机器学习A
5.线性模型I

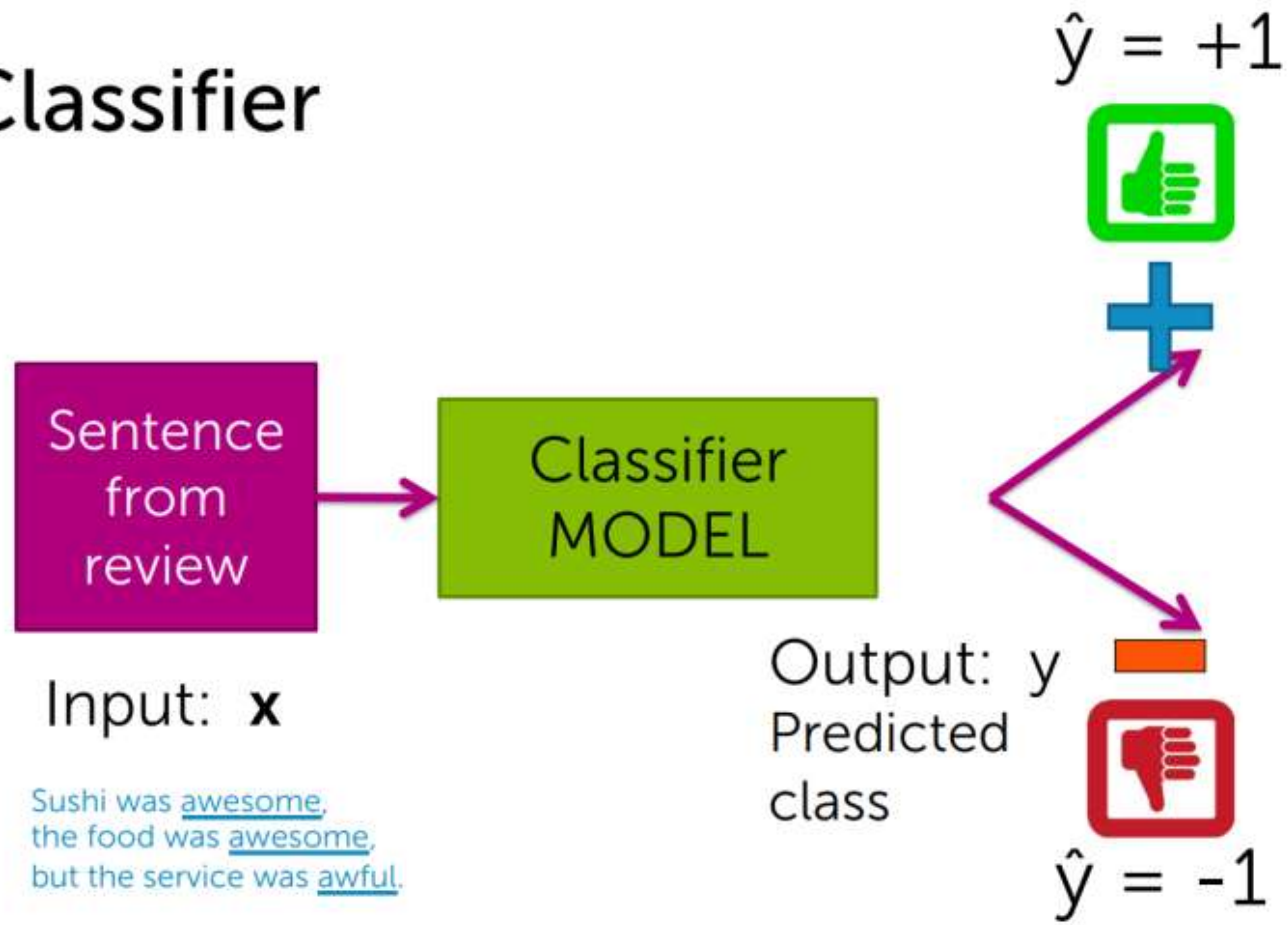
何向南

中国科学技术大学
数据科学实验室LDS

Classification
分类

Classification: Analyzing sentiment
分类器——分析情感

Classifier



Linear classifiers
线性分类器

Feature	Coefficient
…	…

Sentence from review

Input: **x**

Simple linear classifier

Score(**x**) = weighted sum of features of sentence

If *Score* (**x**) > *0*:

$\hat{y}$ = +1

Else:

$\hat{y}$ = -1

Linear classifiers
线性分类器

一个简单的例子：词频(Word Count):

特征(Feature)	系数(Coefficient)
good	1.0
great	1.2
awesome	1.7
bad	-1.0
terrible	-2.1
awful	-3.3
restaurant, the, we, where	0.0
…	…

输入 x_i

Sushi was great.
the food was awesome,
but the service was terrible.

Score = 1.2***1**+1.7***1**+(-2.1)***1**
= 0.8>0 **(加粗数字代表词频)**

$\hat{y}$ =+1 **(表示正面评论)**

Called a linear classifier, because score is weighted sum of features.

这叫线性分类器，因为分数是特征的加权和。

More generically...
更通用的...

Model: $\hat{y}_i = \text{sign}(\text{Score}(\mathbf{x}_i))$

Score(x_i) $= w_0 h_0(\mathbf{x_i}) + w_1 h_1(\mathbf{x_i}) + \ldots + w_D h_D(\mathbf{x_i})$

$$= \sum_{j=0}^{D} w_j h_j(\mathbf{x_i}) = w^\top h(\mathbf{x_i})$$

Sign(pos#)=+1
Sign(neg#)=-1

feature 1 = $h_0(\mathbf{x})$ ... e.g., 1

feature 2 = $h_1(\mathbf{x})$ ... e.g., $\mathbf{x}[1]$ = #awesome

feature 3 = $h_2(\mathbf{x})$ ... e.g., $\mathbf{x}[2]$ = #awful

or, $\log(\mathbf{x}[7])\ \mathbf{x}[2]$ = log(#bad) x #awful

or, tf-idf("awful")

...

feature D+1 = $h_D(\mathbf{x})$ ... some other function of $\mathbf{x}[1], \ldots, \mathbf{x}[d]$

Decision boundaries
决定边界

Suppose only two words had non-zero coefficient
假设我们只有2个非零系数

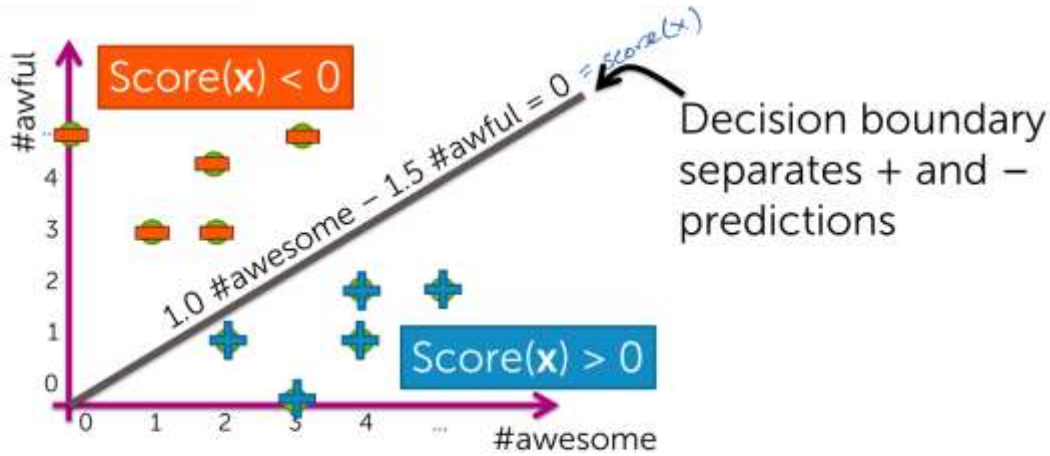
输入 (Input)	系数 (Coefficient)	值 (Value)
	w_0	0.0
#awesome	w_1	1.0
#awful	w_2	-1.5

Score(x)=1.0 #awesome – 1.5 #awful

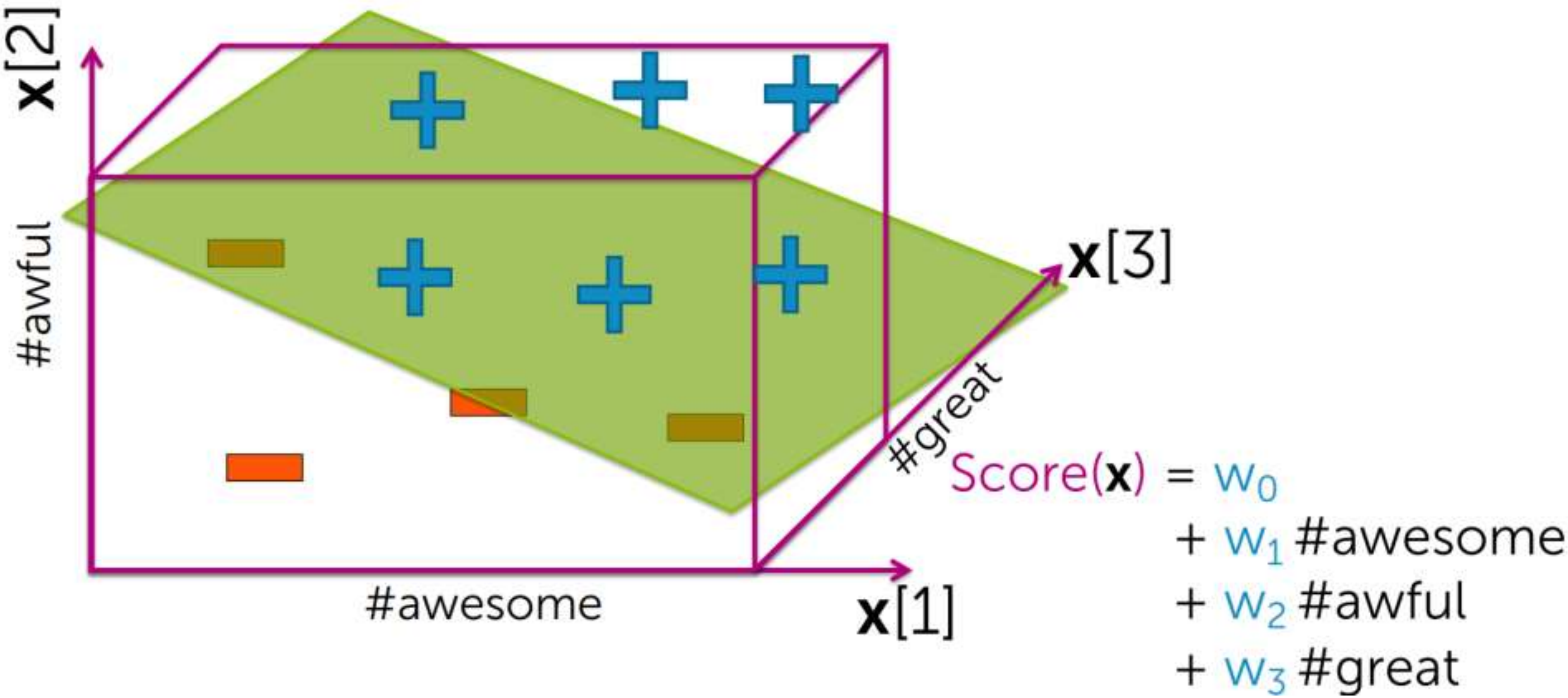
Decision boundaries example
决定边界例子

输入 (Input)	系数 (Coefficient)	值 (Value)
	w_0	0.0
#awesome	w_1	1.0
#awful	w_2	-1.5

Score(x)=1.0 #awesome – 1.5 #awful

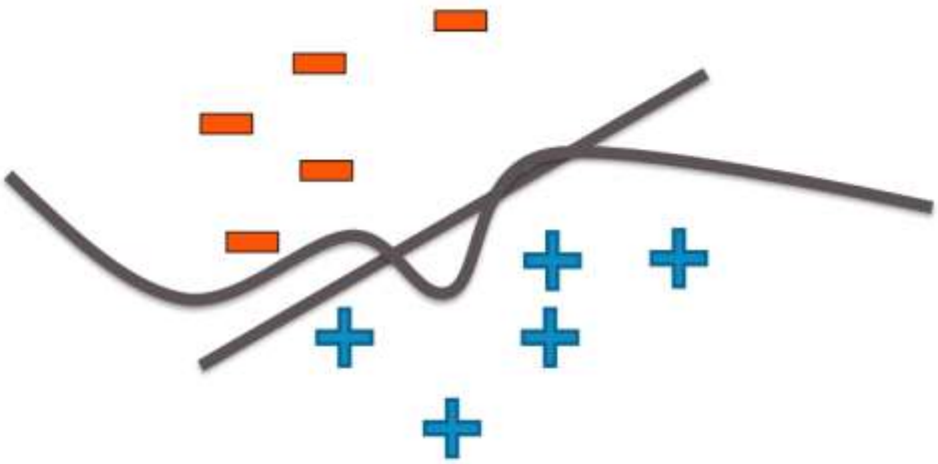


For more inputs (linear features)…
更多输入（线性特征）



For general features...
通用特征

For more general classifiers (not just linear features)
➔ more complicated shapes



Are you sure about the prediction? Class probability

你能确定预测吗？类别概率

How confident is your prediction?
预测的置信度?

- Thus far, we've outputted a prediction **+1** or **-1**
- But, how sure are you about the prediction?

"The sushi & everything else were awesome!"

