THE UNIVERSITY OF CHICAGO

STAT 37710 / CMSC 35400 / CAAM 37710
Machine Learning

Logistic Regression

Cong Ma

Statistical models for classification

- So far, we have focused on regression, e.g., with least-squared loss

$$\ell\big(y; h(\mathbf{x})\big) = (y - h(\mathbf{x}))^2$$

- Are there natural statistical models for classification?

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

- Can have {0,1}, {1,2, …, K}

Risk in classification

- In classification, risk is $R(h) = \mathbb{E}_{X,Y}[1\{Y \neq h(X)\}]$

$$\begin{aligned}
\mathbb{E}_{X,Y}[1\{Y \neq h(X)\}] &= \mathbb{E}_X \mathbb{E}_{Y|X}[1\{Y \neq h(X)\} \mid X = x] \\
&= \mathbb{E}_X \mathbb{P}_{Y|X}[Y \neq h(X) \mid X = x] \\
&= \mathbb{E}_X \left[\sum_{i=1}^{K} \mathbb{P}(Y = i \mid X = x) 1\{h(x) \neq i\} \right] \\
&= \mathbb{E}_X \left[\sum_{i:h(x) \neq i} \mathbb{P}(Y = i \mid X = x) \right] \\
&= \mathbb{E}_X \left[1 - \mathbb{P}(Y = h(X) \mid X = x) \right].
\end{aligned}$$

Bayes classifier

- Suppose (unrealistically) we knew P(**X**,Y).
  - Which *h* minimizes the risk?

$$
\begin{aligned}
h^*(\mathbf{x}) &= \arg\min_{\hat{y}} \mathbb{E}_Y [[Y \neq \hat{y} \mid \mathbf{X} = \mathbf{x}]] \\
&= \arg\min_{\hat{y}} \sum_{y=1}^{c} P(Y = y \mid \mathbf{X} = \mathbf{x})[y \neq \hat{y}] \\
&= \arg\min_{\hat{y}} \sum_{y \neq \hat{y}} P(Y = y \mid \mathbf{X} = \mathbf{x}) \\
&= \arg\max_{\hat{y}} P(Y = \hat{y} \mid \mathbf{X} = \mathbf{x})
\end{aligned}
$$

Bayes' optimal *classifier*

- Assuming the data is generated iid according to

$$(\mathbf{x}_i, y_i) \sim P(\mathbf{X}, Y)$$

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

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

- This hypothesis is called the Bayes' optimal predictor for the classification loss
- Thus, natural approach is again to estimate P(Y|X)

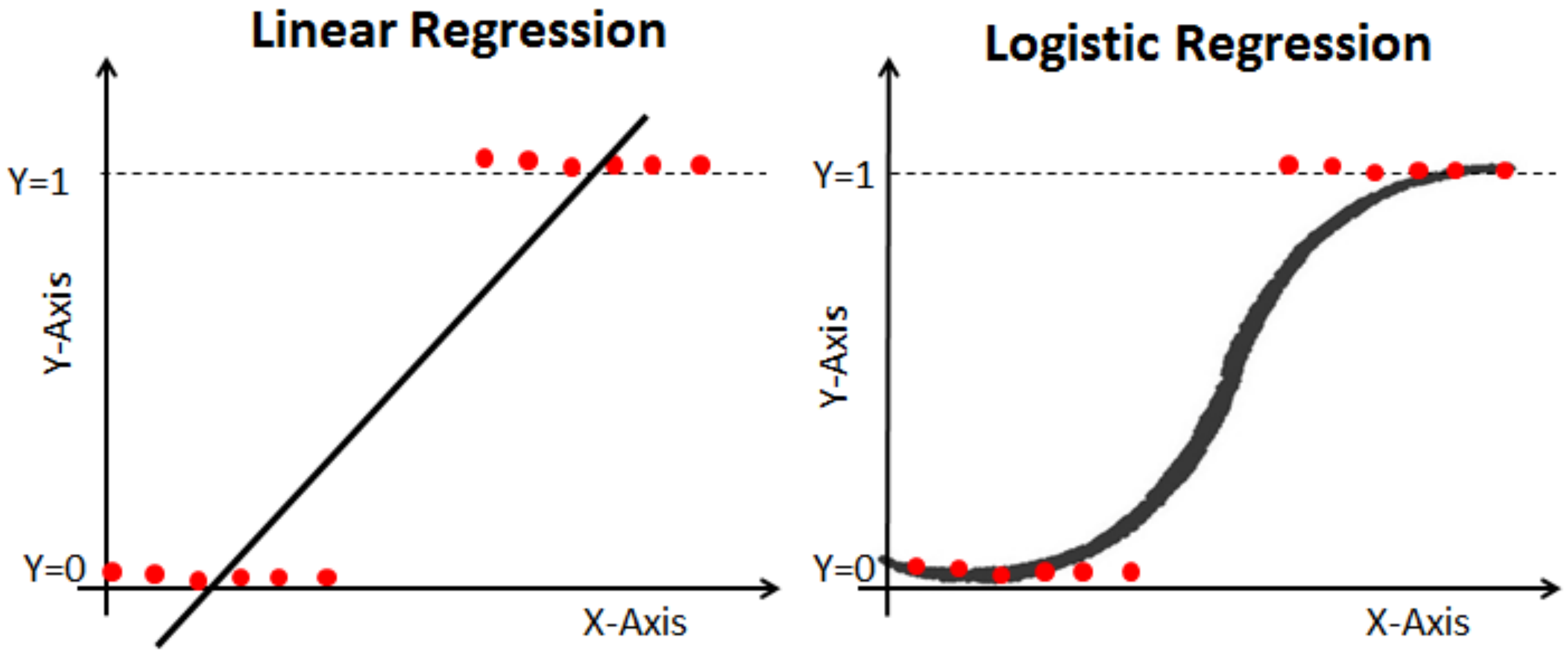
Natural estimator for P(Y|X)

