THE UNIVERSITY OF CHICAGO

Overview of classical ML: classification methods and decision trees

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' 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?

THE UNIVERSITY OF CHICAGO

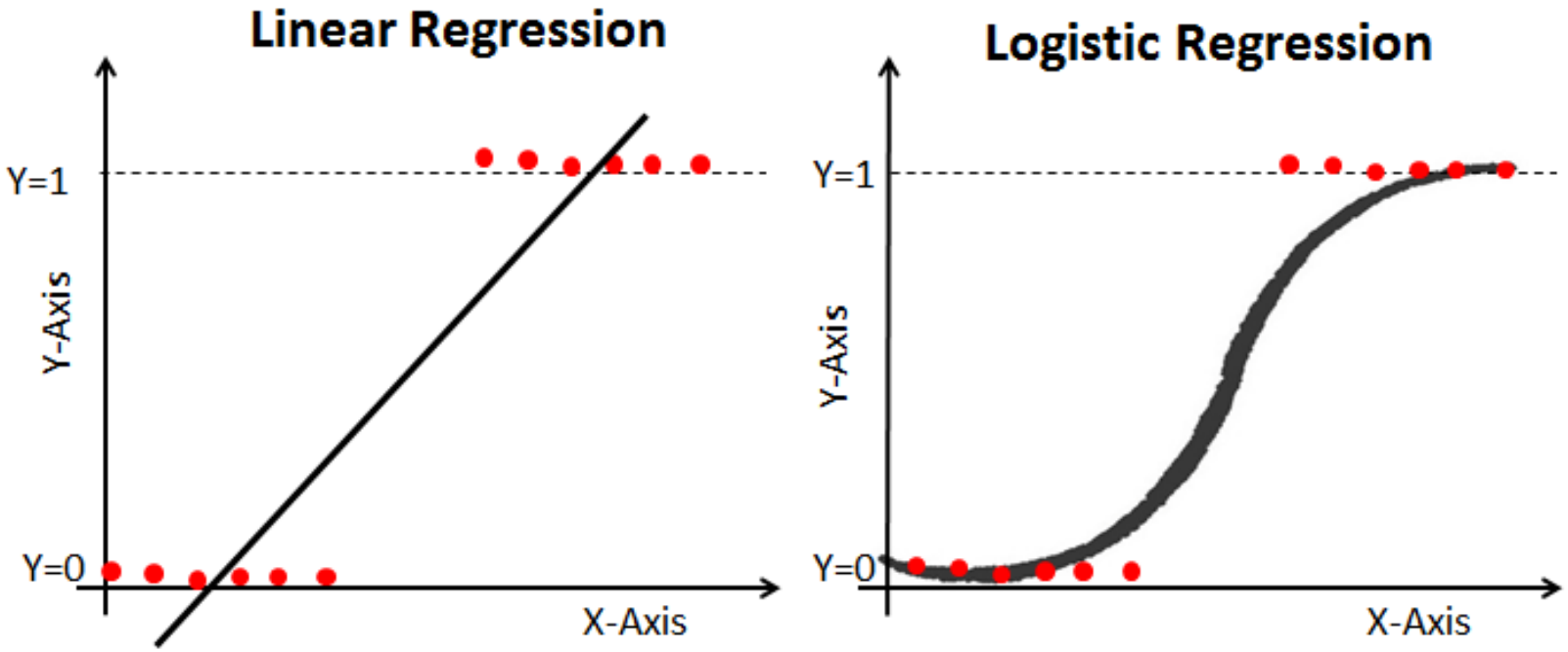
Overview of classical ML: classification methods and decision trees

