Logistic Regression

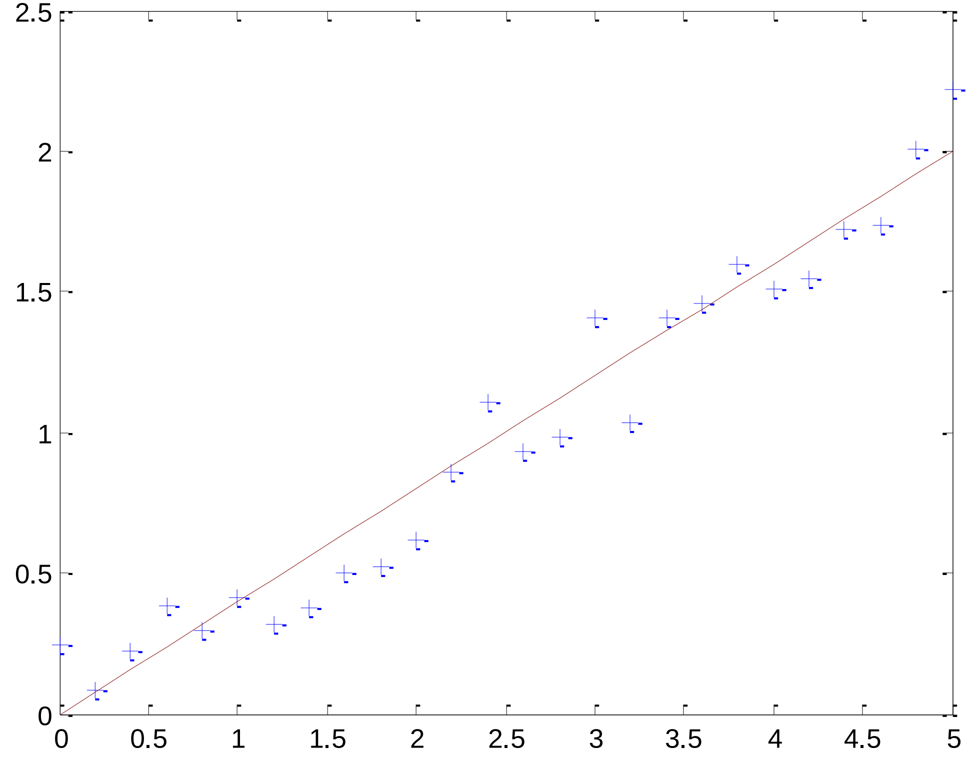
Mausam

Based on slides of Rong Jin, Tom Mitchell, Yi Zhang

Linear Regression

- y is continuous

$$y = \vec{x} \cdot \vec{w} + c$$



Logistic Regression Model

- The log-ratio of positive class to negative class

$$\log \frac{p(y=1|\vec{x})}{p(y=-1|\vec{x})} = \vec{x}\cdot\vec{w}+c \quad \Rightarrow \quad \frac{p(y=1|\vec{x})}{p(y=-1|\vec{x})} = \exp(\vec{x}\cdot\vec{w}+c)$$

$$p(y=1|\vec{x}) + p(y=-1|\vec{x}) = 1$$

Logistic Regression Model

- The log-ratio of positive class to negative class

$$\log \frac{p(y=1|\vec{x})}{p(y=-1|\vec{x})} = \vec{x}\cdot\vec{w}+c \quad \Rightarrow \quad \frac{p(y=1|\vec{x})}{p(y=-1|\vec{x})} = \exp(\vec{x}\cdot\vec{w}+c)$$

$$p(y=1|\vec{x}) + p(y=-1|\vec{x}) = 1$$

- Results

$$\left.\begin{array}{l} p(y=-1|\vec{x}) = \dfrac{1}{1+\exp(\vec{x}\cdot\vec{w}+c)} \\ p(y=1|\vec{x}) = \dfrac{1}{1+\exp(-\vec{x}\cdot\vec{w}-c)} \end{array}\right\} \Rightarrow p(y|\vec{x}) = \frac{1}{1+\exp\left[-y(\vec{x}\cdot\vec{w}+c)\right]}$$

Logistic Regression Model

- Assume the inputs and outputs are related in the log linear function

$$p(y \mid \vec{x};\theta) = \frac{1}{1+\exp\left[-y(\vec{x}\cdot\vec{w}+c)\right]}$$

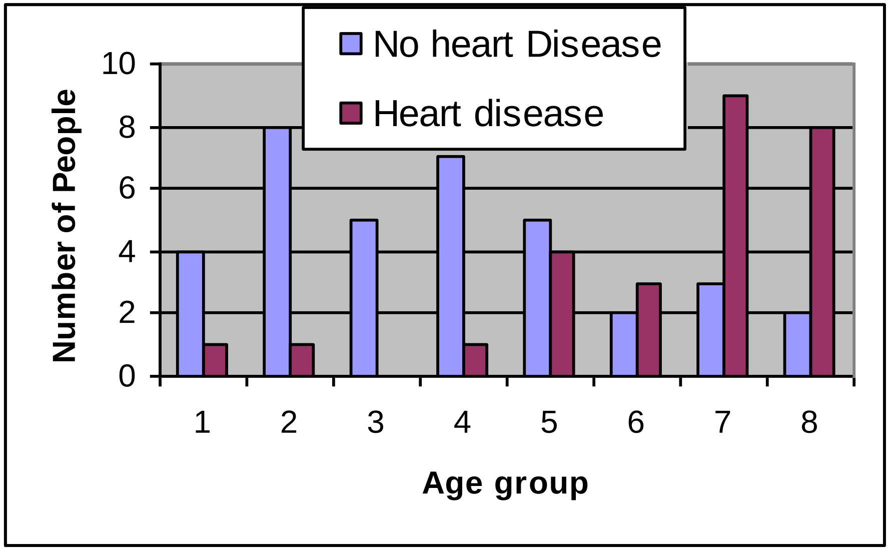
$$\theta = \{w_1, w_2, ..., w_d, c\}$$

- Estimate weights: MLE approach $\{w_1, w_2, ..., w_d, c\}$

$$\left\{\vec{w}, c\right\}^* = \max_{\vec{w},c} l(D_{train}) = \max_{\vec{w},c} \sum_{i=1}^{n} \log p(y_i \mid \vec{x}_i;\theta)$$

$$= \max_{\vec{w},c} \sum_{i=1}^{n} \log \frac{1}{1+\exp(-y\left[\vec{x}\cdot\vec{w}+c\right])}$$