- Fix some x in X
- Find out all x_i that are equal to x; suppose we have m such samples
- A natural estimator would be

- What's the problem of this?

We need a model for P(Y=1 | X = x)

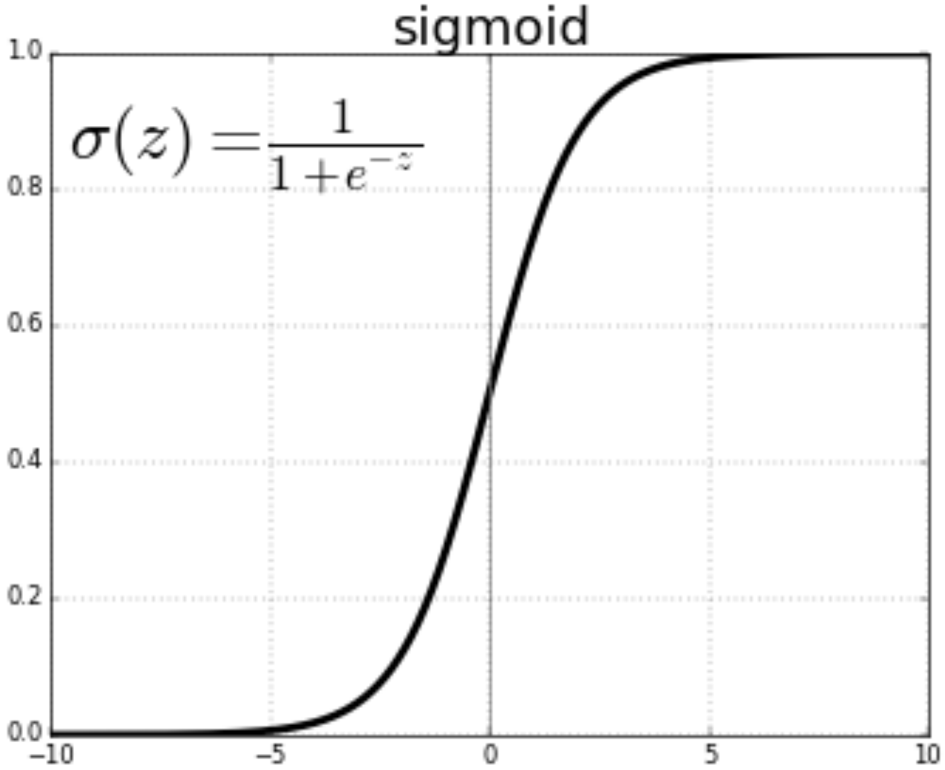
- What about a linear model?



Link function for logistic regression

- Link function

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



Logistic regression

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

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

Key observation

- Decision boundary is linear!
  - What's the decision boundary?
  - Why is it linear?