↑

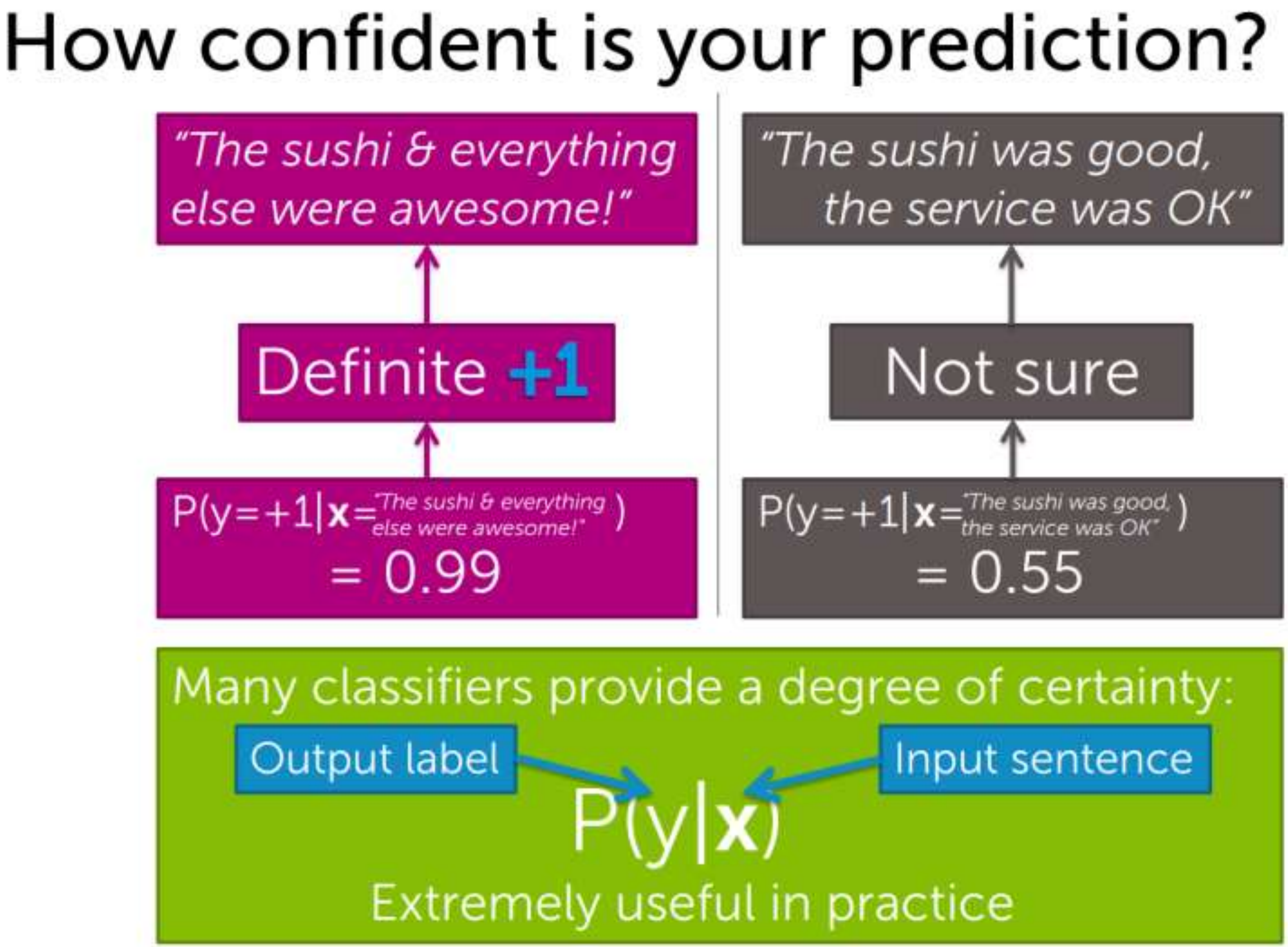
Definite **+1**

"The sushi was good, the service was OK"

↑

Not sure

Using probabilities in classification
在分类中使用概率



Goal: Learn conditional probabilities from data
目标：从数据中学习条件概率

Training data: N observations ($\mathbf{x}_i$,y_i)

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
0	2	-1
3	3	-1
4	1	+1
…	…	…

Optimize **quality metric** on training data

Find best model $\hat{P}$ by finding best $\hat{\mathbf{w}}$

Useful for predicting $\hat{y}$

Goal: Learn conditional probabilities from data
目标：从数据中学习条件概率

Sentence from review

Input: **x**

Predict most likely class

$\hat{P}(y|x)$ = estimate of class probabilities

If $\hat{P}(y=+1|x) > 0.5$:

$\hat{y} = +1$

Else:

$\hat{y} = -1$

Estimating $\hat{P}(y|x)$ improves **interpretability**:

– Predict $\hat{y} = +1$ **and** tell me how sure you are

Predicting class probabilities with logistic regression

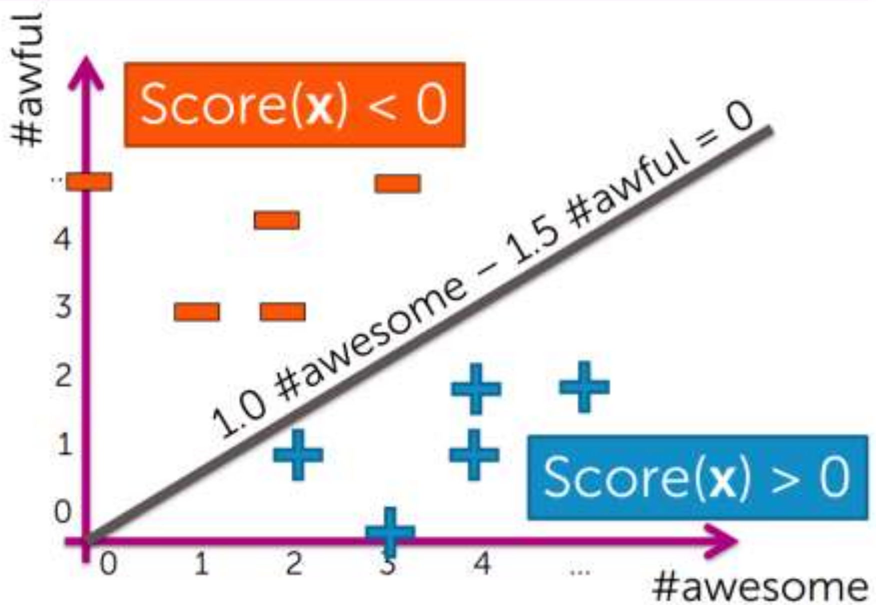
用Logistic回归预测类别概率

Predicting class probabilities with logistic regression
用Logistic回归预测类别概率

Thus far, we focused on decision boundaries

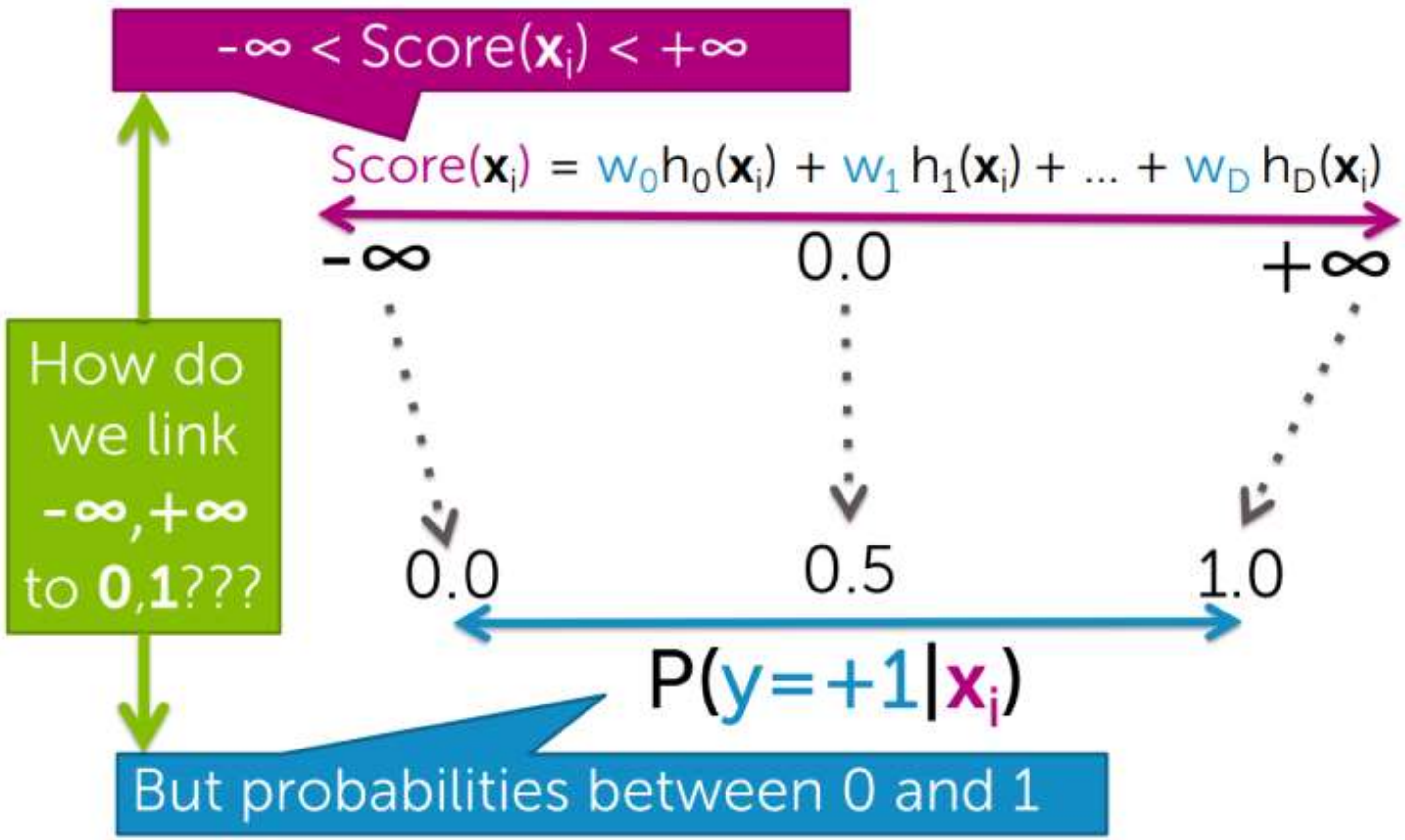
目前为止，我们关注决策边界

$$\text{Score}(\mathbf{x}_i) = w_0\, h_0(\mathbf{x}_i) + w_1\, h_1(\mathbf{x}_i) + \ldots + w_D\, h_D(\mathbf{x}_i) = \mathbf{w}^T h(\mathbf{x}_i)$$



Relate Score($\mathbf{x}_i$) to $\hat{P}(y=+1|\mathbf{x},\hat{\mathbf{w}})$?

Use regression to build classifier
用回归构建分类

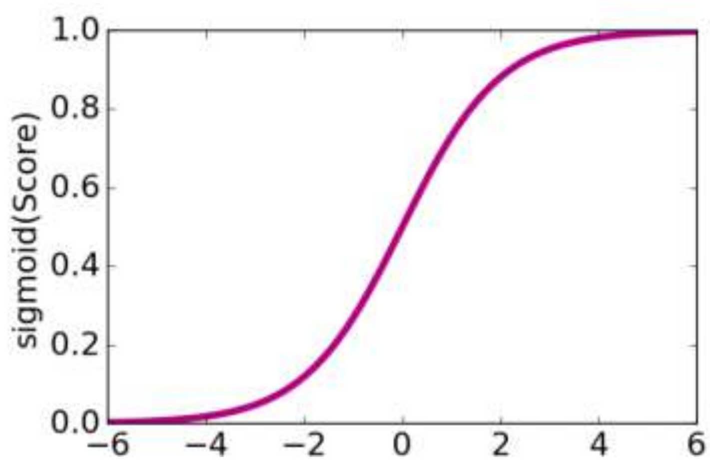


Use regression to build classifier
用回归构建分类

Logistic 函数 （也叫做Sigmoid，Logit）

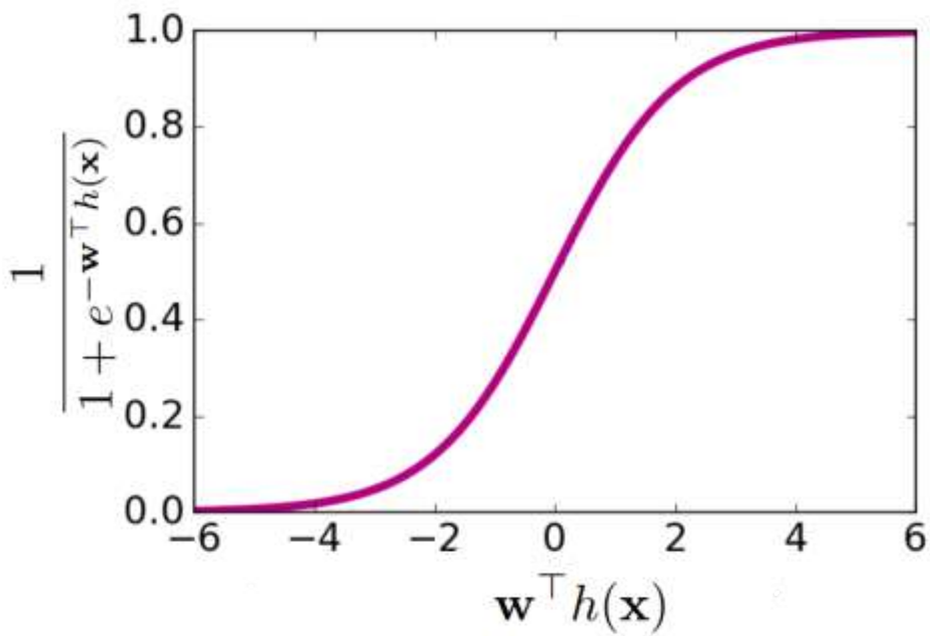
$$\mathbf{Sigmoid}(\boldsymbol{Score}) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-\boldsymbol{Score}}}$$