Example 1: Heart Disease



1: 25-29

2: 30-34

3: 35-39

4: 40-44

5: 45-49

6: 50-54

7: 55-59

8: 60-64

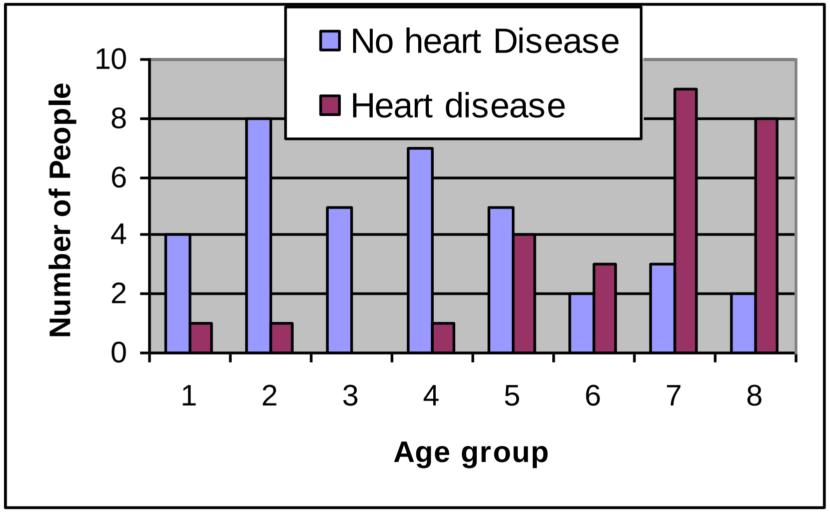
- Input feature *x*: age group id
- output *y*: having heart disease or not
  - +1: having heart disease
  - -1: no heart disease

Example 1: Heart Disease

- Logistic regression model

$$p(y \mid x) = \frac{1}{1 + \exp\left[-y(xw + c)\right]}$$

$$\theta = \{w, c\}$$



- Learning *w* and *c*: MLE approach

$$l(D_{train}) = \sum_{i=1}^{8} \{n_i(+) \log p(+ \mid i) + n_i(-) \log p(- \mid i)\}$$

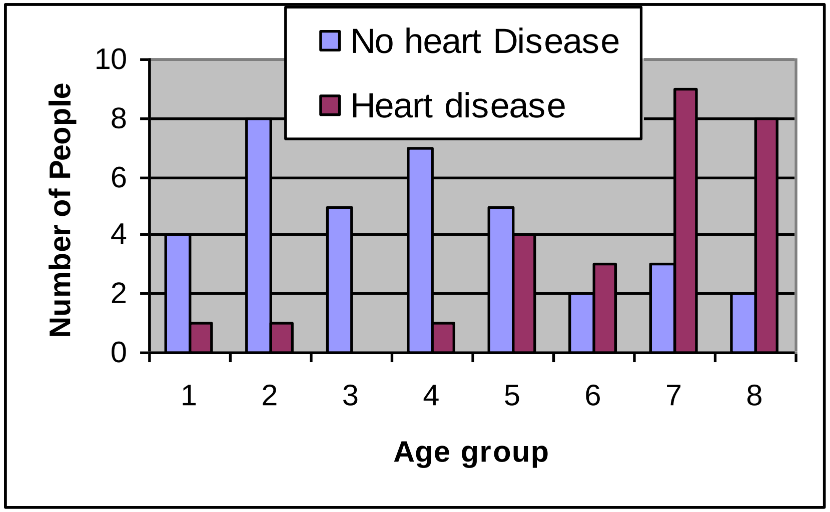
$$= \sum_{i=1}^{8} \left\{ n_i(+) \log \frac{1}{1 + \exp\left[-iw - c\right]} + n_i(-) \log \frac{1}{1 + \exp\left[iw + c\right]} \right\}$$

- Numerical optimization: w = 0.58, c = -3.34

Example 1: Heart Disease

$$p(+|x;\theta) = \frac{1}{1+\exp[-xw-c]}; p(-|x;\theta) = \frac{1}{1+\exp[xw+c]}$$

- W = 0.58
  - An old person is more likely to have heart disease
- C = -3.34
  - xw+c < 0 → p(+|x) < p(-|x)
  - xw+c > 0 → p(+|x) > p(-|x)
  - xw+c = 0 → decision boundary
    - x* = 5.78 → 53 year old



Example: Text Classification

- Learn to classify text into predefined categories
- Input $\vec{x}$: a document
  - Represented by a vector of words
  - Example: {(president, 10), (bush, 2), (election, 5), …}
- Output *y*: if the document is politics or not
  - +1 for political document, -1 for not political document
- Training data:

$$\underbrace{\left\{\vec{d}_1^{\,+},\vec{d}_2^{\,+},\ldots,\vec{d}_{n_+}^{\,+}\right\};\left\{\vec{d}_1^{\,-},\vec{d}_2^{\,-},\ldots,\vec{d}_{n-}^{\,-}\right\}}_{N=n_+ + n_-}$$

$$\vec{d}_i^{(\pm)} = \left\{\left(word_1, t_{i,1}^{\pm}\right),\left(word_2, t_{i,2}^{\pm}\right),\ldots,\left(word_n, t_{i,n}^{\pm}\right)\right\}$$

Example 2: Text Classification

- Logistic regression model
  - Every term t_i is assigned with a weight w_i $d = \{(word_1, t_1), (word_2, t_2), ..., (word_n, t_n)\}$

$$p(y \mid d; \theta) = \frac{1}{1 + \exp\left[-y(\sum_i w_i \cdot t_i + c)\right]}$$

$$\theta = \{w_1, w_2, ..., w_n, c\}$$

Example 2: Text Classification

- Logistic regression model
  - Every term t_i is assigned with a weight w_i $d = \{(word_1, t_1), (word_2, t_2), ..., (word_n, t_n)\}$

$$p(y \mid d; \theta) = \frac{1}{1 + \exp\left[-y(\sum_i w_i \cdot t_i + c)\right]}$$

$$\theta = \{w_1, w_2, ..., w_n, c\}$$

- Learning parameters: MLE approach

$$l(D_{train}) = \sum_{i=1}^{n_+} \log p(+ \mid d_i^+) + \sum_{i=1}^{n_-} \log p(- \mid d_i^-)$$

$$= \sum_{i=1}^{n_+} \log \frac{1}{1 + \exp\left[-\sum_j w_j \cdot t_{i,j} - c\right]} + \sum_{i=1}^{n_-} \log \frac{1}{1 + \exp\left[\sum_j w_j \cdot t_{i,j} + c\right]}$$

- Need numerical solutions

Example 2: Text Classification

- Weight w_i
  - $w_i > 0$: term t_i is a positive evidence
  - $w_i < 0$: term t_i is a negative evidence
  - $w_i = 0$: term t_i is irrelevant to the category of documents
  - The larger the $| w_i |$, the more important t_i term is determining whether the document is interesting.

