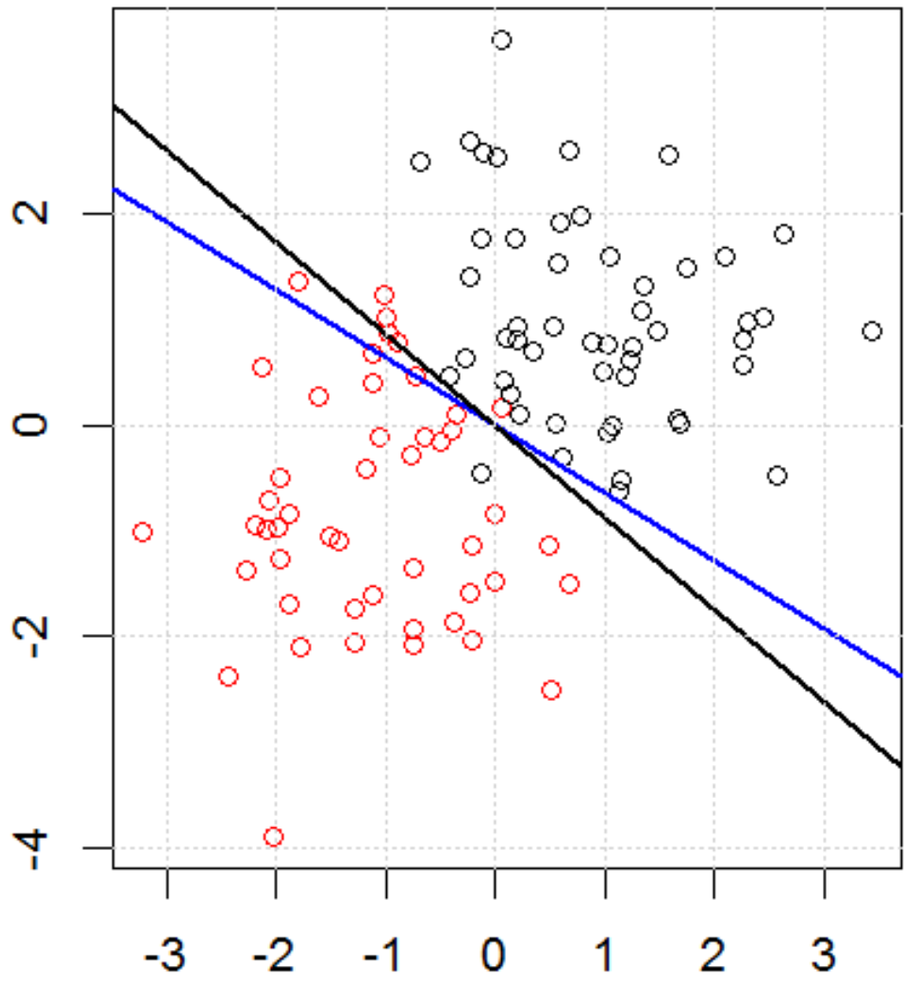
$$P(y \mid w, x) = \frac{1}{1 + \exp(-y w^\top x)}$$