Score	-∞	-2	0	+2	+∞
Sigmoid	0	0.12	0.5	0.88	1



Understanding the logistic regression model
理解Logistic回归

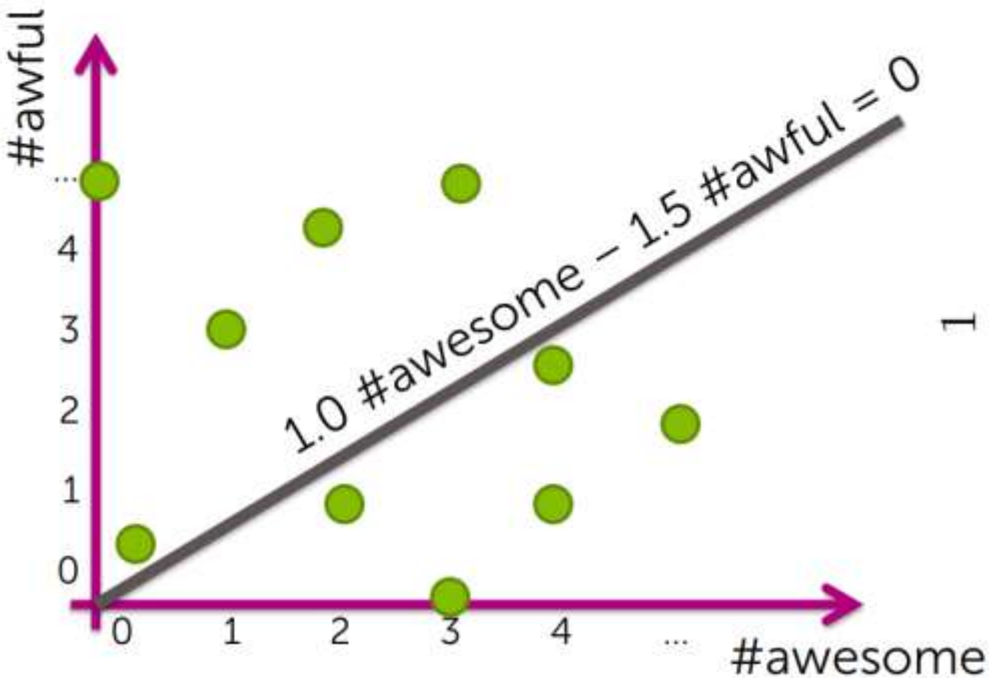
$$P(y=+1|\mathbf{x}_i,\mathbf{w}) = \text{sigmoid}(\text{Score}(\mathbf{x}_i)) = \frac{1}{1+e^{-\mathbf{w}^{T}h(\mathbf{x})}}$$

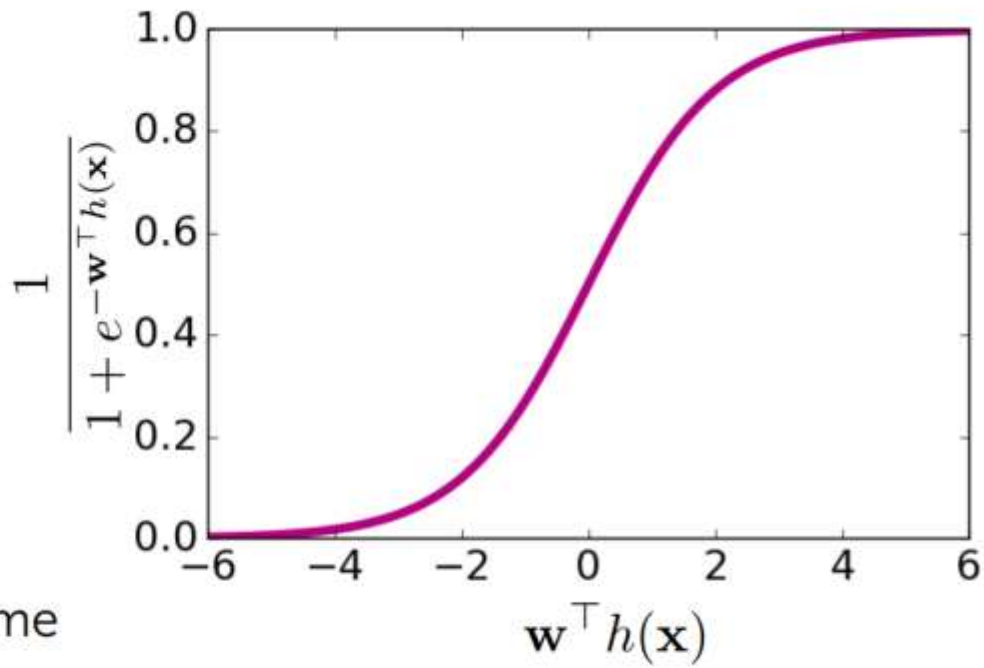


Score($\mathbf{x}_i$)	$P(y=+1\|\mathbf{x}_i,\mathbf{w})$
0	0.5
-2	0.12
2	0.88
4	0.98

$$Score = \boldsymbol{w}^{T}h(x)$$

Logistic regression -> Linear decision boundary
Logistic回归 -> 线性决策边界



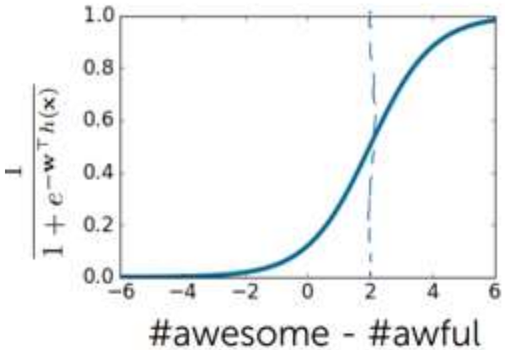


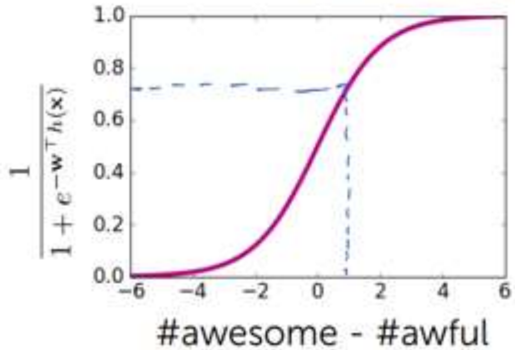
Effect of coefficients on logistic regression model
Logistic回归模型系数的影响

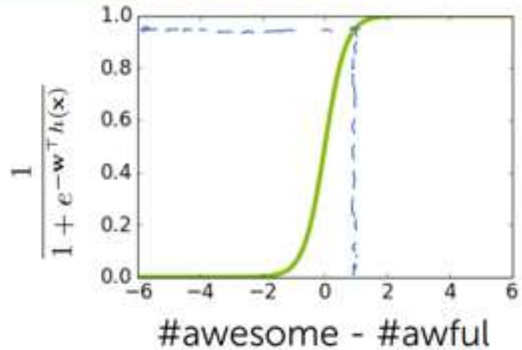
w_0	-2
$w_{\#awesome}$	+1
$w_{\#awful}$	-1

w_0	0
$w_{\#awesome}$	+1
$w_{\#awful}$	-1

w_0	0
$w_{\#awesome}$	+3
$w_{\#awful}$	-3







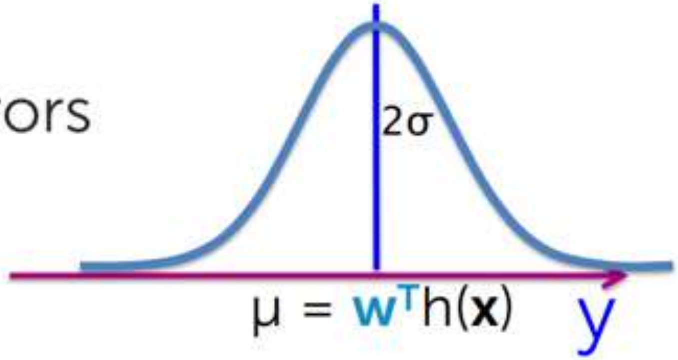
Score(x) = -2 + #awesome - #awful

Compare and contrast regression models
对比回归模型

- Linear regression with Gaussian errors

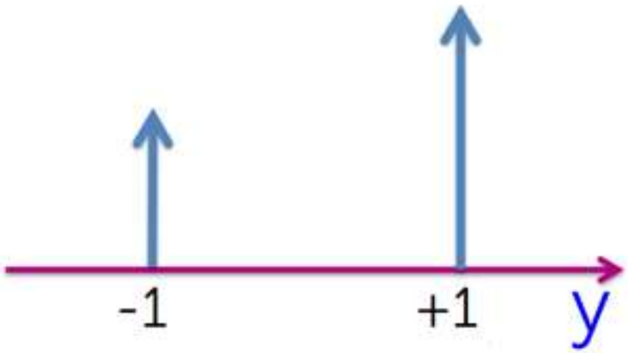
  $y_i = \mathbf{w}^T h(\mathbf{x}_i) + \varepsilon_i \quad \varepsilon_i \sim N(0,\sigma^2)$

  $\rightarrow p(y|\mathbf{x},\mathbf{w}) = N(y;\, \mathbf{w}^T h(\mathbf{x}),\, \sigma^2)$



- Logistic regression

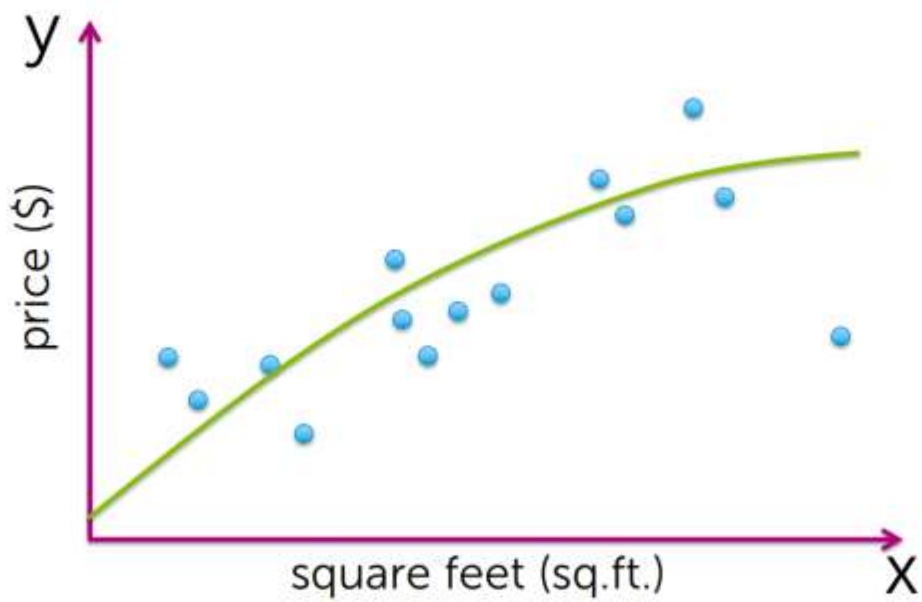
$$P(y|\mathbf{x},\mathbf{w}) = \begin{cases} \dfrac{1}{1+e^{-\mathbf{w}^T h(\mathbf{x})}} & y = +1 \\ \dfrac{e^{-\mathbf{w}^T h(\mathbf{x})}}{1+e^{-\mathbf{w}^T h(\mathbf{x})}} & y = -1 \end{cases}$$



Loss function for logistic regression: (Negative log-) Likelihood for maximum likelihood estimation (MLE)

Logistic回归的损失函数：最大似然估计（MLE）的（负对数）似然性

Recall: Gaussian linear regression model
回顾：高斯线性回归模型



ε_i 的模型：

初步假设：$E[\varepsilon_i] = 0$
更强的假设：$\varepsilon_i \sim N(0, \sigma^2)$

y_i的分布：

$$\mathrm{y_i} = h^T(x_i)w + \varepsilon_i$$
$$P(\mathrm{y_i}|x_i; w) \sim N\left(h^T(x_i)w, \sigma^2\right)$$

Recall: Maximum likelihood estimation
回顾：最大似然估计

Maximize log-likelihood w.r.t w

$$\ln p(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{w}) = \ln[(\frac{1}{\sigma\sqrt{2\pi}})^N \times \prod_{i=1}^{N} \exp(\frac{-(y_i - \boldsymbol{h}(\boldsymbol{x_i})^T\boldsymbol{w})^2}{2\sigma^2})]$$

$$\underset{w}{\operatorname{argmax}}\ p(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{w}) = \underset{w}{\operatorname{argmax}} \ln p(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{w})$$

$$= \underset{w}{\operatorname{argmax}} \left[-N \ln \sigma\sqrt{2\pi} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \boldsymbol{h}(\boldsymbol{x_i})^T\boldsymbol{w})^2\right]$$

$$= \underset{w}{\operatorname{argmin}} \sum_{i=1}^{N}(y_i - \boldsymbol{h}(\boldsymbol{x_i})^T\boldsymbol{w})^2$$

MLE估计最终形式和最小均方差形式相同

Finding best coefficients
寻找最好的系数

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
0	2	-1
3	3	-1
4	1	+1
1	1	+1
2	4	-1
0	3	-1
0	1	-1
2	1	+1

Finding best coefficients
寻找最好的系数

x[1] = #awesome	x[2] = #awful	y = sentiment
0	2	-1
3	3	-1
2	4	-1
0	3	-1
0	1	-1
2	4	-1
0	3	-1
0	1	-1

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
4	1	+1
1	1	+1
2	1	+1
1	1	+1
2	1	+1

Finding best coefficients
寻找最好的系数

x[1] = #awesome	x[2] = #awful	y = sentiment
0	2	-1
3	3	-1
2	4	-1
0	3	-1
0	1	-1

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
4	1	+1
1	1	+1
2	1	+1

$$P(y=+1|\mathbf{x}_i,\mathbf{w}) = 0.0 \qquad P(y=+1|\mathbf{x}_i,\mathbf{w}) = 1.0$$

Want $\hat{\mathbf{w}}$ that makes

Finding best coefficients
寻找最好的系数

Learn logistic regression model with maximum likelihood estimation (MLE)
用MLE学习Logistic回归

Data point	x[1]	x[2]	y	Choose w to maximize
$\mathbf{x}_1, y_1$	2	1	+1	$P(y=+1 \mid \mathbf{x}[1]=2, \mathbf{x}[2]=1, \mathbf{w})$
$\mathbf{x}_2, y_2$	0	2	-1	$P(y=-1 \mid \mathbf{x}[1]=0, \mathbf{x}[2]=2, \mathbf{w})$
$\mathbf{x}_3, y_3$	3	3	-1	$P(y=-1 \mid \mathbf{x}[1]=3, \mathbf{x}[2]=3, \mathbf{w})$
$\mathbf{x}_4, y_4$	4	1	+1	$P(y=+1 \mid \mathbf{x}[1]=4, \mathbf{x}[2]=1, \mathbf{w})$