Example 2: Text Classification

- Weight w_i
  - $w_i > 0$: term t_i is a positive evidence
  - $w_i < 0$: term t_i is a negative evidence
  - $w_i = 0$: term t_i is irrelevant to the category of documents
  - The larger the $| w_i |$, the more important t_i term is determining whether the document is interesting.
- Threshold c

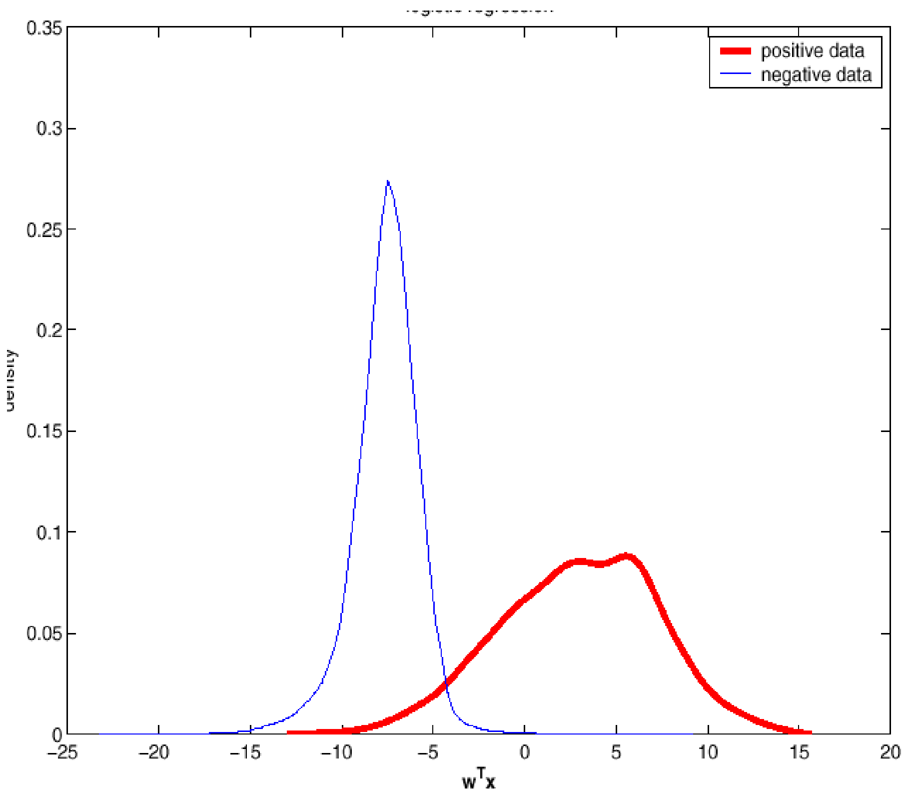
$$\sum_i w_i \cdot t_i + c > 0: \text{ more likely to be a political document}$$

$$\sum_i w_i \cdot t_i + c < 0: \text{ more likely to be a non-political document}$$

$$\sum_i w_i \cdot t_i + c = 0: \text{ decision boundary}$$

Example 2: Text Classification

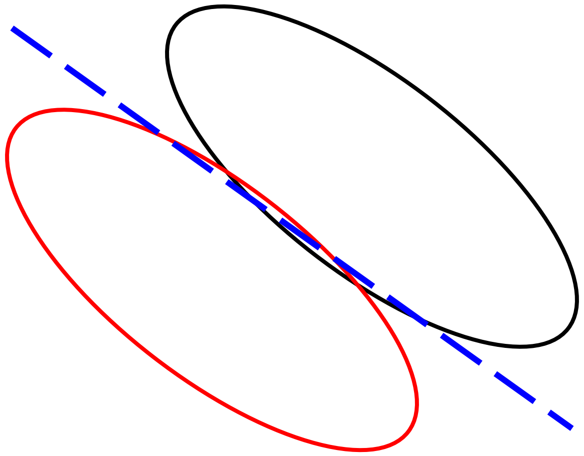
- Dataset: Reuter-21578
- Classification accuracy
    - Naïve Bayes: 77%
    - Logistic regression: 88%



Discriminative Model

- Logistic regression model is a discriminative model
  - Models the conditional probability p(y|x), i.e., the decision boundary

- Generative model
  - Models p(x|y), i.e., input patterns of different classes



Generative vs. Discriminative Classifiers

- Discriminative classifiers
  - Assume some functional form for P(Y|X)
  - Estimate parameters of P(Y|X) directly from training data
- Generative classifiers
  - Assume some functional form for P(X|Y), P(X)
  - Estimate parameters of P(X|Y), P(X) directly from training data
  - Use Bayes rule to calculate $P(Y|X = x_i)$

Asymptotic Difference

- Notation: let $\epsilon(h_{A,m})$ denote error of hypothesis learned via algorithm A, from *m* examples
- If assumed model correct (e.g., naïve Bayes model), and finite number of parameters, then

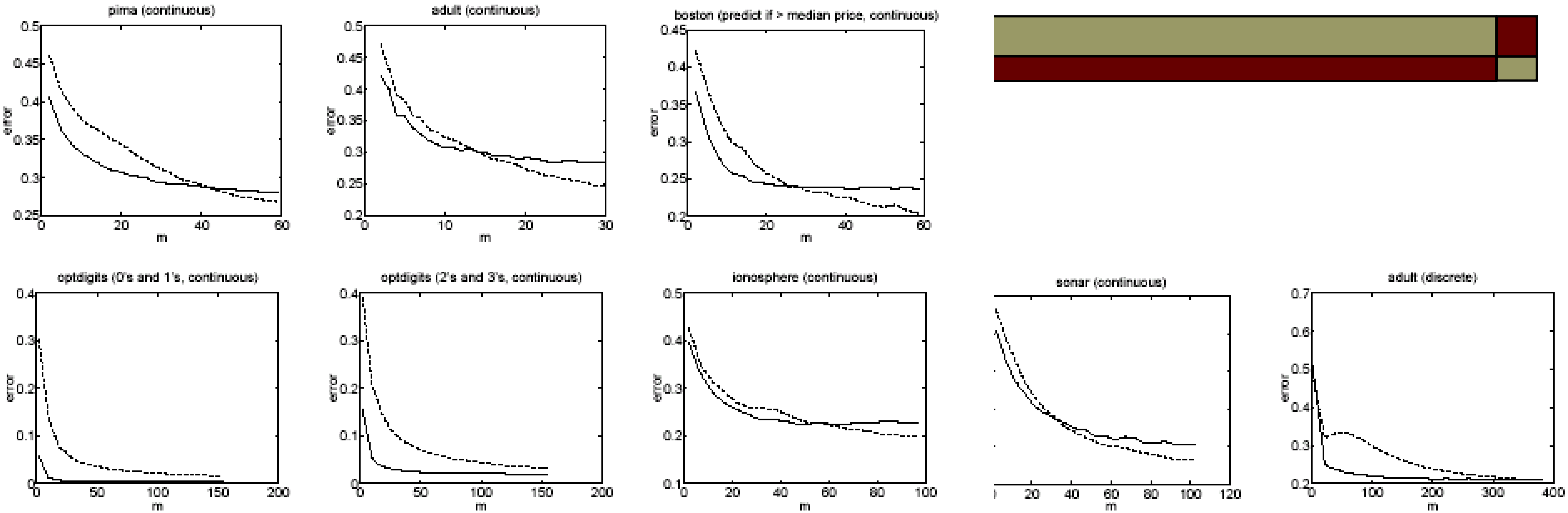
$$\epsilon(h_{Dis,\infty}) = \epsilon(h_{Gen,\infty})$$