Logistic regression

Cong Ma

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

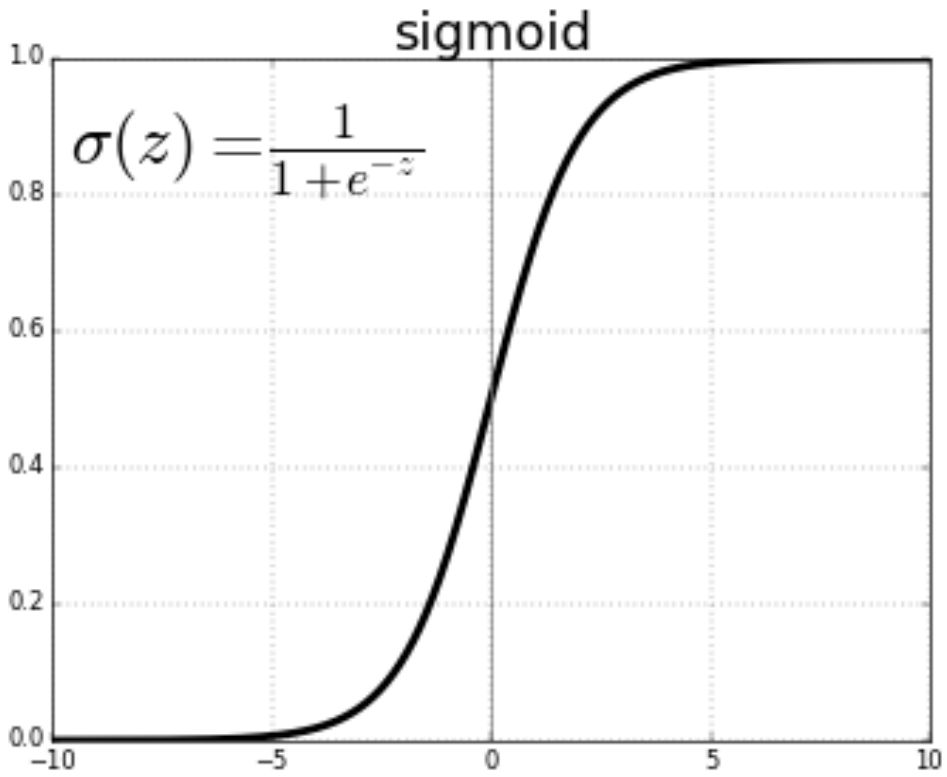
- What about a linear model?



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?

$$\mathbb{P}(Y=0|w,X)=\frac{1}{1+\exp(w_0+\sum_k w_k X_k)}$$

$$\mathbb{P}(Y=1|w,X)=1-\mathbb{P}(Y=0|w,X)=\frac{\exp(w_0+\sum_k w_k X_k)}{1+\exp(w_0+\sum_k w_k X_k)}$$

$$\frac{\mathbb{P}(Y=1|w,X)}{\mathbb{P}(Y=0|w,X)}=\exp(w_0+\sum_k w_k X_k)$$

Linear Decision Rule!

$$\log\frac{\mathbb{P}(Y=1|w,X)}{\mathbb{P}(Y=0|w,X)}=w_0+\sum_k w_k X_k$$

How to fit logistic regression

- **Have a bunch of iid data:** $\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$

$$P(Y = -1|x, w) = \frac{1}{1 + \exp(w^T x)}$$

$$P(Y = 1|x, w) = \frac{\exp(w^T x)}{1 + \exp(w^T x)}$$

- **This is equivalent to:**

$$P(Y = y|x, w) = \frac{1}{1 + \exp(-y\, w^T x)}$$

MLE for logistic regression

$$\mathbf{w}^* \in \arg\max_{\mathbf{w}} P(D \mid \mathbf{w}) = \arg\max_{\mathbf{w}} \prod_{i=1}^{n} P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

$$= \arg\max_{\mathbf{w}} \sum_{i=1}^{n} \log P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