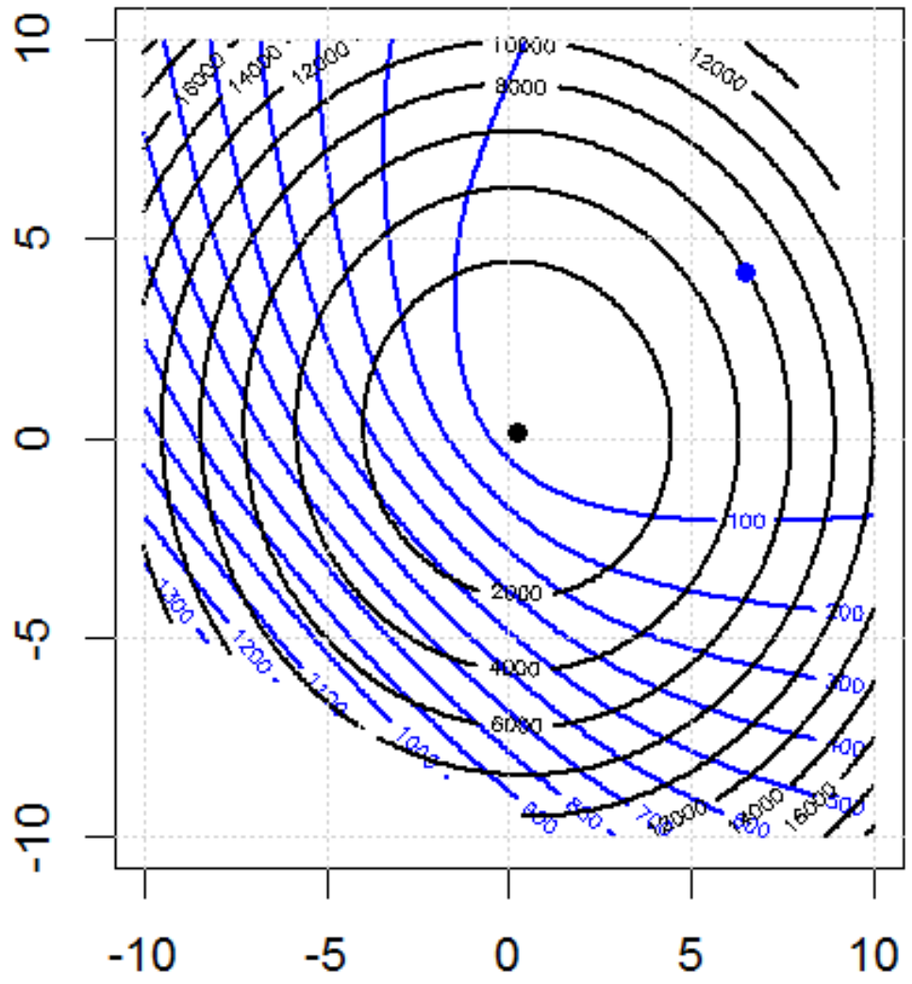
- Predict the more likely class label

$$\hat{y} = \arg\max_y P(y \mid x, \widehat{w})$$

Demo: LR on synthesized binary data set

- **Left**: synthesized dataset generated from two Gaussian; learned decision boundary (blue: LR without regularization; black: LR with L2 regularization)
- **Right**: the objective function contour. The blue and black points in right figure are optimal parameters for objective function.





demo code: `https://bit.ly/357CdWf`

Extension to multi-class logistic regression

- Maintain one weight vector per class and model

$$P(Y = i \mid x, w_1, \ldots, w_c) = \frac{\exp\left(w_i^\top x\right)}{\sum_{j=1}^{c} \exp(w_j^\top x)}$$

Extension to multi-class logistic regression

- Maintain one weight vector per class and model

$$P(Y = i \mid x, w_1, \ldots, w_c) = \frac{\exp\left(w_i^\top x\right)}{\sum_{j=1}^{c} \exp(w_j^\top x)}$$

- Not unique – can force uniqueness by setting $w_c = 0$ (this recovers logistic regression as special case)