- If assumed model incorrect

$$\epsilon(h_{Dis,\infty}) \leq \epsilon(h_{Gen,\infty})$$

- Note assumed discriminative model can be correct even when generative model incorrect, but not vice versa

Some experiments from UCI data sets



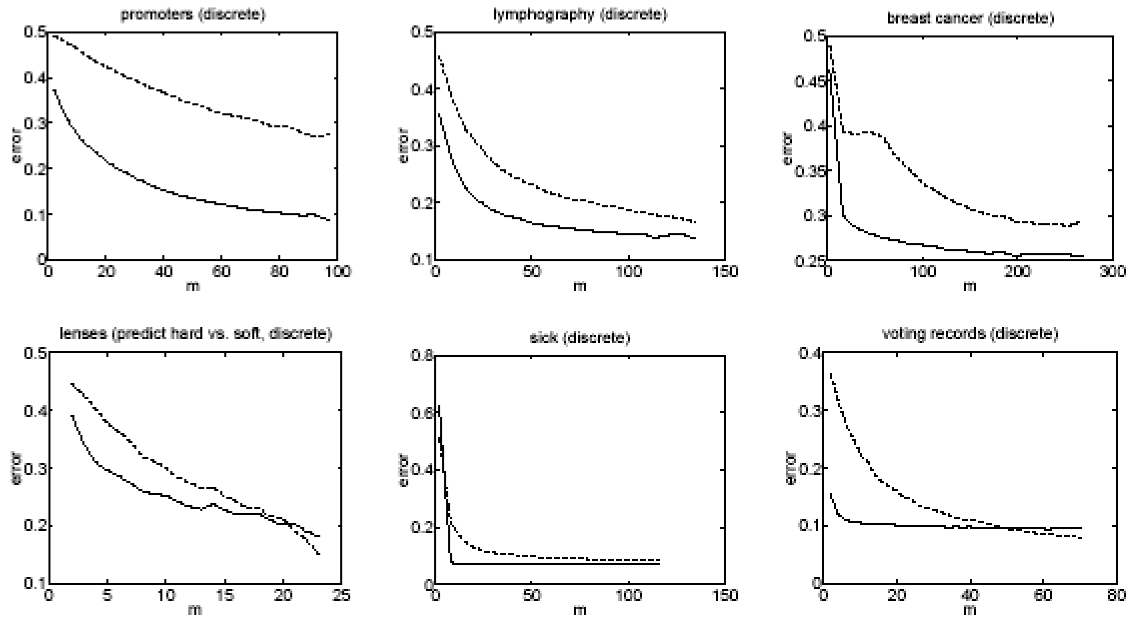


Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs. m (averaged over 1000 random train/test splits). Dashed line is logistic regression; solid line is naive Bayes.

Comparison

Generative Model

- Model P(x|y)
  - Model the input patterns

Discriminative Model

- Model P(y|x) directly
  - Model the decision boundary

Comparison

Generative Model

- Model P(x|y)
  - Model the input patterns
- Usually fast converge
- Cheap computation
- Robust to noise data

But

- Usually performs worse

Discriminative Model

- Model P(y|x) directly
  - Model the decision boundary
- Usually good performance

But

- Slow convergence
- Expensive computation
- Sensitive to noise in data

The Bias-Variance Decomposition (Regression)

- Assume that $Y = f(X) + \varepsilon$ where $E(\varepsilon) = 0$ and , $Var(\varepsilon) = \sigma_\varepsilon^2$ then at an input point, $X = x_0$

$$Err(x_0) = E[(Y - \hat{f}(x_0))^2 \mid X = x_0]$$

$$= \sigma_\varepsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - E\hat{f}(x_0)]^2$$

$$= \sigma_\varepsilon^2 + Bias^2(\hat{f}(x_0)) + Var(\hat{f}(x_0))$$

= Irreducible Error + Bias^2 + Variance

Bias, Variance and Model Complexity

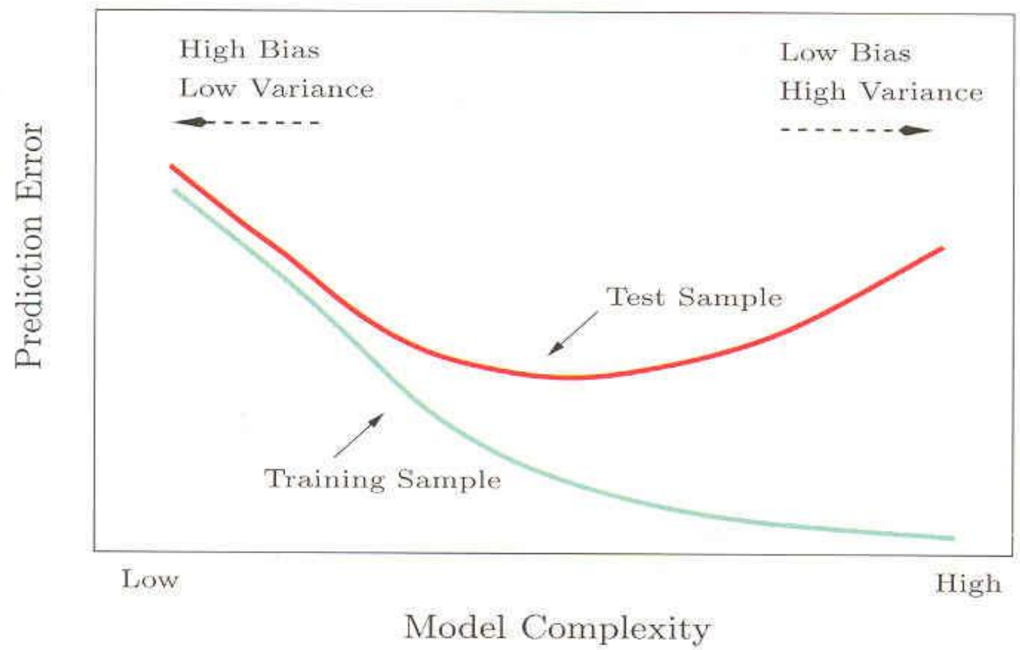
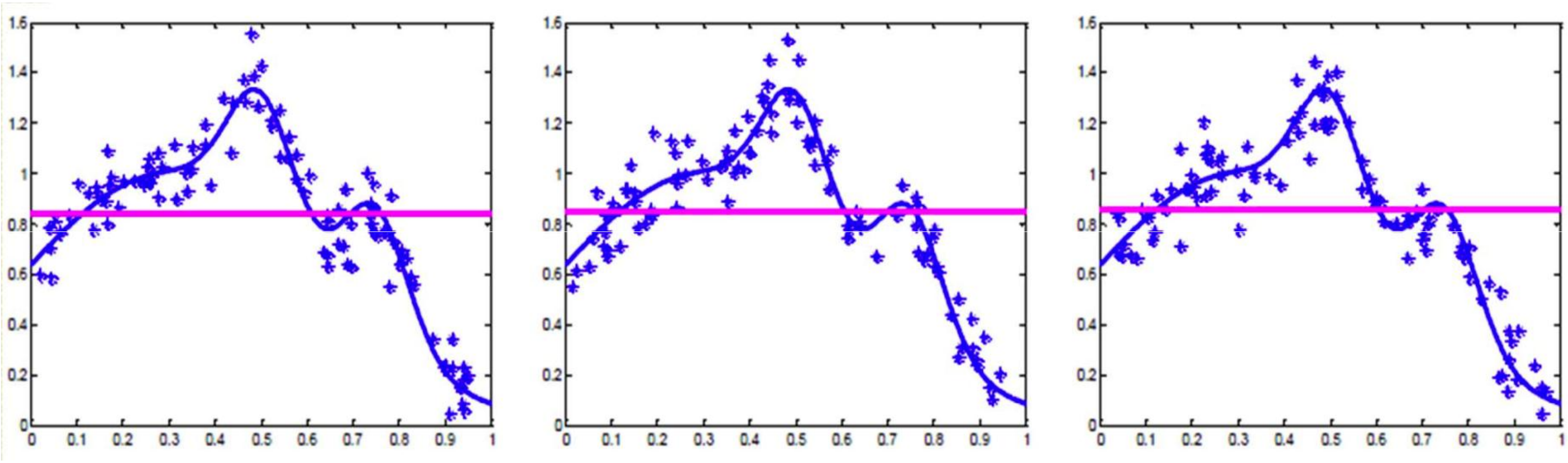