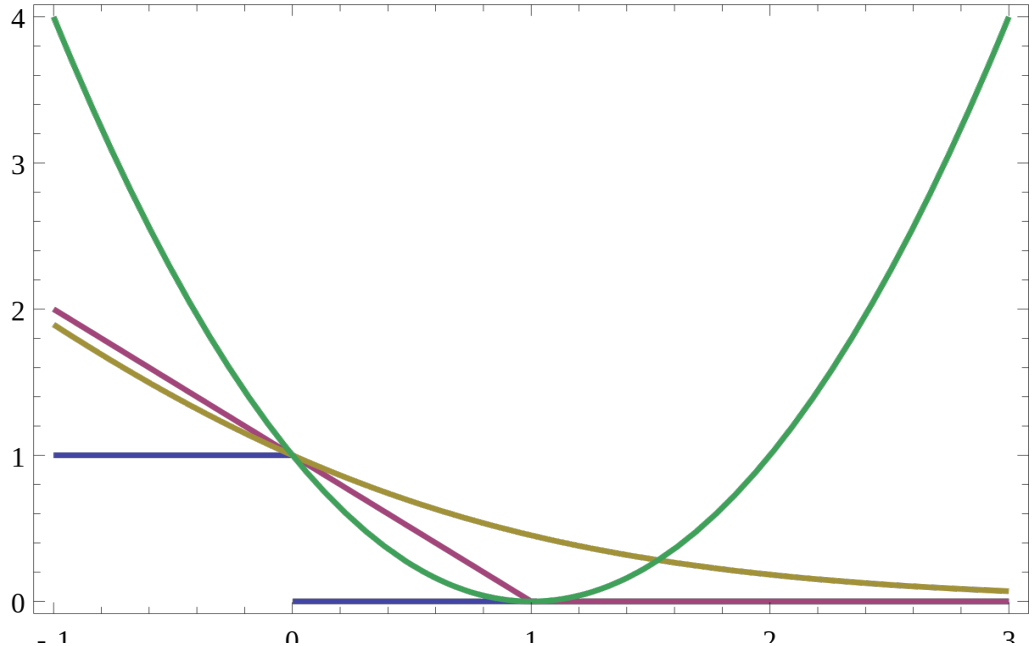
$$= \arg\min_{\mathbf{w}} \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \mathbf{w}^\top \mathbf{x}_i\right)\right)$$

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

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

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

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)

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

Extension to multi-class logistic regression

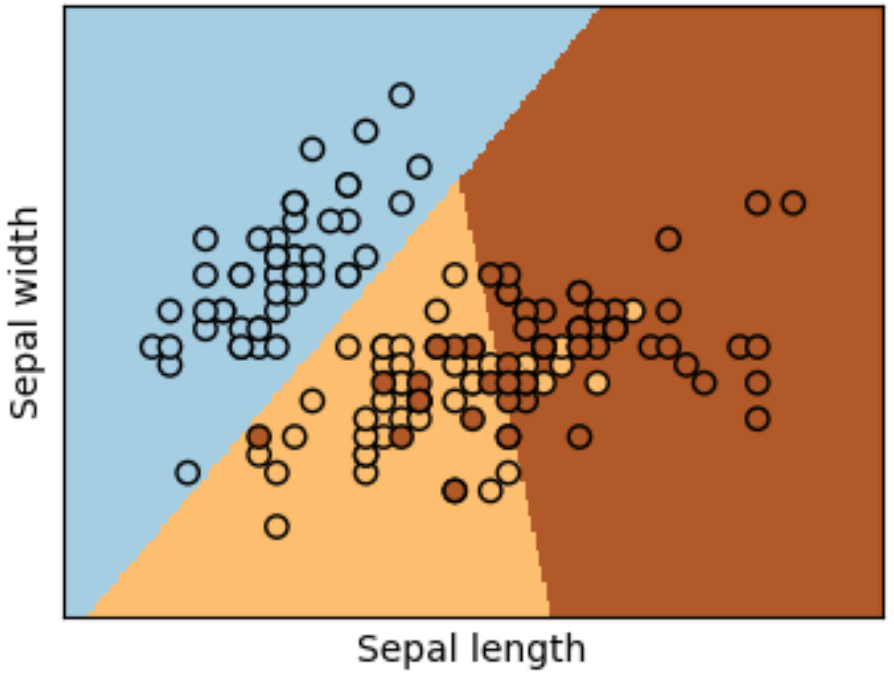
- Maintain one weight vector per class and model