MLE for logistic regression

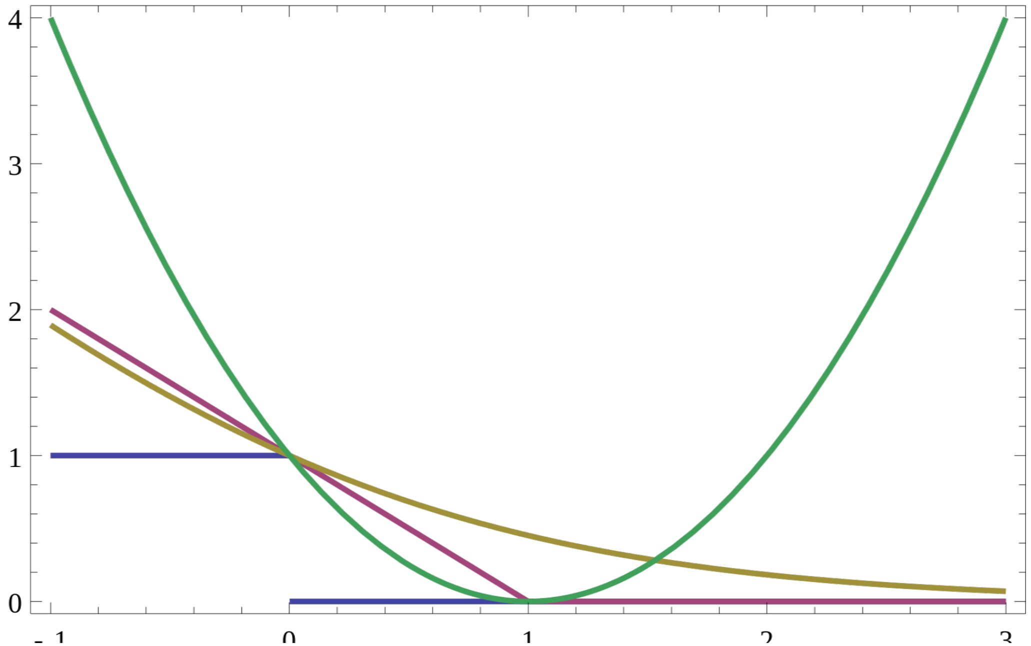
$$\mathbf{w}^* \in \arg\max_{\mathbf{w}} P(D \mid \mathbf{w}) = \arg\max_{\mathbf{w}} \prod_{i=1}^{n} P(y_i \mid \mathbf{x}_i, \mathbf{w})$$
$$= \arg\max_{\mathbf{w}} \sum_{i=1}^{n} \log P(y_i \mid \mathbf{x}_i, \mathbf{w})$$
$$= \arg\min_{\mathbf{w}} \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \mathbf{w}^\top \mathbf{x}_i\right)\right)$$

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

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

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

Logistic loss (log loss)



Gradient for logistic regression

- Loss for data point $(\mathbf{x}, y)$

$$\ell(h_{\mathbf{w}}(\mathbf{x}), y) = \log\left(1 + \exp\left(-y\mathbf{w}^\top \mathbf{x}\right)\right)$$

- Gradient $\nabla_{\mathbf{w}}\ell(h_{\mathbf{w}}(\mathbf{x}), y) = \frac{1}{1 + \exp(-y\mathbf{w}^\top \mathbf{x})} \cdot \exp\left(-y\mathbf{w}^\top \mathbf{x}\right) \cdot (-y\mathbf{x})$

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

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

Optimization: logistic regression

- Initialize **w**
- For t = 1, 2, ... do
  - Pick data point (x, y) uniformly at random from data D
  - Compute probability of misclassification with current model

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

  - Take gradient step $\mathbf{w} \leftarrow \mathbf{w} + \eta_t \cdot y\mathbf{x} \cdot \hat{P}(Y = -y \mid \mathbf{w}, \mathbf{x})$

Logistic regression and regularization

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

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \mathbf{w}^\top \mathbf{x}_i\right)\right)$$

- Estimate MAP/solve regularized problem
  - L2 (Gaussian prior)

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

  - L1 (Laplace prior)

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

Optimization: regularized logistic regression

- Initialize **w**
- For t = 1, 2, … do
  - Pick data point (x, y) uniformly at random from data D
  - Compute probability of misclassification with current model

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

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

Regularized logistic regression

- Learning
  - Find optimal weights by minimizing logistic loss + regularizer

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \mathbf{w}^\top \mathbf{x}_i\right)\right) + \lambda \|\mathbf{w}\|_2^2$$
$$= \arg\max_{\mathbf{w}} P(\mathbf{w} \mid \mathbf{x}_1, \ldots, \mathbf{x}_n, y_1, \ldots, y_n)$$