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied.*

- The figure is taken from Pg 194 of the book *The Elements of Statistical Learning* by Hastie, Tibshirani and Friedman.

Bias-Variance Tradeoff

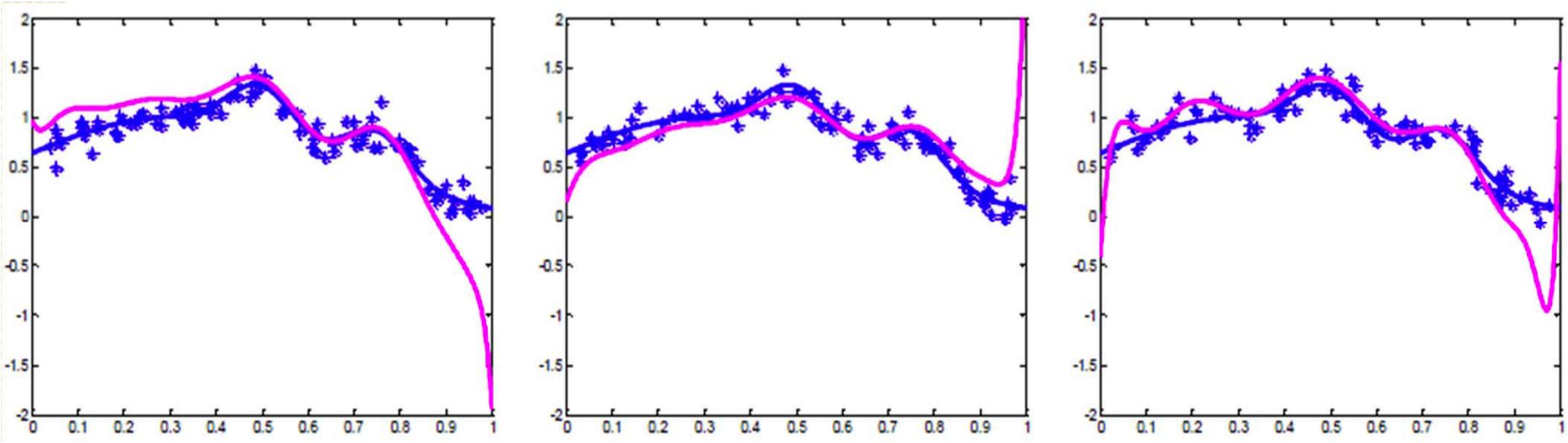
- Minimize both bias and variance ? No free lunch
- Simple models: low variance but high bias



  - Results from 3 random training sets ***D***
  - Estimation is very stable over 3 runs (low variance)
  - But estimated models are ***too simple*** (high bias)

Bias-Variance Tradeoff

- Minimize both bias and variance ? No free lunch
- Complex models: low bias but high variance



  - Results from 3 random training sets ***D***
  - Estimated models complex enough (low bias)
  - But estimation is unstable over 3 runs (high variance)

Bias-Variance Tradeoff

- We need a good tradeoff between bias and variance
- Class of models are not too simple (so that we can ***approximate*** the true function well)
- But not too complex to overfit the training samples (so that the ***estimation*** is ***stable***)

Problems with Logistic Regression?

$$p(\pm \mid \vec{x};\theta) = \frac{1}{1+\exp\left[\mp(c + x_1 w_1 + x_2 w_2 + ... + x_m w_m)\right]}$$

$$\theta = \{w_1, w_2, ..., w_m, c\}$$

How about words that only appears in one class?

Overfitting Problem with Logistic Regression

- Consider word t that only appears in one document d, and d is a positive document. Let w be its associated weight

$$
\begin{aligned}
l(D_{train}) &= \sum_{i=1}^{N(+)} \log p(+|d_i^+) + \sum_{i=1}^{N(-)} \log p(-|d_i^-) \\
&= \log p(+|d) + \sum_{d_i^+ \neq d} \log p(+|d_i^+) + \sum_{i=1}^{N(-)} \log p(-|d_i^-) \\
&= \log p(+|d) + l_+ + l_-
\end{aligned}
$$

Overfitting Problem with Logistic Regression

- Consider word *t* that only appears in one document *d*, and *d* is a positive document. Let *w* be its associated weight