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

- Not unique – can force uniqueness by setting w_1 = 0
  - 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

THE UNIVERSITY OF CHICAGO

STAT 37710 / CMSC 35400 / CAAM 37710 Machine Learning

Generative Models for Classification

Cong Ma

Discriminative modeling

- Discriminative models aim to estimate conditional distribution

$$P(y \mid \mathbf{x})$$

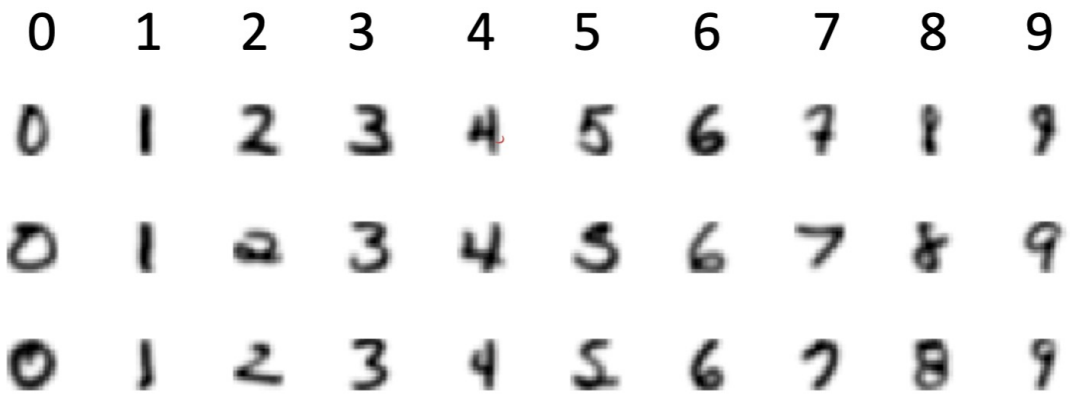
- Generative models aim to estimate joint distribution

$$P(y, \mathbf{x})$$

- Can derive conditional from joint distribution, but not vice versa.

Typical approaches to generative modeling

- Estimate prior on labels P(y)
- Estimate conditional distribution P(**x** | y) for each class y
- Obtain predictive distribution using Bayes' rule: P(y | **x**) = P(y) P(**x** | y) / Z



Example: hand-written digits

Gaussian Bayes classifiers (GBC)

- Model class label as generated from categorical variable

$$P(Y = y) = p_y, \;\; y \in \mathcal{Y} = \{1, \ldots, c\}$$

- Model features as multivariate Gaussians

$$P(\mathbf{x} \mid y) = \mathcal{N}(\mathbf{x}; \mu_y, \Sigma_y)$$

- How do we estimate the parameters?

MLE for GBC

- Given data $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
- MLE for class label distribution

$$\hat{P}(Y = y) = \hat{p}_y = \frac{\text{Count}(Y = y)}{n}$$

- MLE for feature distribution:

$$\hat{P}(\mathbf{x} \mid y) = \mathcal{N}(\mathbf{x}; \hat{\mu}_y, \hat{\Sigma}_y^2)$$

$$\hat{\mu}_y = \frac{1}{\text{Count}(Y = y)} \sum_{j:y_j=y} \mathbf{x}_j, \quad \hat{\Sigma} y = \frac{1}{\text{Count}(Y = y)} \sum_{j:y_j=y} (\mathbf{x}_j - \hat{\mu}_y)(\mathbf{x}_j - \hat{\mu}_y)^\top$$