- Classification
  - Use conditional distribution $P(Y = y \mid \mathbf{w}, x) = \dfrac{1}{1 + \exp(-y\mathbf{w}^\top \mathbf{x})}$
  - Predict the more likely class label $\hat{y} = \arg\max_{y} P(y \mid \mathbf{x}, \hat{\mathbf{w}})$

Extension to multi-class logistic regression

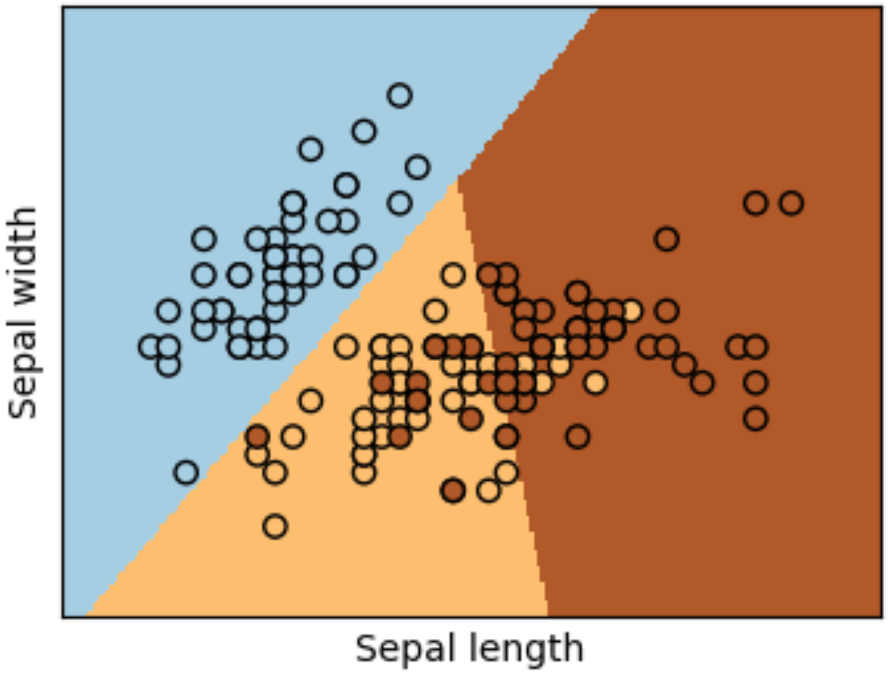
- Maintain one weight vector per class and model

$$P(Y = i \mid \mathbf{x}, \mathbf{w}_1, \ldots, \mathbf{w}_c) = \frac{\exp\left(\mathbf{w}_i^\top x\right)}{\sum_{j=1}^{c} \exp(\mathbf{w}_j^\top \mathbf{x})}$$

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

$$\ell(y; \mathbf{x}, \mathbf{w}_1, \ldots, \mathbf{w}_c) = -\log P(Y = y \mid \mathbf{x}, \mathbf{w}_1, \ldots, \mathbf{w}_c)$$

Illustration: logistic regression 3-class classifier



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

Summary

- Logistic regression is a supervised machine learning classifier that extracts real-valued features from the input, multiplies each by a weight, sums them, and passes the sum through a **sigmoid** function to generate a probability. A threshold is used to make a decision.
- Logistic regression can be used with two classes (e.g., positive and negative sentiment) or with multiple classes (**multinomial logistic regression**, for example for n-ary text classification, part-of-speech labeling, etc.).
- Multinomial logistic regression uses the **softmax** function to compute probabilities.
- The weights (vector w and bias b) are learned from a labeled training set via a loss function, such as the **cross-entropy loss**, that must be minimized.
- Minimizing this loss function is a **convex optimization** problem, and iterative algorithms like **gradient descent** are used to find the optimal weights.
- **Regularization** is used to avoid overfitting.
- Logistic regression is also one of the most useful analytic tools, because of its ability to transparently study the importance of individual features.

References & acknowledgement

- C. Bishop (2006). “Pattern Recognition and Machine Learning”
  - Ch 4.3.2, “Logistic regression”
  - Ch 4.3.4, “Multiclass logistic regression”

- K. Murphy (2021). “Probabilistic Machine Learning: An Introduction”
  - 10.2 “Binary logistic regression”
  - 10.3 “Multinomial logistic regression”

- A. Krause, “Introduction to Machine Learning” (ETH Zurich, 2019)