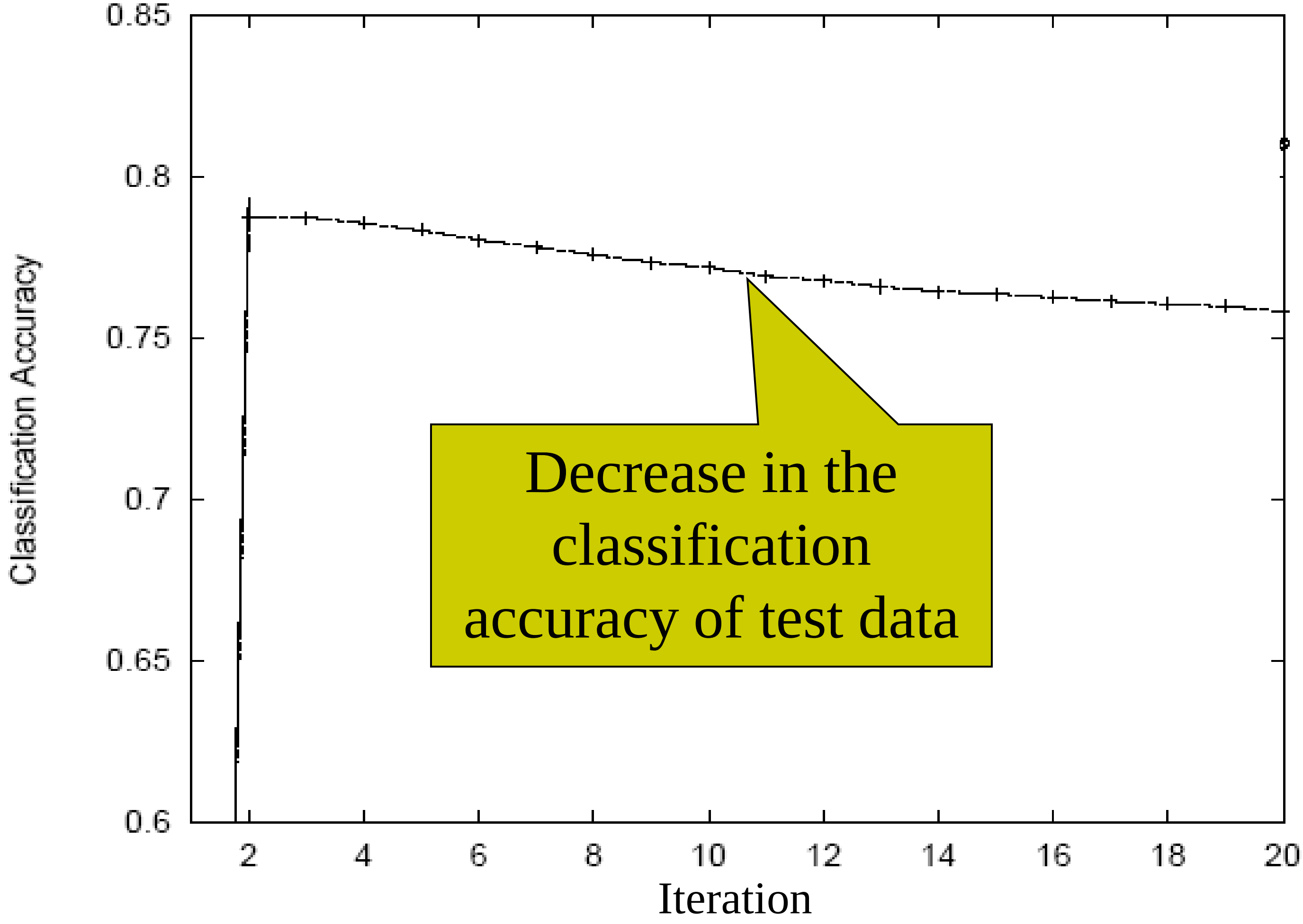
$$
\begin{aligned}
l(D_{train}) &= \sum_{i=1}^{N(+)} \log p(+|d_i^+) + \sum_{i=1}^{N(-)} \log p(-|d_i^-) \\
&= \log p(+|d) + \sum_{d_i^+ \neq d} \log p(+|d_i^+) + \sum_{i=1}^{N(-)} \log p(-|d_i^-) \\
&= \log p(+|d) + l_+ + l_-
\end{aligned}
$$

- Consider the derivative of $l(D_{train})$ with respect to *w*

$$
\frac{\partial l(D_{train})}{\partial w} = \frac{\partial \log p(+|d)}{\partial w} + \frac{\partial l_+}{\partial w} + \frac{\partial l_-}{\partial w} = \frac{1}{1+\exp\left[c + \vec{x}\cdot\vec{w}\right]} + 0 + 0 > 0
$$

- *w* will be infinite !

Example of Overfitting for LogRes



Solution: Regularization

- Regularized log-likelihood

$$l_{reg}(D_{train}) = l(D_{train}) - s\|\vec{w}\|_2^2$$

$$= \sum_{i=1}^{N(+)} \log p(+ \mid d_i^+) + \sum_{i=1}^{N(-)} \log p(- \mid d_i^-) - s\sum_{i=1}^{m} w_i^2$$

- $s||w||_2$ is called the regularizer
  - Favors small weights
  - Prevents weights from becoming too large

The Rare Word Problem

- Consider word *t* that only appears in one document d, and *d* is a positive document. Let *w* be its associated weight

$$
\begin{aligned}
l(D_{train}) &= \sum_{i=1}^{N(+)} \log p(+ \mid d_i^+) + \sum_{i=1}^{N(-)} \log p(- \mid d_i^-) \\
&= \log p(+ \mid d) + \sum_{d_i^+ \neq d} \log p(+ \mid d_i^+) + \sum_{i=1}^{N(-)} \log p(- \mid d_i^-) \\
&= \log p(+ \mid d) + l_+ + l_-
\end{aligned}
$$

$$
\begin{aligned}
l_{reg}(D_{train}) &= \sum_{i=1}^{N(+)} \log p(+ \mid d_i^+) + \sum_{i=1}^{N(-)} \log p(- \mid d_i^-) - s\sum_{i=1}^{m} w_i^2 \\
&= \log p(+ \mid d) + \sum_{d_i^+ \neq d} \log p(+ \mid d_i^+) + \sum_{i=1}^{N(-)} \log p(- \mid d_i^-) - s\sum_{i=1}^{m} w_i^2 \\
&= \log p(+ \mid d) + l_+ + l_- - s\sum_{i=1}^{m} w_i^2
\end{aligned}
$$

The Rare Word Problem

- Consider the derivative of $l(D_{train})$ with respect to w

$$\frac{\partial l(D_{train})}{\partial w} = \frac{\partial \log p(+|d)}{\partial w} + \frac{\partial l_+}{\partial w} + \frac{\partial l_-}{\partial w} = \frac{1}{1+\exp\left[c+\vec{x}\cdot\vec{w}\right]} + 0 + 0 > 0$$

⇓

$$\frac{\partial l_{reg}(D_{train})}{\partial w} = \frac{\partial \log p(+|d)}{\partial w} + \frac{\partial l_+}{\partial w} + \frac{\partial l_-}{\partial w} - 2sw$$

$$= \frac{1}{1+\exp\left[c+\vec{x}\cdot\vec{w}\right]} + 0 + 0 - 2sw$$

The Rare Word Problem

- Consider the derivative of $l(D_{train})$ with respect to w

$$\frac{\partial l(D_{train})}{\partial w} = \frac{\partial \log p(+|d)}{\partial w} + \frac{\partial l_+}{\partial w} + \frac{\partial l_-}{\partial w} = \frac{1}{1+\exp\left[c+\vec{x}\cdot\vec{w}\right]} + 0 + 0 > 0$$