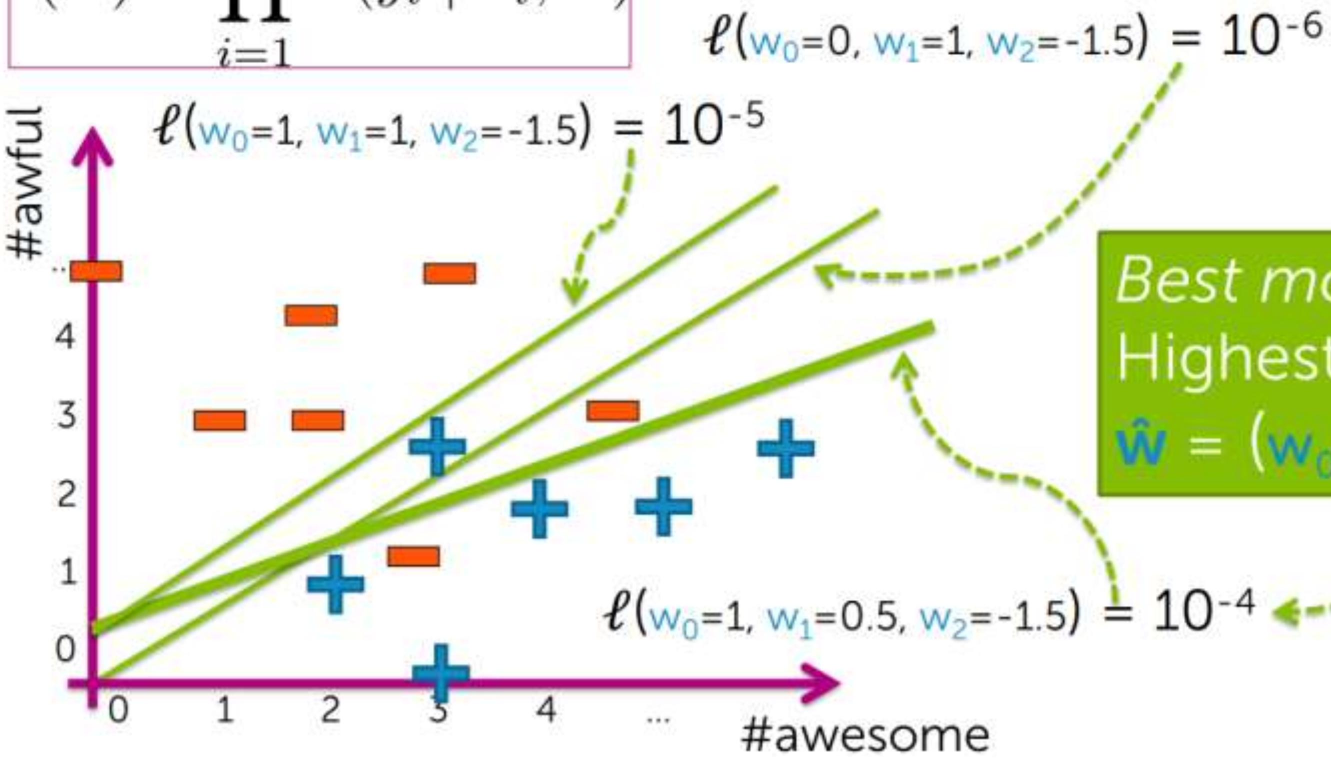
$$P(y \mid \mathbf{x}, \mathbf{w}) = \begin{cases} \dfrac{1}{1 + e^{-\mathbf{w}^\top h(\mathbf{x})}} & y = +1 \\[2ex] \dfrac{e^{-\mathbf{w}^\top h(\mathbf{x})}}{1 + e^{-\mathbf{w}^\top h(\mathbf{x})}} & y = -1 \end{cases}$$

$$\ell(\mathbf{w}) = \underbrace{\overbrace{P(y_1 \mid \mathbf{x}_1, \mathbf{w})} \; \overbrace{P(y_2 \mid \mathbf{x}_2, \mathbf{w})} \; \overbrace{P(y_3 \mid \mathbf{x}_3, \mathbf{w})} \; \overbrace{P(y_4 \mid \mathbf{x}_4, \mathbf{w})}}_{\prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i, \mathbf{w})}$$

Find best classifier
寻找最好的分类器

Maximize likelihood over all possible w_0, w_1, w_2

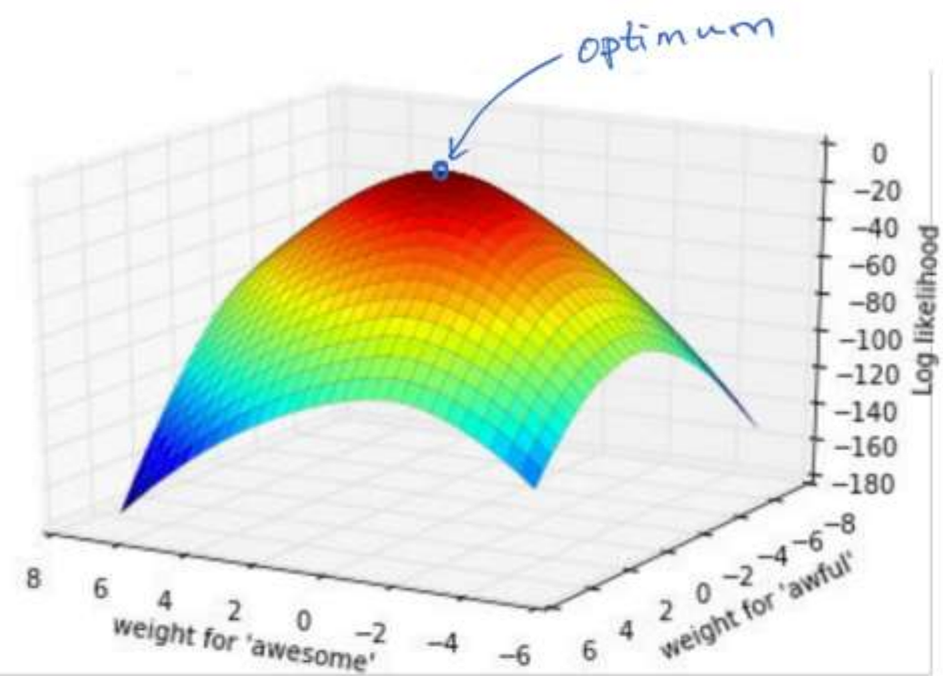
$$\ell(\mathbf{w}) = \prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i, \mathbf{w})$$



Best model:
Highest likelihood $\ell(\mathbf{w})$
$\hat{\mathbf{w}} = (w_0=1, w_1=0.5, w_2=-1.5)$

Gradient ascent for logistic regression
Logistic回归的梯度上升

Maximizing likelihood 最大似然



Maximize function over all possible w_0, w_1, w_2

$$\underbrace{\max_{w_0,w_1,w_2} \prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i, \mathbf{w})}$$

$\ell(w_0,w_1,w_2)$ is a function of 3 variables

从最大化概率似然的角度出发，概率越大越好，因此应该执行梯度上升，而不是针对损失函数的梯度下降

Gradient ascent for logistic regression
Logistic回归的梯度上升

Our optimization objective 优化目标

- **Can compute gradient, but no closed-form solution to:**
  能够计算梯度，但不一定是闭式解（指有公式的解）：
  $$\nabla l(w) = 0$$
- **Use gradient ascent**
  使用梯度上升
- **As with MLE for Gaussians, rewrite objective as**
  对于高斯分布的MLE，把目标重写为
  $$\hat{w} = \underset{w}{argmax}\, l(w) = \underset{w}{argmax}\, ll(w) \text{ (Log-Likelihood)}$$

Gradient of logistic log-likelihood
Logistic对数似然的梯度

Sum over data points

Feature value

$$\frac{\partial \ell\ell(\mathbf{w})}{\partial \mathbf{w}_j} = \sum_{i=1}^{N} h_j(\mathbf{x}_i)\Big(\underbrace{\overbrace{\mathbb{1}[y_i = +1] - P(y = +1 \mid \mathbf{x}_i, \mathbf{w})}^{\text{Difference between truth and prediction}}}_{\Delta_i}\Big)$$

$$\frac{\partial ll(\boldsymbol{w})}{\partial \boldsymbol{w_j}} = \sum\nolimits_{y_i=1} \frac{\partial \ln \frac{1}{1+\exp(-\boldsymbol{w}^T \boldsymbol{h}(\boldsymbol{x}))}}{\partial \boldsymbol{w_j}} + \sum\nolimits_{y_i=-1} \frac{\partial \ln \frac{\exp(-\boldsymbol{w}^T \boldsymbol{h}(\boldsymbol{x}))}{1+\exp(-\boldsymbol{w}^T \boldsymbol{h}(\boldsymbol{x}))}}{\partial \boldsymbol{w_j}}$$

$$= -\sum\nolimits_{y_i=1} \frac{1}{1+u} \frac{\partial u}{\partial \boldsymbol{w_j}} + \sum\nolimits_{y_i=-1} \frac{1}{(1+u)u} \frac{\partial u}{\partial \boldsymbol{w_j}} \quad \ldots\ldots \ \underline{u \equiv} \exp\left(-\boldsymbol{w}^T \boldsymbol{h}(\boldsymbol{x})\right)$$

$$= \sum\nolimits_{y_i=1} \frac{u}{1+u}\, h_j(\boldsymbol{x}) - \sum\nolimits_{y_i=-1} \frac{1}{(1+u)} h_j(\boldsymbol{x}) = \text{上式}$$

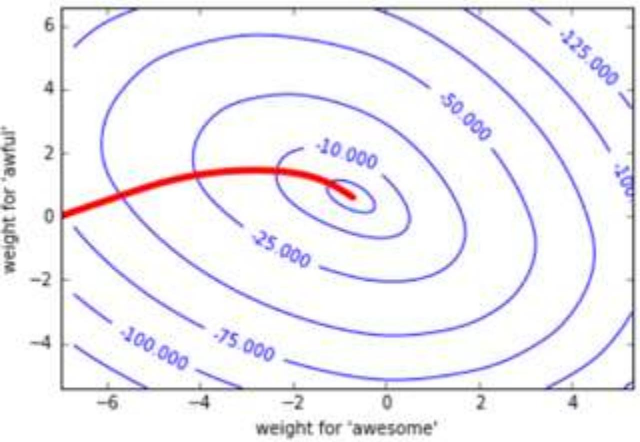
(注：$1[Condition] = 1$ 当且仅当condition为真，否则为0。)

Gradient of logistic log-likelihood
Logistic对数似然的梯度

$$\frac{\partial \ell\ell(\mathbf{w})}{\partial \mathbf{w}_j} = \underbrace{\sum_{i=1}^{N}}_{\text{Sum over data points}} \underbrace{h_j(\mathbf{x}_i)}_{\text{Feature value}} \overbrace{\underbrace{\left(\mathbb{1}[y_i = +1] - P(y = +1 \mid \mathbf{x}_i, \mathbf{w})\right)}_{\Delta_i}}^{\text{Difference between truth and prediction}}$$

	$P(y=+1\|x_i,w) \approx 1$	$P(y=+1\|x_i,w) \approx 0$
$y_i = +1$	$\Delta_i \approx 0$ 不改变	$\Delta_i \approx 1$ 增加w_j 增加$P(y=+1\|x_i,w)$
$y_i = -1$	$\Delta_i \approx 1$ 减少w_j 减少$P(y=+1\|x_i,w)$	$\Delta_i \approx 0$ 不改变

Gradient ascent for logistic regression
Logistic回归的梯度上升



init $\mathbf{w}^{(1)}=0$ (or randomly, or smartly), $t=1$

while $||\nabla \ell(\mathbf{w}^{(t)})|| > \varepsilon$

for j=0,…,D

$$\text{partial[j]} = \sum_{i=1}^{N} h_j(\mathbf{x}_i)\overbrace{\left(\mathbb{1}[y_i=+1] - \underbrace{P(y=+1 \mid \mathbf{x}_i, \mathbf{w}^{(t)})}_{\frac{1}{1+\exp(-\boldsymbol{w}^T \boldsymbol{h}(\boldsymbol{x}_i))}}\right)}^{\text{Difference between truth and prediction}}$$

$$w_j^{(t+1)} \leftarrow w_j^{(t)} + \eta\ \underbrace{\text{partial[j]}}_{\frac{\partial ll(\boldsymbol{w}^{(t)})}{\partial \boldsymbol{w_j}}}$$

$t \leftarrow t+1$

Summary for linear classifiers and logistic regression
线性分类器和Logistic回归的总结

- Describe decision boundaries and linear classifiers
- Use class probability to express degree of confidence in prediction
- Define a logistic regression model
- Interpret logistic regression outputs as class probabilities
- Describe impact of coefficient values on logistic regression output
- Measure quality of a classifier using the likelihood function
- Learn a logistic regression model with gradient descent for negative log-likelihood loss

Linear classifiers: overfitting

线性分类器：过拟合

Review: Bias, variance and error
回顾：偏差，方差，误差

如果我们将我们尝试预测的变量表示为Y和我们的协变量作为X，我们可以假设存在一种将一个与另一个相关的关系，例如$Y = f(X) + \varepsilon$其中 error 项ε呈正态分布，均值为零，如下所示$\varepsilon \sim \mathcal{N}(0、\ \sigma_\varepsilon)$