Discriminant functions for GBC

- Given $P(Y = +1) = p_+$; $P(\mathbf{x} \mid y) = \mathcal{N}(\mathbf{x}; \mu_y, \Sigma_y)$
- GBC is given by

$$f(\mathbf{x}) = \log \frac{P(Y = +1 \mid \mathbf{x})}{P(Y = -1 \mid \mathbf{x})}$$

$$= \log \frac{p_+}{1 - p_+} + \frac{1}{2} \log \frac{\left|\hat{\Sigma}_-\right|}{\left|\hat{\Sigma}_+\right|} +$$

$$\frac{1}{2}\left[\left((\mathbf{x} - \hat{\mu}_-)^\top \hat{\Sigma}_-^{-1} (\mathbf{x} - \hat{\mu}_-)\right) - \left((\mathbf{x} - \hat{\mu}_+)^\top \hat{\Sigma}_+^{-1} (\mathbf{x} - \hat{\mu}_+)\right)\right]$$

Fisher's linear discriminant analysis (LDA), c = 2

- Suppose we fix $p_+ = 0.5$
- Further, assume covariances are equal: $\Sigma_+ = \Sigma_- = \Sigma$
- Then the discriminant function for GBC could be simplified as

$$
\begin{aligned}
f(\mathbf{x}) &= \log \frac{p_+}{1-p_+} + \frac{1}{2}\left[\log \frac{\left|\hat{\Sigma}_-\right|}{\left|\hat{\Sigma}_+\right|} + \left((\mathbf{x}-\hat{\mu}_-)^\top \hat{\Sigma}_-^{-1}(\mathbf{x}-\hat{\mu}_-)\right) - \left((\mathbf{x}-\hat{\mu}_+)^\top \hat{\Sigma}_+^{-1}(\mathbf{x}-\hat{\mu}_+)\right)\right] \\
&= \frac{1}{2}\left[\left((\mathbf{x}-\hat{\mu}_-)^\top \hat{\Sigma}^{-1}(\mathbf{x}-\hat{\mu}_-)\right) - \left((\mathbf{x}-\hat{\mu}_+)^\top \hat{\Sigma}^{-1}(\mathbf{x}-\hat{\mu}_+)\right)\right] \\
&=
\end{aligned}
$$

Fisher’s LDA

- Assuming
  - binary classification $\mathcal{Y} = \{-1, +1\}$
  - equal class probabilities $p_+ = 0.5$
  - equal covariances $\Sigma_+ = \Sigma_- = \Sigma$

- Fisher's LDA predicts

$$y = \text{sign}(f(\mathbf{x})) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$$

where $\mathbf{w} = \hat{\Sigma}^{-1}(\hat{\mu}_+ - \hat{\mu}_-)$ and $b = \frac{1}{2}\left(\hat{\mu}_-^\top \hat{\Sigma}^{-1} \hat{\mu}_- - \hat{\mu}_+^\top \hat{\Sigma}^{-1} \hat{\mu}_+\right)$

LDA vs logistic regression

- Fisher's LDA uses the discriminant function

$$f(\mathbf{x}) = \log \frac{P(Y=+1 \mid \mathbf{x})}{P(Y=-1 \mid \mathbf{x})} := \mathbf{w}^\top \mathbf{x} + b$$

$$\Leftrightarrow \quad P(Y=+1 \mid \mathbf{x}) = \frac{1}{1+\exp\left(-f(\mathbf{x})\right)} = \sigma(f(\mathbf{x}))$$

- Therefore, the class probability of LDA is

$$P(Y=+1 \mid \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b)$$

This is of the same form as logistic regression.

Fisher's LDA vs logistic regression

- Fisher's LDA
  - Generative model, i.e., models P(X,Y)
  - Assumes normality of X
  - not very robust against violation of this assumption

- Logistic regression
  - Discriminative model, i.e., models P(Y | X) only
  - Makes no assumptions on X
  - More robust