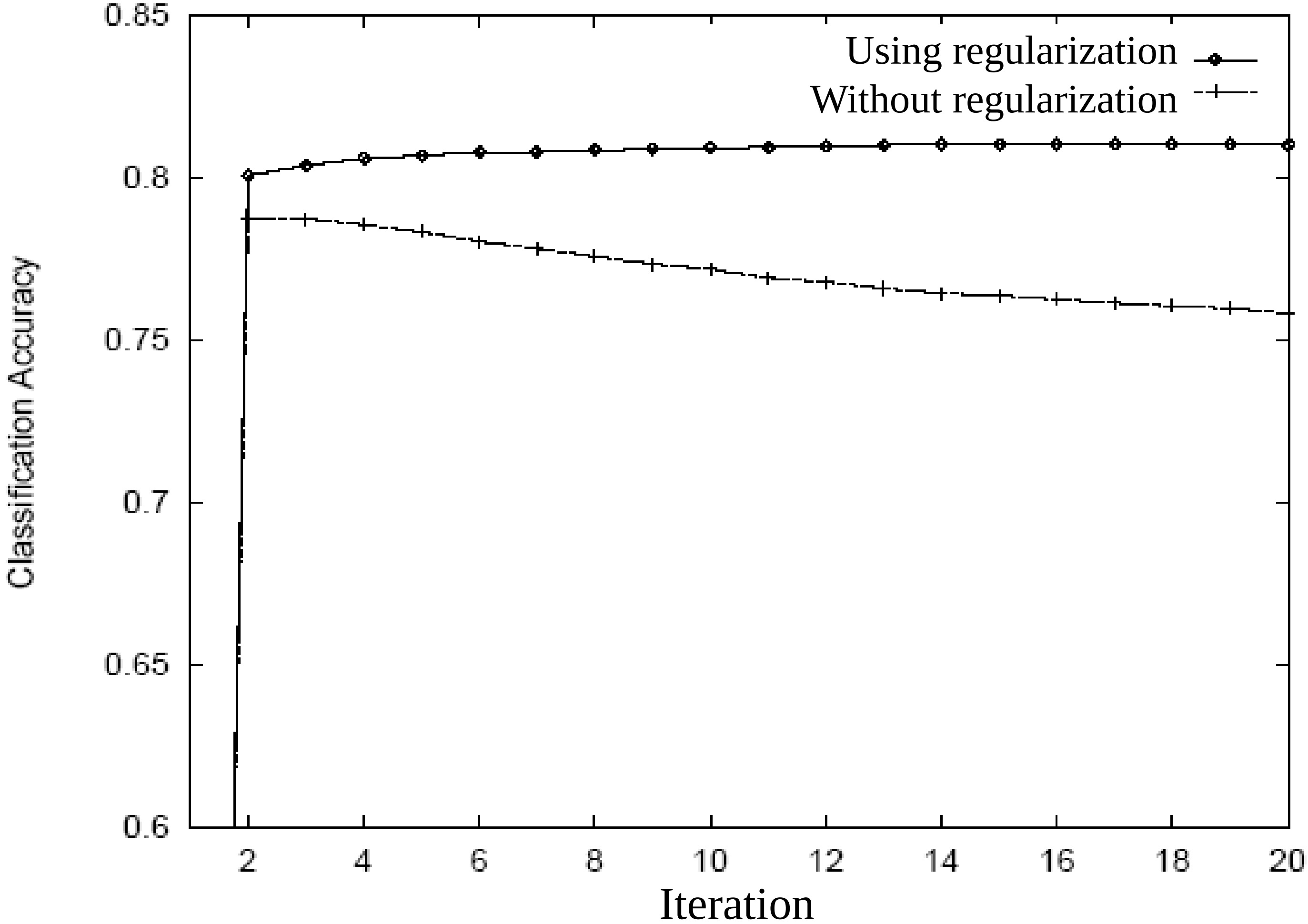
$$\Downarrow$$

$$\frac{\partial l_{reg}(D_{train})}{\partial w} = \frac{\partial \log p(+|d)}{\partial w} + \frac{\partial l_+}{\partial w} + \frac{\partial l_-}{\partial w} - 2sw$$

$$= \frac{1}{1+\exp\left[c+\vec{x}\cdot\vec{w}\right]} + 0 + 0 - 2sw$$

- When w is small, the derivative is still positive
- But, it becomes negative when w is large

Regularized Logistic Regression



Sparse Solution

- What does the solution of regularized logistic regression look like ?

Sparse Solution

- What does the solution of regularized logistic regression look like ?
- A sparse solution
  - Most weights are small and close to zero

Why do We Need Sparse Solution?

- Two types of solutions
  1. Many non-zero weights but many of them are small
  2. Only a small number of non-zero weights, and many of them are large
- Occam's Razor: the simpler the better
  - A simpler model that fits data unlikely to be coincidence
  - A complicated model that fit data might be coincidence
  - Smaller number of non-zero weights

    → less amount of evidence to consider

    → simpler model

    → case 2 is preferred

L1 vs. L2 Regularization

- L2 Regularizer
  - many weights are closer to zero
  - Easy to optimize
- L1 Regularizer

$$l_{reg}(D_{train}) = l(D_{train}) - s\left\|\vec{w}\right\|_1$$

  - Many weights are zero
  - More difficult to optimize

Feature Selection (discrete)

- Score each feature and ***select a subset***
  - Iterative method:
    - Select a highest score feature from the pool
    - ***Re-score*** the rest, e.g., cross-validation accuracy on already-selected features (plus this one)
    - Iterate
- Can also do in reverse direction
  - (remove one at a time)

Gradient Ascent

- Maximize the log-likelihood by iteratively adjusting the parameters in small increments
- In each iteration, we adjust w in the direction that increases the log-likelihood (toward the gradient)

Preventing weights from being too large

Prediction Errors

$$\frac{\partial}{\partial \vec{w}}\left\{\sum_{i=1}^{N}\log p(y_i \mid \vec{x}_i) - s\sum_{i=1}^{m} w_i^2\right\}$$

$$= \vec{w} + \varepsilon\left\{-s\vec{w} + \sum_{i=1}^{N}\vec{x}_i\left[y_i(1 - p(y_i \mid \vec{x}_i))\right]\right\}$$

Gradient Ascent

- Maximize the log-likelihood by iteratively adjusting the parameters in small increments
- In each iteration, we adjust w in the direction that increases the log-likelihood (toward the gradient)