我们可以估计一个模型$\hat{f}(X)$之$f(X)$使用线性回归或其他建模技术。在这种情况下，某个点的预期平方预测误差x是：

$$Err(x) = E\left[(Y - \hat{f}(x))^2\right]$$

然后，此错误可以分解为偏差分量和方差分量：

$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] + \sigma_e^2$$

$$Err(x) = \mathrm{Bias}^2 + \mathrm{Variance} + \mathrm{Irreducible\ Error}$$

第三项，不可约误差，是真实关系中的噪声项，任何模型都无法从根本上简化。给定真实模型和无限数据来校准它，我们应该能够将偏差和方差项都减少到 0。然而，在模型不完美且数据有限的世界中，需要在最小化偏差和最小化方差之间进行权衡。

Review: Bias, variance and error
回顾：偏差，方差，误差

$$Y = f(X) + \epsilon$$

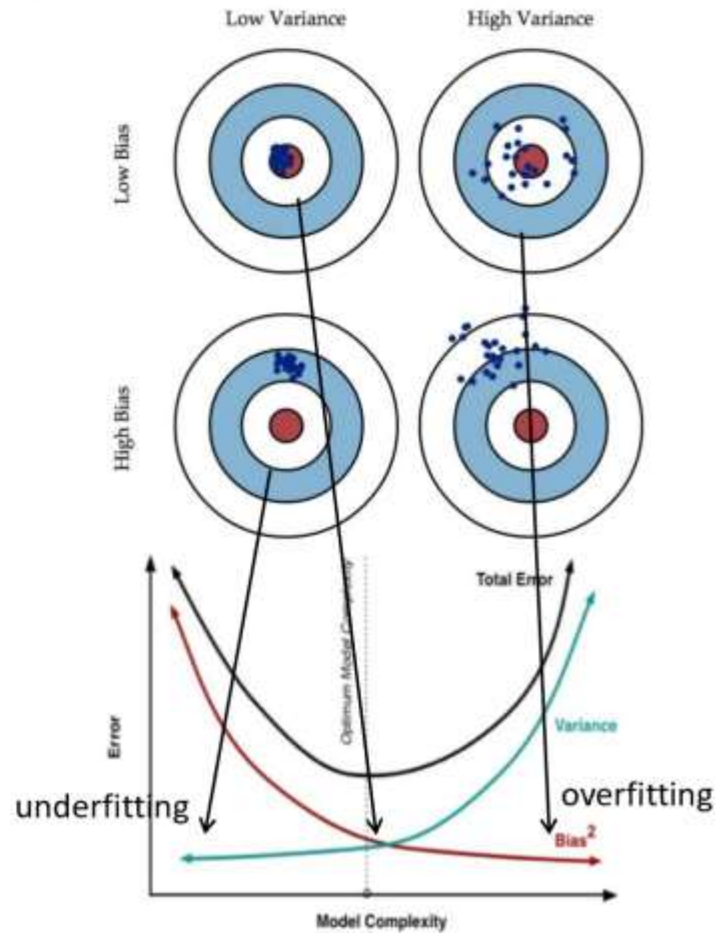
观测值 真实值 误差

观测值 预测值

- 泛化误差$E[(y - \hat{f}(\boldsymbol{x}))^2]$

$$= Var(\boldsymbol{x}) + bias^2(\boldsymbol{x}) + variance(\boldsymbol{x})$$

Irreducible Error	偏差的平方	方差
真实值与观测值之间的方差	真实值与模型预测值期望之间的方差	模型预测值与预测值期望之间的方差



- 偏差($bias$):模型依靠自身能力进行预测的平均准确程度（准）
- 方差($variance$):模型在不同训练集上表现出来的差异程度（确）

More complex models tend to have less bias…
更复杂的模型倾向于更小的偏差

Sentiment classifier using single words can do OK, but…

Never classifies correctly:
"The sushi was not good."

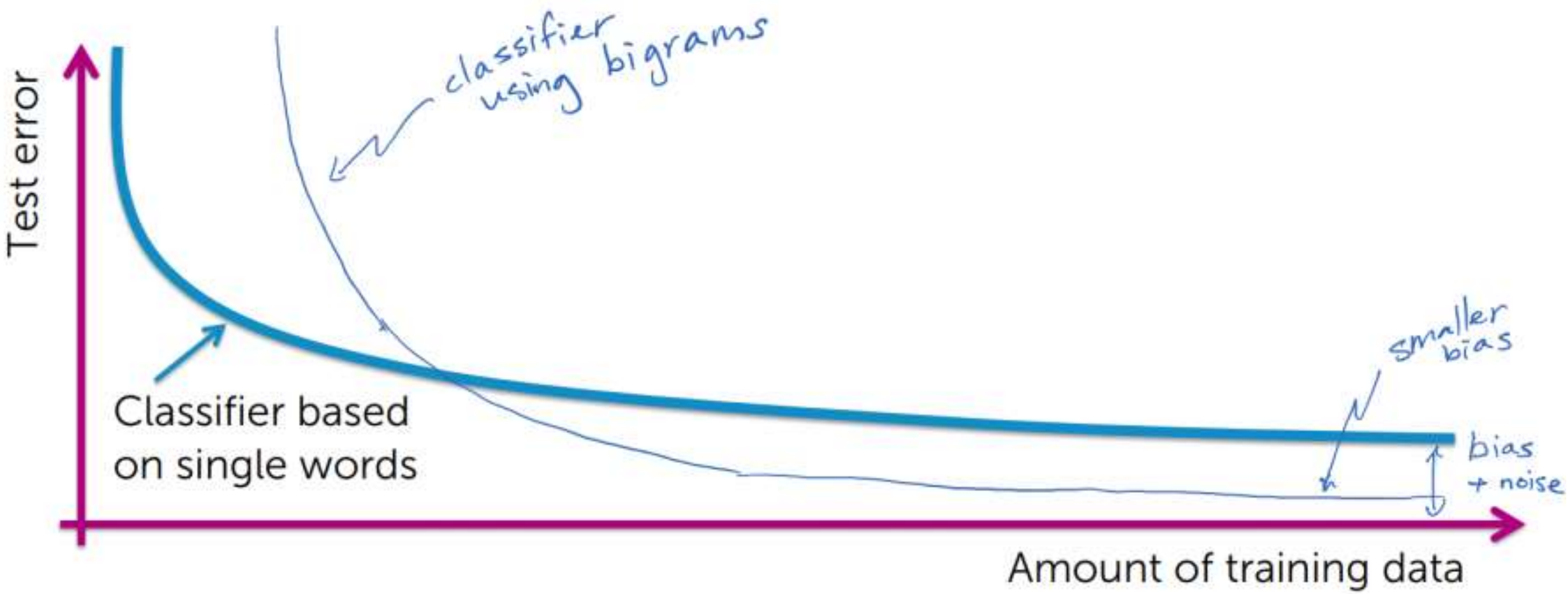
More complex model:
consider pairs of words (bigrams)

Word	Weight
good	+1.5
not good	-2.1

Less bias →
potentially more accurate,
needs more data to learn

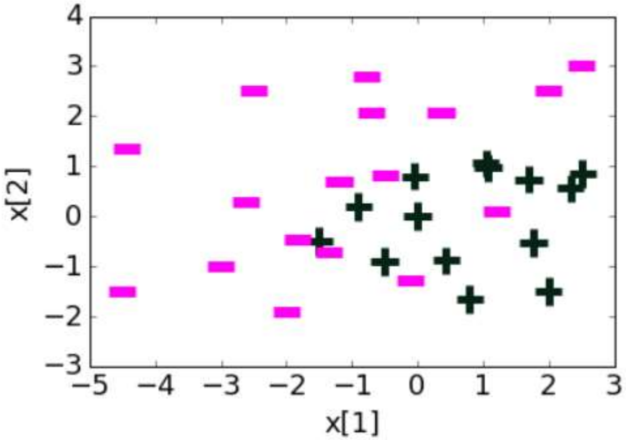
More complex models tend to have less bias…
更复杂的模型倾向于更小的偏差

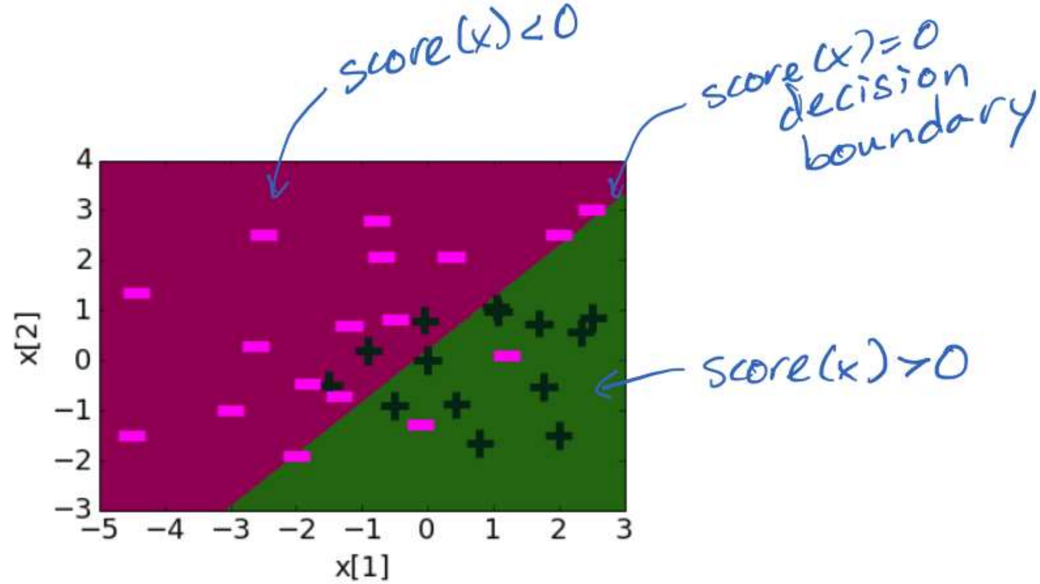
Models with less bias tend to need more data to learn well, but do better with sufficient data
偏差小的模型需要更多数据学习，但数据充足时做的更好



Learned decision boundary
学习得到的决策边界

Feature	Value	Coefficient learned
$h_0(\mathbf{x})$	1	0.23
$h_1(\mathbf{x})$	$\mathbf{x}[1]$	1.12
$h_2(\mathbf{x})$	$\mathbf{x}[2]$	-1.07

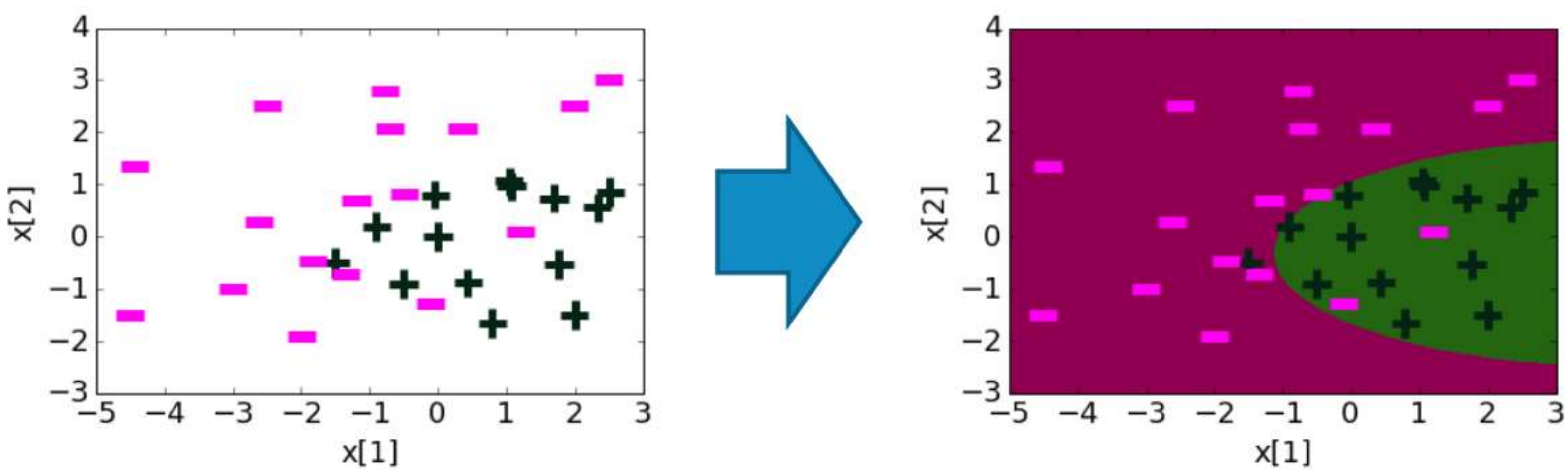




Quadratic features (in 2d)
二维平方特征

Feature	Value	Coefficient learned
$h_0(\mathbf{x})$	1	1.68
$h_1(\mathbf{x})$	$\mathbf{x}[1]$	1.39
$h_2(\mathbf{x})$	$\mathbf{x}[2]$	-0.59
$h_3(\mathbf{x})$	$(\mathbf{x}[1])^2$	-0.17
$h_4(\mathbf{x})$	$(\mathbf{x}[2])^2$	-0.96

Note: we are not including cross terms for simplicity

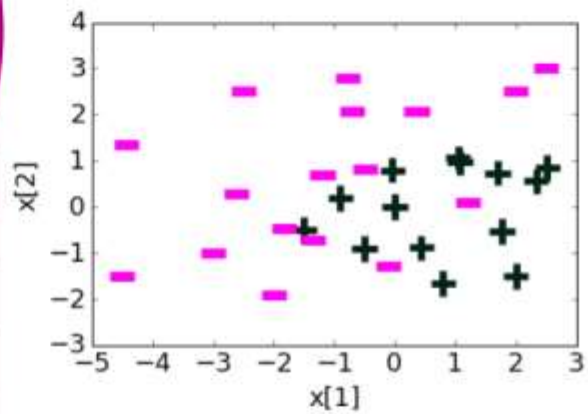


Degree 6 features (in 2d)
二维6次方特征

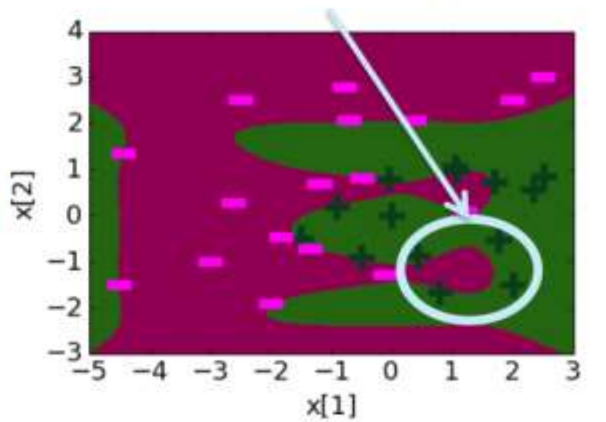
Note: we are not including cross terms for simplicity