Extension to multi-class logistic regression

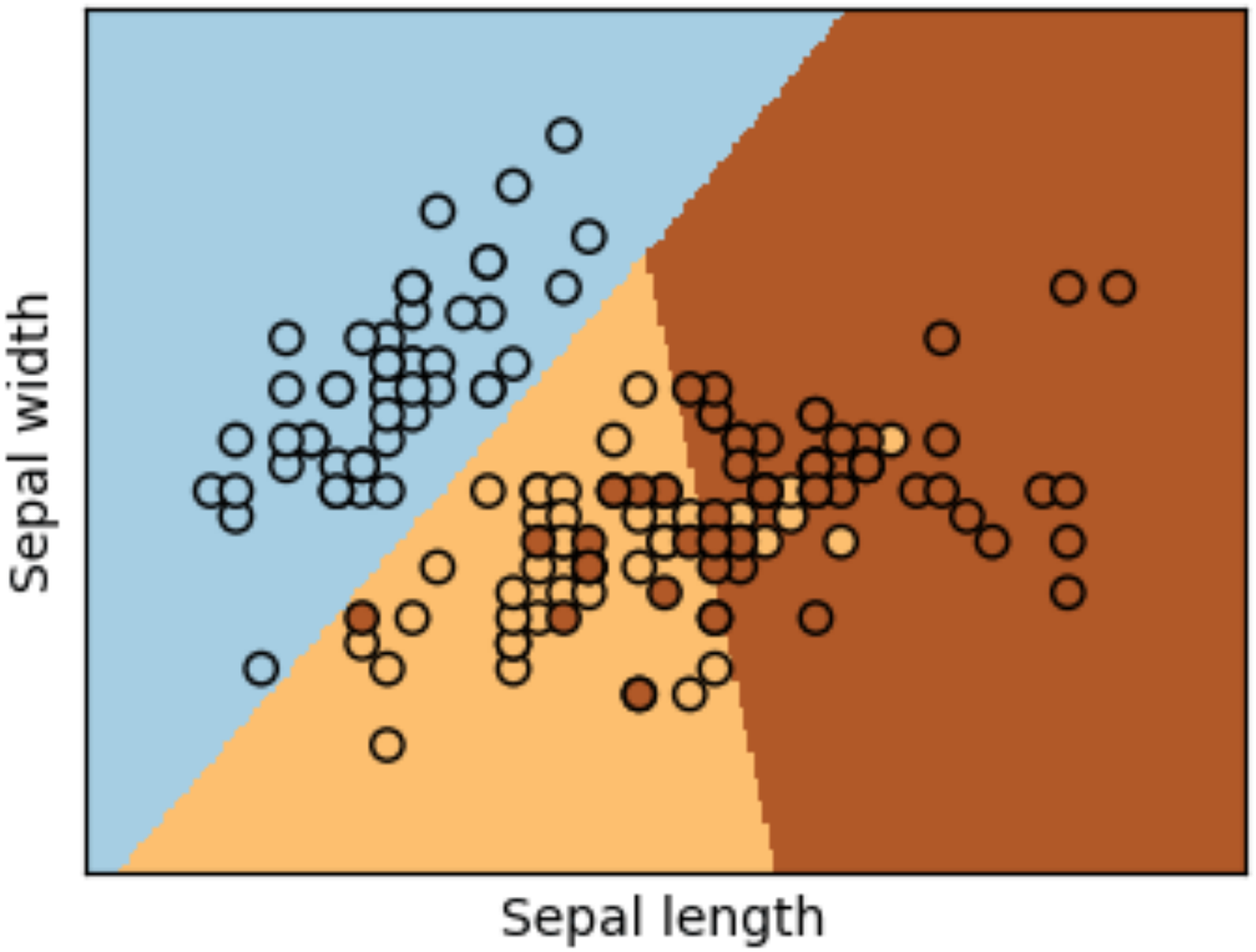
- Maintain one weight vector per class and model

$$P(Y = i \mid x, w_1, \ldots, w_c) = \frac{\exp\left(w_i^\top x\right)}{\sum_{j=1}^{c} \exp(w_j^\top x)}$$

- Not unique – can force uniqueness by setting $w_c = 0$ (this recovers logistic regression as special case)
- Corresponding loss function (cross-entropy loss)

$$\ell(y; x, w_1, \ldots, w_c) = -\log P(Y = y \mid x, w_1, \ldots, w_c)$$

Illustration: logistic regression 3-class classifier



Dataset (Iris Data Set) and demo code: `https://bit.ly/3bJ98CQ`

Outlook: Bayes classifier (next lecture)

"Optimization" based learning (MAP, MLE, ...)

$$\widehat{w} = \arg\max_{w} P(w \mid D)$$

$$P(y \mid x, \widehat{w})$$

- Ignores uncertainty in model, typically efficient to optimize

"Integration" based learning / Bayesian model averaging

$$P(y \mid x, D) = \int P(y \mid x, w) P(w \mid D)$$

- Quantifies model uncertainty, integration typically intractable

Logistic regression in the statistical learning framework

- Data
  - feature: linear features/hypotheses
  - label: binary/categorical
- Probabilistic model
  - logistic loss = Bernoulli likelihood
  - cross-entropy loss = Categorical likelihood
  - L2 norm = Gaussian prior
  - L1 norm = Laplace prior
- Method
  - Stochastic gradient descent
- Evaluation metric
  - log-likelihood on validation set
- Model selection
  - (k-fold) cross-validation