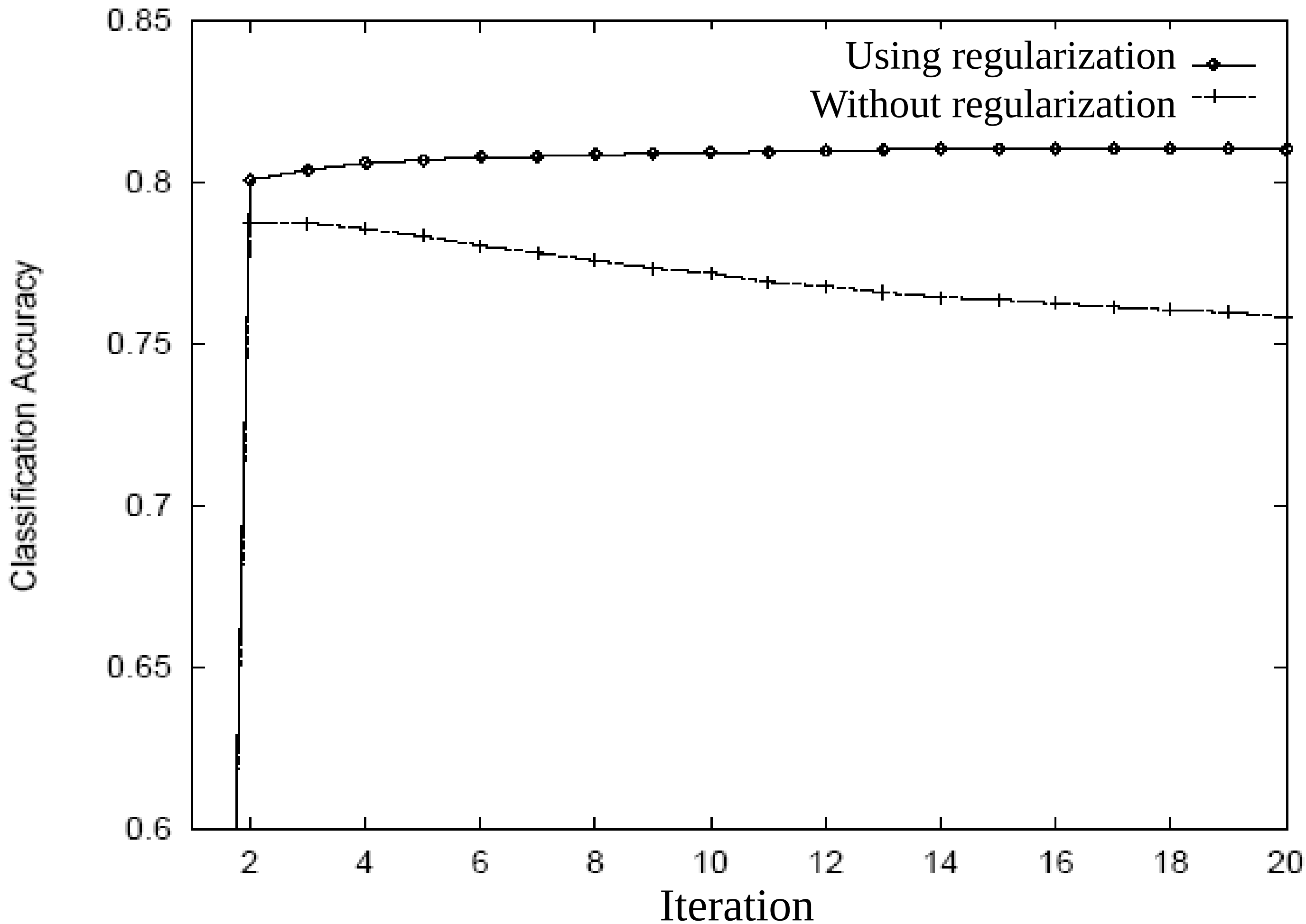
$$\vec{w} \leftarrow \vec{w} + \varepsilon \frac{\partial}{\partial \vec{w}} \left\{ \sum_{i=1}^{N} \log p(y_i \mid \vec{x}_i) - s \sum_{i=1}^{m} w_i^2 \right\}$$

$$= \vec{w} + \varepsilon \left\{ -s\vec{w} + \sum_{i=1}^{N} \vec{x}_i \left[y_i (1 - p(y_i \mid \vec{x}_i)) \right] \right\}$$

$$c \leftarrow c + \varepsilon \frac{\partial}{\partial c} \left\{ \sum_{i=1}^{N} \log p(y_i \mid \vec{x}_i) - s \sum_{i=1}^{m} w_i^2 \right\}$$

$$= c + \varepsilon \left\{ \sum_{i=1}^{N} y_i (1 - p(y_i \mid \vec{x}_i)) \right\}$$

where ε is learning rate.



0.85
0.8
0.75
0.7
0.65
0.6
Classification Accuracy
2
4
6
8
10
12
14
16
18
20
Iteration
Using regularization
Without regularization

When should Stop?

- The gradient ascent learning method converges when there is no incentive to move the parameters in any particular direction:

$$\frac{\partial}{\partial \vec{w}}\left\{\sum_{i=1}^{N}\log p(y_i \mid \vec{x}_i) - \sum_{i=1}^{m} w_i^2\right\} = \left\{-s\vec{w} + \sum_{i=1}^{N}\vec{x}_i\left[y_i(1 - p(y_i \mid \vec{x}_i))\right]\right\} = 0$$

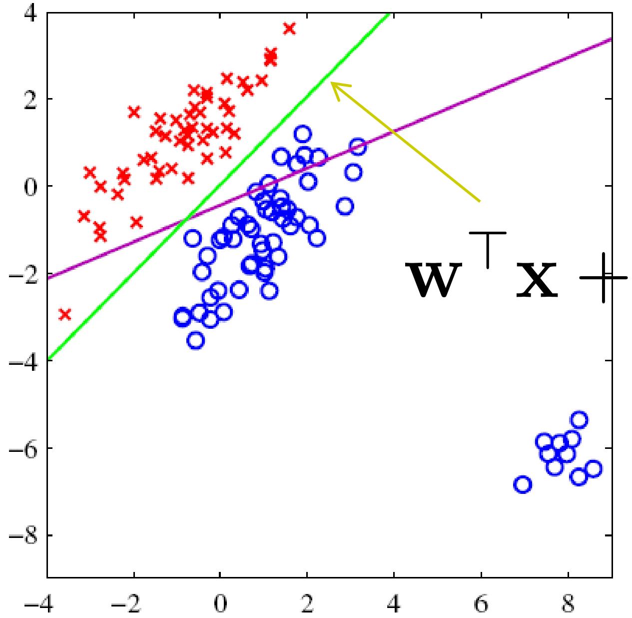
$$\frac{\partial}{\partial c}\left\{\sum_{i=1}^{N}\log p(y_i \mid \vec{x}_i) - \sum_{i=1}^{m} w_i^2\right\} = \left\{\sum_{i=1}^{N} y_i(1 - p(y_i \mid \vec{x}_i))\right\} = 0$$

Multi-class Logistic Regression

- How to extend logistic regression model to multi-class classification ?

$$\ln \frac{p(y=1|\mathbf{x})}{p(y=-1|\mathbf{x})} = \mathbf{w}^\top \mathbf{x}$$

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



$$\mathbf{w}^\top \mathbf{x} + b = 0$$

Conditional Exponential Model

- Let classes be $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K$

$$p(\mathcal{C}_k|\mathbf{x}) \propto \exp(\mathbf{w}_k^\top \mathbf{x})$$

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp(\mathbf{w}_k^\top \mathbf{x})$$

Normalization factor (partition function) $Z(\mathbf{x}) = \sum_{k=1}^{K} \exp(\mathbf{w}_k^\top \mathbf{x})$

- Need to learn $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K$

Conditional Exponential Model

- Learn weights **ws** by maximum conditional likelihood estimation

$$\mathcal{L}(W) = \sum_{i=1}^{N} \ln p(y_i|\mathbf{x}_i) = \sum_{i=1}^{N} \ln \frac{\exp(\mathbf{x}_i^\top \mathbf{w}_{y_i})}{\sum_{k=1}^{K} \exp(\mathbf{x}_i^\top \mathbf{w}_k)}$$

$$W^* = \arg\max_W \mathcal{L}(W)$$