Feature	Value	Coefficient learned
$h_0(\mathbf{x})$	1	21.6
$h_1(\mathbf{x})$	$\mathbf{x}[1]$	5.3
$h_2(\mathbf{x})$	$\mathbf{x}[2]$	-42.7
$h_3(\mathbf{x})$	$(\mathbf{x}[1])^2$	-15.9
$h_4(\mathbf{x})$	$(\mathbf{x}[2])^2$	-48.6
$h_5(\mathbf{x})$	$(\mathbf{x}[1])^3$	-11.0
$h_6(\mathbf{x})$	$(\mathbf{x}[2])^3$	67.0
$h_7(\mathbf{x})$	$(\mathbf{x}[1])^4$	1.5
$h_8(\mathbf{x})$	$(\mathbf{x}[2])^4$	48.0
$h_9(\mathbf{x})$	$(\mathbf{x}[1])^5$	4.4
$h_{10}(\mathbf{x})$	$(\mathbf{x}[2])^5$	-14.2
$h_{11}(\mathbf{x})$	$(\mathbf{x}[1])^6$	0.8
$h_{12}(\mathbf{x})$	$(\mathbf{x}[2])^6$	-8.6

Coefficient values getting large



Score($\mathbf{x}$) < 0

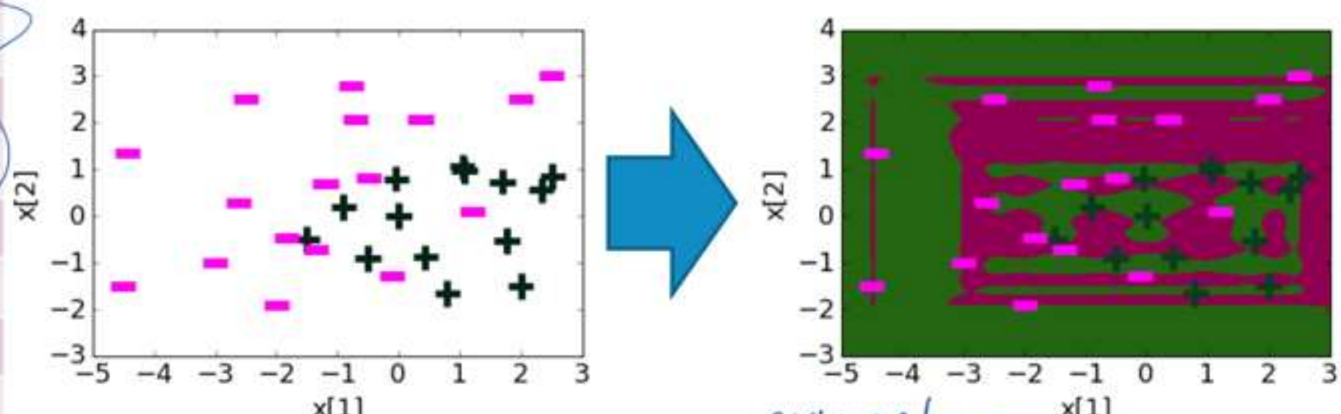


Degree 20 features (in 2d)
二维20次方特征

Feature	Value	Coefficient learned
$h_0(\mathbf{x})$	1	8.7
$h_1(\mathbf{x})$	$\mathbf{x}[1]$	5.1
$h_2(\mathbf{x})$	$\mathbf{x}[2]$	78.7
…	…	…
$h_{11}(\mathbf{x})$	$(\mathbf{x}[1])^6$	-7.5
$h_{12}(\mathbf{x})$	$(\mathbf{x}[2])^6$	**3803**
$h_{13}(\mathbf{x})$	$(\mathbf{x}[1])^7$	21.1
$h_{14}(\mathbf{x})$	$(\mathbf{x}[2])^7$	**-2406**
…	…	…
$h_{37}(\mathbf{x})$	$(\mathbf{x}[1])^{19}$	$-2*10^{-6}$
$h_{38}(\mathbf{x})$	$(\mathbf{x}[2])^{19}$	-0.15
$h_{39}(\mathbf{x})$	$(\mathbf{x}[1])^{20}$	$-2*10^{-8}$
$h_{40}(\mathbf{x})$	$(\mathbf{x}[2])^{20}$	0.03

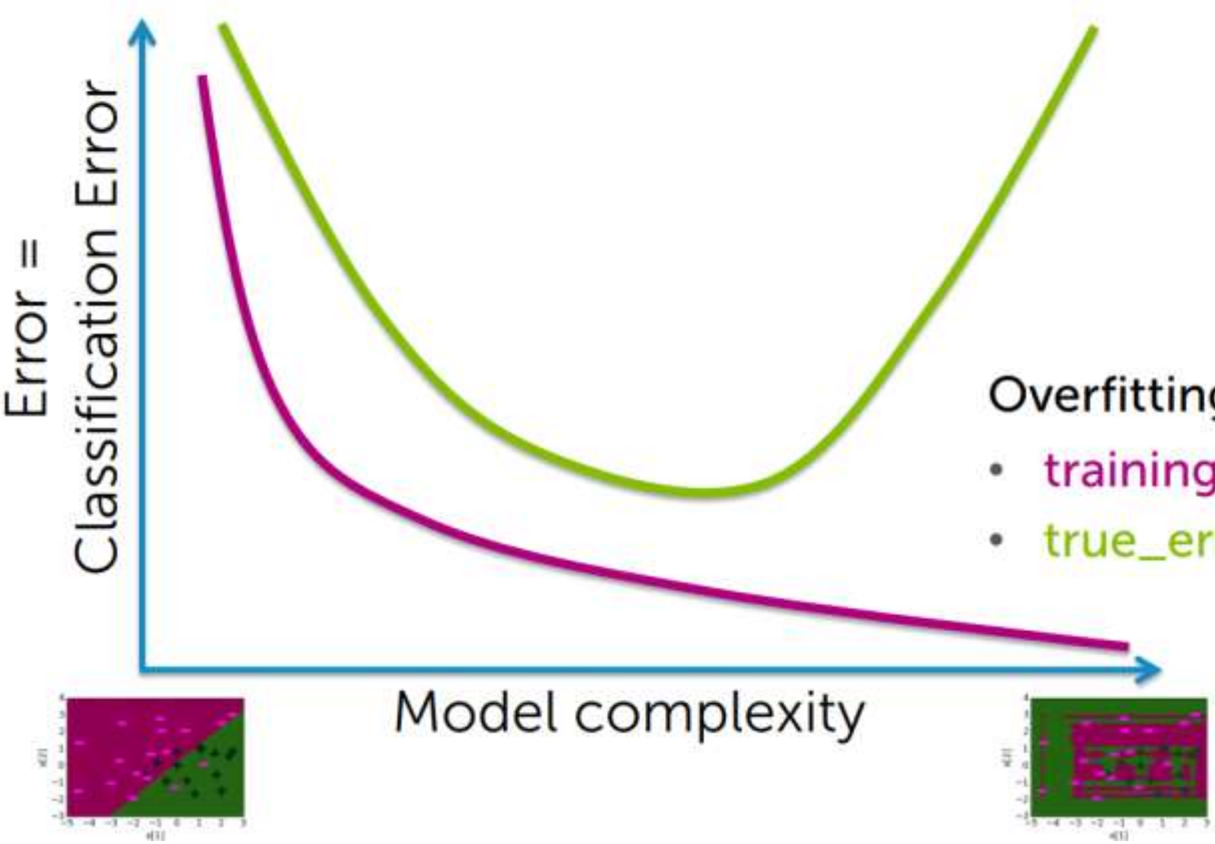
Note: we are not including cross terms for simplicity

Often, overfitting associated with very large estimated coefficients $\hat{\mathbf{w}}$



extremely complex decision boundary

Overfitting in classification
分类中的过拟合

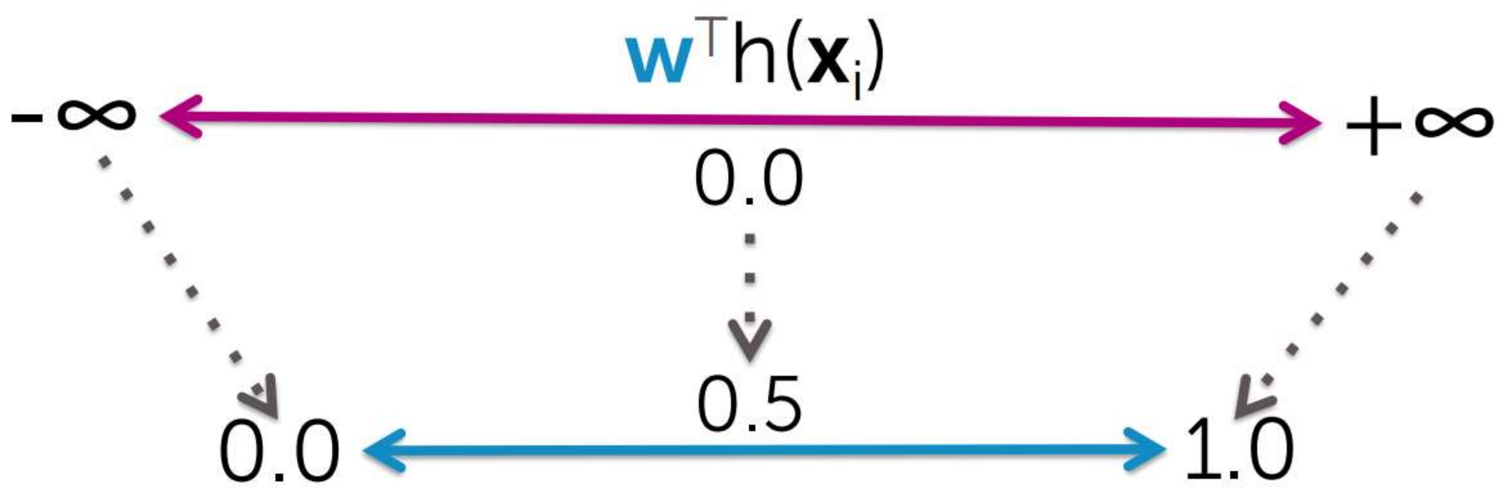


Overfitting if there exists $\mathbf{w}^*$:

- training_error($\mathbf{w}^*$) > training_error($\hat{\mathbf{w}}$)
- true_error($\mathbf{w}^*$) < true_error($\hat{\mathbf{w}}$)

Overfitting in classifiers->Overconfident predictions
分类器过拟合导致预测过于自信

Logistic regression model



$$P(y=+1|\mathbf{x}_i,\mathbf{w}) = \text{sigmoid}(\mathbf{w}^T h(\mathbf{x}_i))$$

Overfitting in classifiers->Overconfident predictions
过拟合的分类器->过于自信的预测

The subtle (negative) consequence of overfitting in logistic regression

Logistic回归过拟合的细微（负面）影响

Overfitting → Large coefficient values

↓

$\hat{\mathbf{w}}^T h(\mathbf{x}_i)$ is very positive (or very negative) → sigmoid($\hat{\mathbf{w}}^T h(\mathbf{x}_i)$) goes to 1 (or to 0)

↓

Model becomes extremely overconfident of predictions

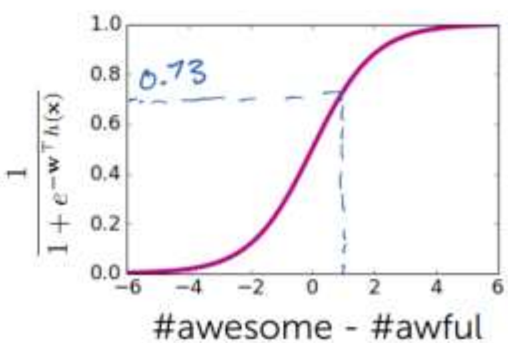
Effect of coefficients on logistic regression model
Logistic回归模型系数的影响

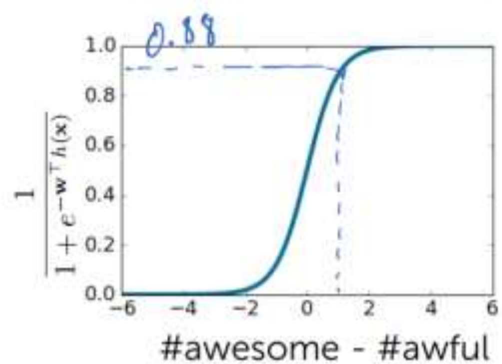
Input x: #awesome=2, #awful=1

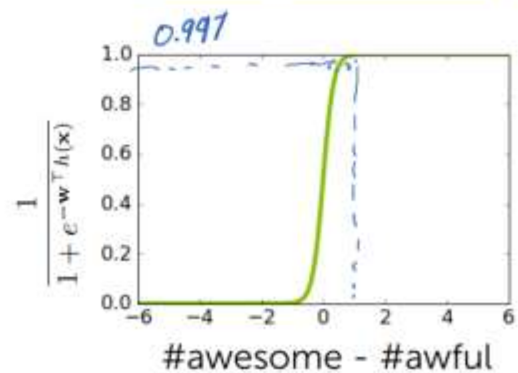
w_0	0
$w_{\#awesome}$	+1
$w_{\#awful}$	-1

w_0	0
$w_{\#awesome}$	+2
$w_{\#awful}$	-2

w_0	0
$w_{\#awesome}$	+6
$w_{\#awful}$	-6





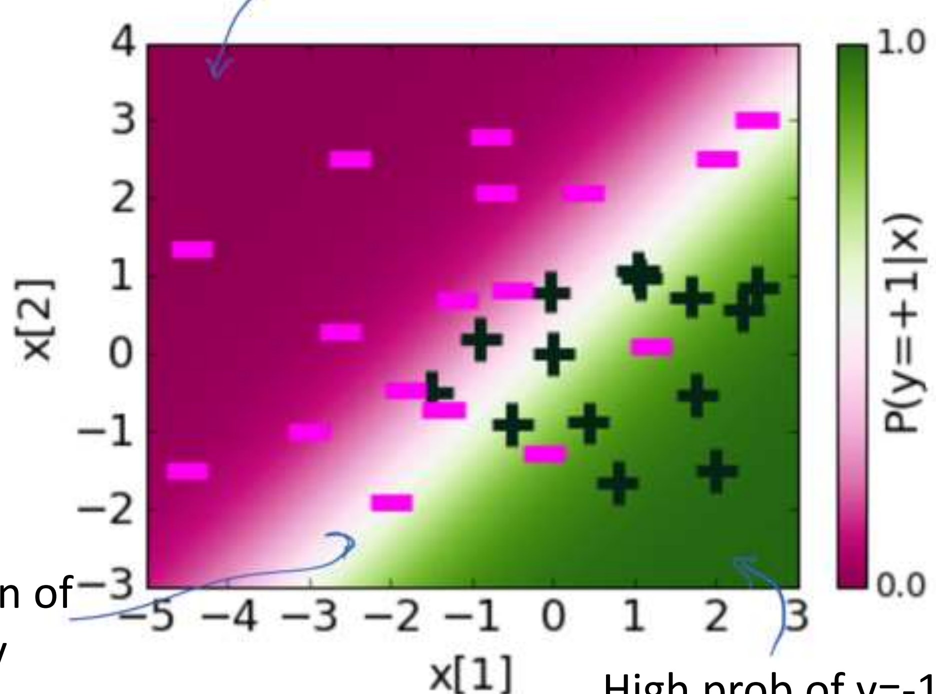


Learned probabilities
学习到的概率

Feature	Value	Coefficient learned
$h_0(\mathbf{x})$	1	0.23
$h_1(\mathbf{x})$	$\mathbf{x}[1]$	1.12
$h_2(\mathbf{x})$	$\mathbf{x}[2]$	-1.07

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

High prob of y=+1



Wide region of uncertainty

High prob of y=-1

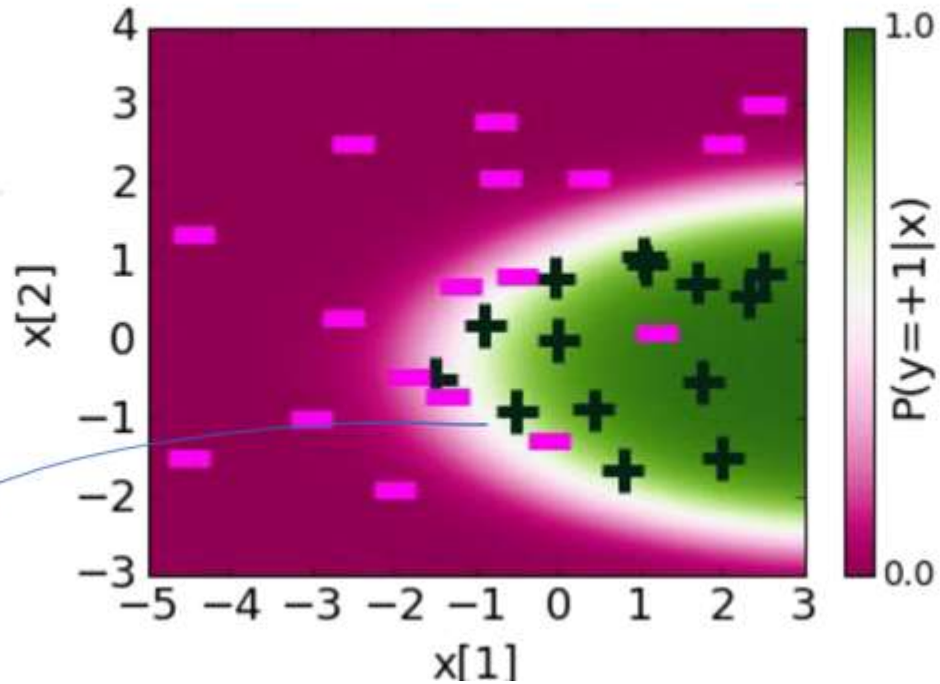
Quadratic features: Learned probabilities
二次特征：学习到的概率

更好拟合数据

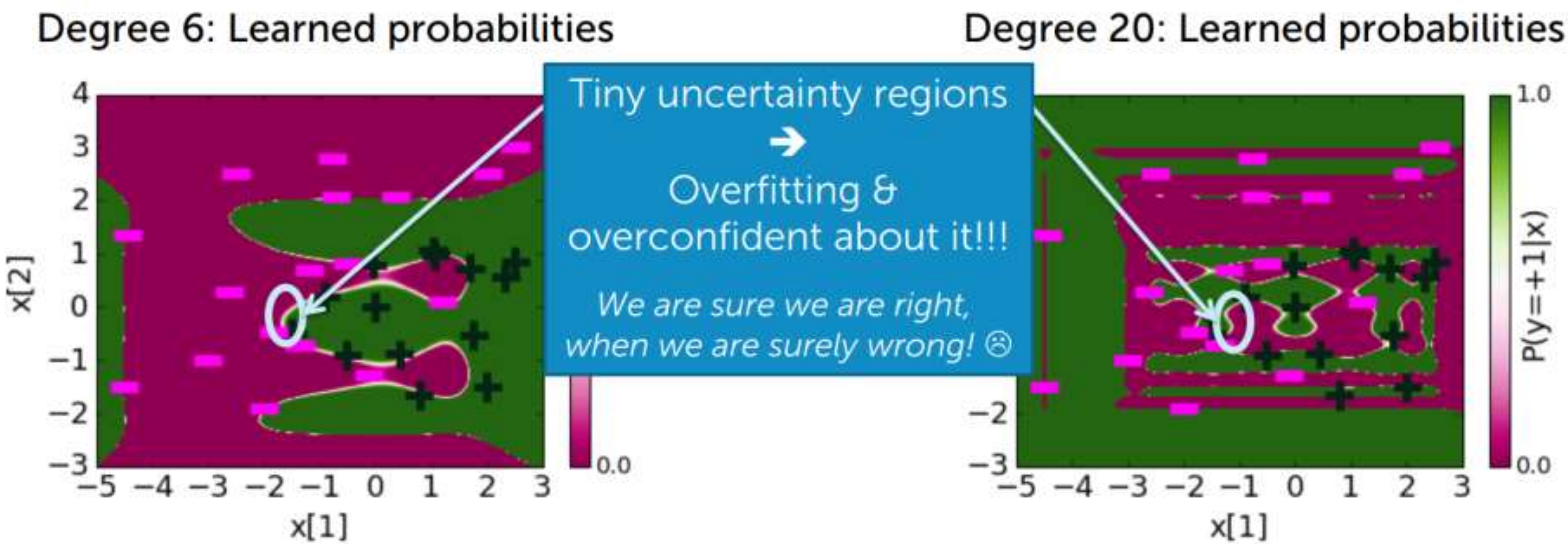
Feature	Value	Coefficient learned
$h_0(\mathbf{x})$	1	1.68
$h_1(\mathbf{x})$	$\mathbf{x}[1]$	1.39
$h_2(\mathbf{x})$	$\mathbf{x}[2]$	-0.58
$h_3(\mathbf{x})$	$(\mathbf{x}[1])^2$	-0.17
$h_4(\mathbf{x})$	$(\mathbf{x}[2])^2$	-0.96

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

Uncertainty region narrow S

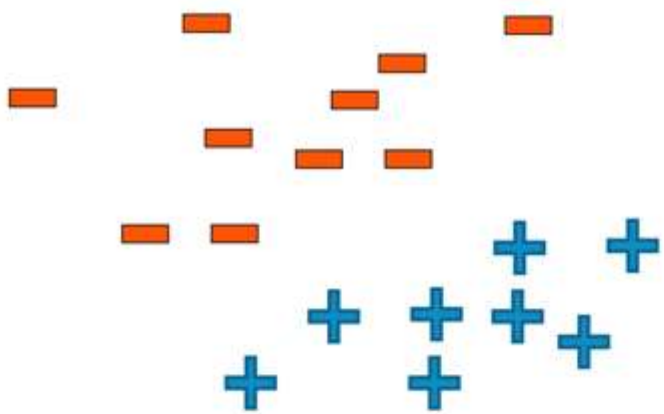


Overfitting -> Overconfident predictions
过拟合->过于自信的预测



Overfitting in logistic regression: Another perspective
Logistic回归的过拟合：另一个角度

Linearly-separable data　　线性可分数据



Note 1: If you are using D features, linear separability happens in a D-dimensional space

Note 2: If you have enough features, data are (almost) always linearly separable

Data are linearly separable if:

- There exist coefficients $\hat{\mathbf{w}}$ such that:
  - For all positive training data

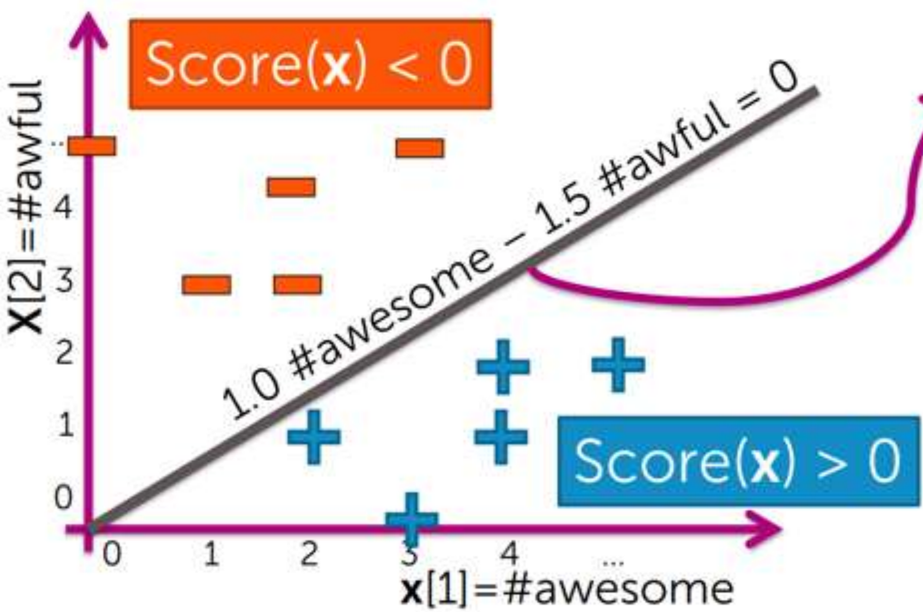
    $$score(x) = \hat{\mathrm{w}}^{\mathrm{T}} h(x) > 0$$

  - For all negative training data

    $$score(x) = \hat{\mathrm{w}}^{\mathrm{T}} h(x) < 0$$

training_error($\hat{\mathbf{w}}$) = 0

Effect of linear separability on coefficients
系数对线性可分性的影响



Data are linearly separable with $\hat{w}_1$=1.0 and $\hat{w}_2$=-1.5

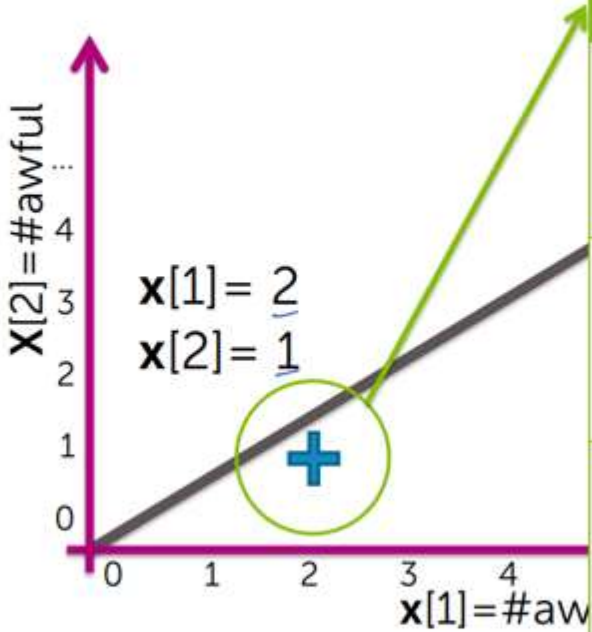
Data also linearly separable with $\hat{w}_1$=10 and $\hat{w}_2$=-15

Data also linearly separable with $\hat{w}_1=10^9$ and $\hat{w}_2=-1.5\text{x}10^9$

Effect of linear separability on coefficients

系数对线性可分性的影响

Maximum likelihood estimation (MLE) prefers most certain model ➔
Coefficients go to infinity for linearly-separable data!!!



$P(y=+1 \mid \mathbf{x}_i, \hat{w})$
$\frac{1}{1+e^{-0.5}} = 0.62$
$\frac{1}{1+e^{-5}} = 0.99$
$\frac{1}{1+e^{-5*10^8}} \approx 1$

Data is linearly separable with $\hat{w}_1$=1.0 and $\hat{w}_2$=-1.5

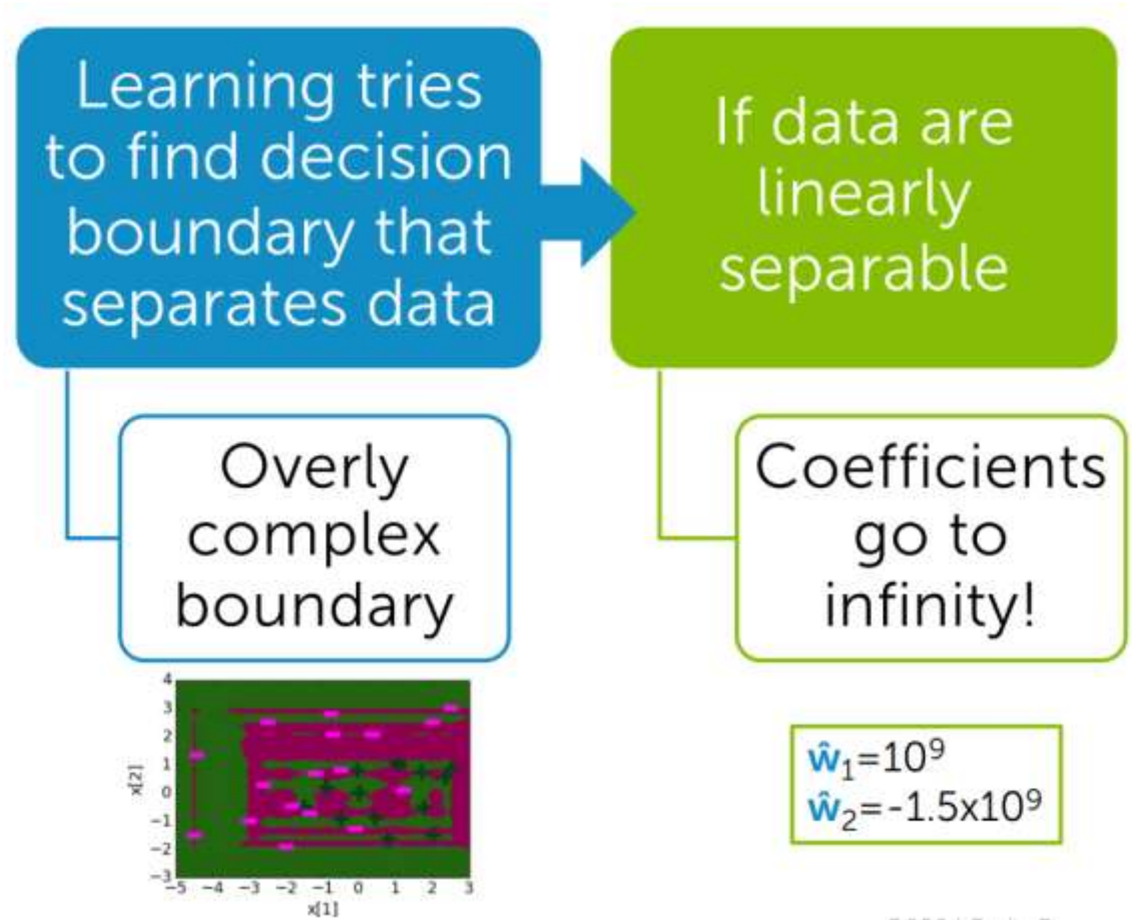
↓

Data also linearly separable with $\hat{w}_1$=10 and $\hat{w}_2$=-15

↓

Data also linearly separable with $\hat{w}_1$=10^9 and $\hat{w}_2$=-1.5x10^9

Overfitting in logistic regression is “twice as bad”
Logistic回归的过拟合是” twice as bad”



Penalizing large coefficients to mitigate overfitting

惩罚大系数，缓解过拟合

Desired total cost format
期望的总代价格式

Want to balance:

i. How well function fits data

ii. Magnitude of coefficients

want to balance

Total quality =

measure of fit - measure of magnitude of coefficients

(data likelihood)
large # = good fit to training data

large # = overfit

Regularization and Lagrange multiplier method
正则化与Lagrange乘子法

- 通过在模型中添加惩罚项或约束条件来控制模型复杂度，获得bias-variance trade-off
  - 可以减小线性回归的过度拟合和多重共线性等问题

岭回归和LASSO

具体来讲，岭回归和LASSO分别对应ℓ_2和ℓ_1正则化，对系数向量$\boldsymbol{w}$提出的先验假设分别为$\|\boldsymbol{w}\|_2 \leq C$和$\|\boldsymbol{w}\|_1 \leq C$，$C$为预先取定的常数. 也就是说，我们关注下面带约束的优化问题，对于岭回归

$$\min_{w} \quad \|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|_2^2 \qquad \text{s.t.} \quad \|\boldsymbol{w}\|_2 \leq C,$$

利用拉格朗日乘子法，以上约束优化问题等价于无约束的惩罚函数优化问题

$$\min_{w} \quad \|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2$$

其中正则化系数$\lambda > 0$是依赖于C的常数.

类似的，如果我们采用L_1正则化，则可获得**LASSO （Least Absolute Shrinkage and Section Operator）：**

$$\begin{aligned}\min_{w} \quad & \|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & \|\boldsymbol{w}\|_1 \leq C.\end{aligned} \;\Rightarrow\; \min_{w} \quad \frac{1}{2}\|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1,$$

L2 regularized logistic regression
L2正则化的Logistic回归

考虑以下目标:

Select $\hat{\mathbf{w}}$ to maximize:

$$\ell(\mathbf{w}) - \lambda||\mathbf{w}||_2^2$$

tuning parameter = balance of fit and magnitude

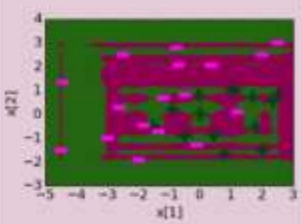
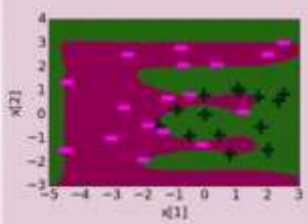
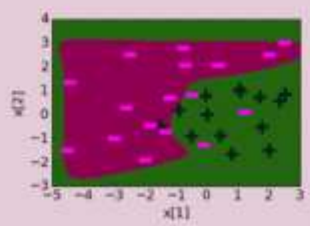
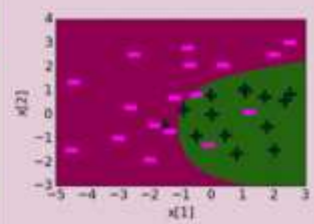
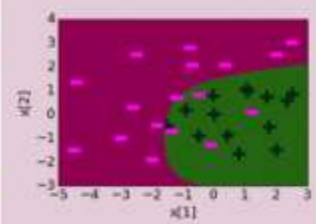
L_2 regularized logistic regression

Pick λ using:
- Validation set (for large datasets)
- Cross-validation (for smaller datasets)

(as in ridge/lasso regression)

L2 regularized logistic regression
L2正则化的Logistic回归

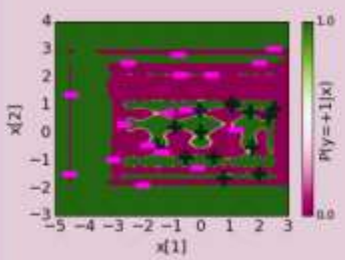
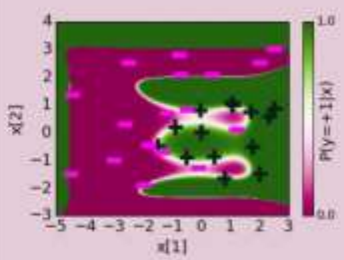
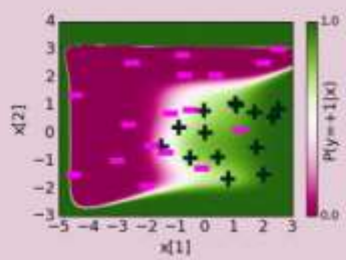
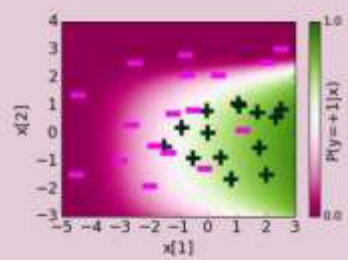
Degree 20 features, effect of regularization penalty λ
20次方特征下正则化惩罚λ的影响

Regularization	λ = 0	λ = 0.00001	λ = 0.001	λ = 1	λ = 10
Range of coefficients	-3170 to 3803	-8.04 to 12.14	-0.70 to 1.25	-0.13 to 0.57	-0.05 to 0.22
Decision boundary					

L2 regularized logistic regression
L2正则化的Logistic回归

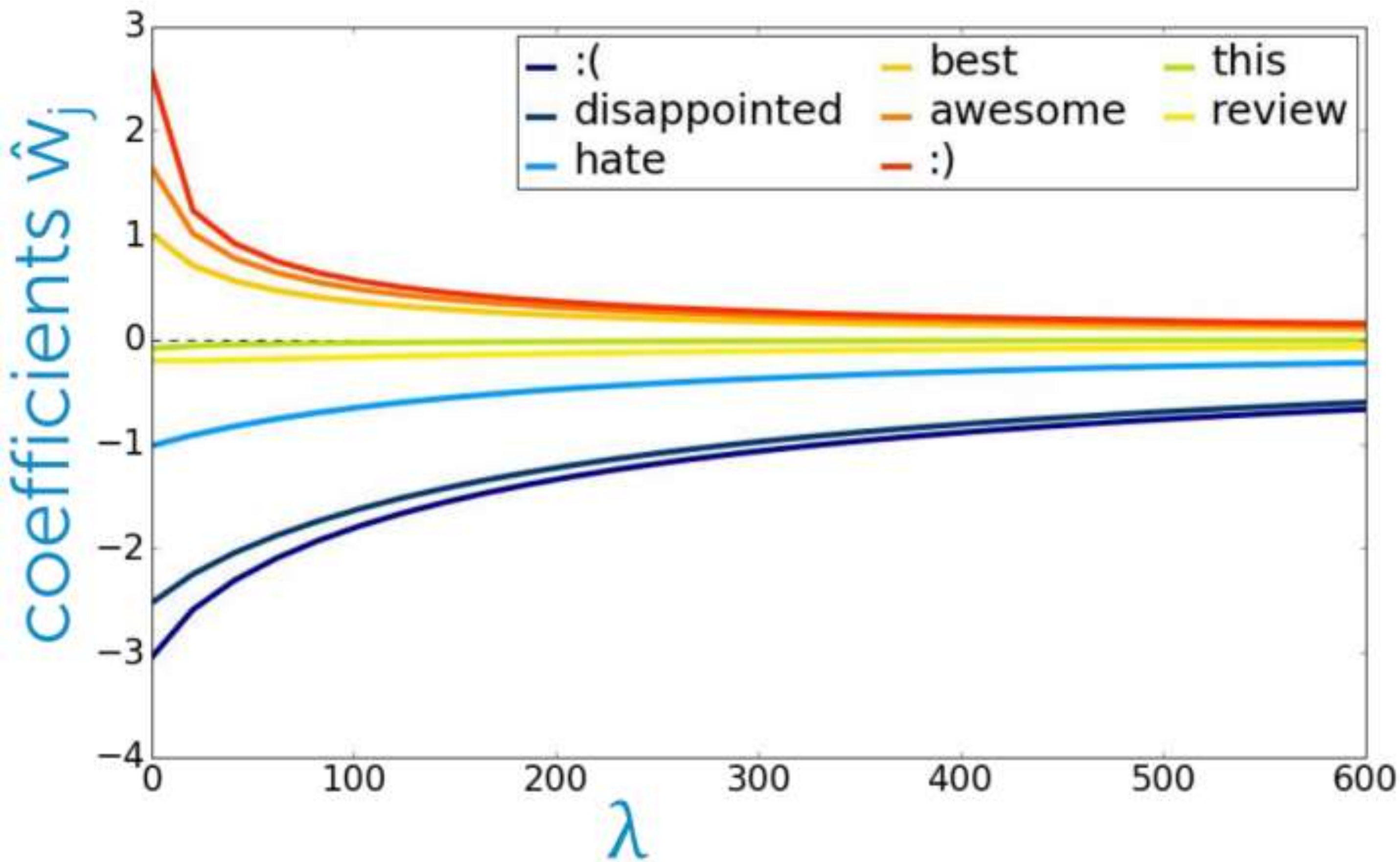
Degree 20 features: regularization reduces "overconfidence"

20次方特征下，正则化减缓了“过自信”的问题

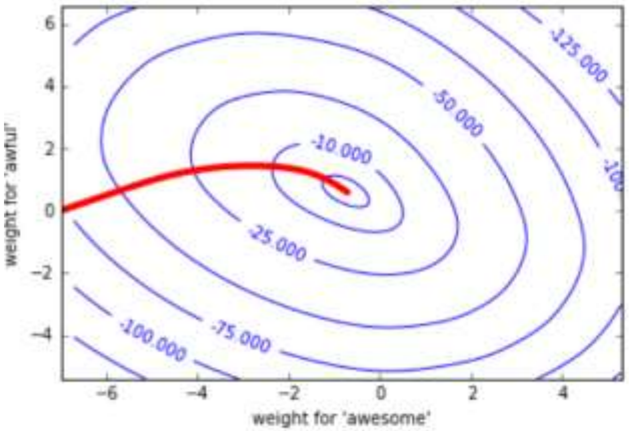
Regularization	λ = 0	λ = 0.00001	λ = 0.001	λ = 1
Range of coefficients	-3170 to 3803	-8.04 to 12.14	-0.70 to 1.25	-0.13 to 0.57
Learned probabilities				

L2 regularized logistic regression
L2正则化的Logistic回归

Coefficient path 系数路径



Gradient ascent for L2 regularized logistic regression
L2正则化Logistic回归的梯度上升



init $\mathbf{w}^{(1)}=0$ (or randomly, or smartly), $t=1$

while not converged:

for $j=0,\ldots,D$

$$\text{partial}[j] = \sum_{i=1}^{N} h_j(\mathbf{x}_i)\Big(\mathbb{1}[y_i=+1] - P(y=+1 \mid \mathbf{x}_i, \mathbf{w}^{(t)})\Big)$$

$$\mathbf{w}_j^{(t+1)} \leftarrow \mathbf{w}_j^{(t)} + \eta\,(\text{partial}[j] - \underline{2\lambda\,\mathbf{w}_j^{(t)}})$$

only change

$t \leftarrow t+1$

L1 regularized logistic regression
L1正则化的Logistic回归

Sparse logistic regression with L1 penalty
L1惩罚用于稀疏的Logistic回归

Select $\hat{\mathbf{w}}$ to maximize:

$$\ell(\mathbf{w}) - \lambda||\mathbf{w}||_1$$

tuning parameter = balance of fit and magnitude

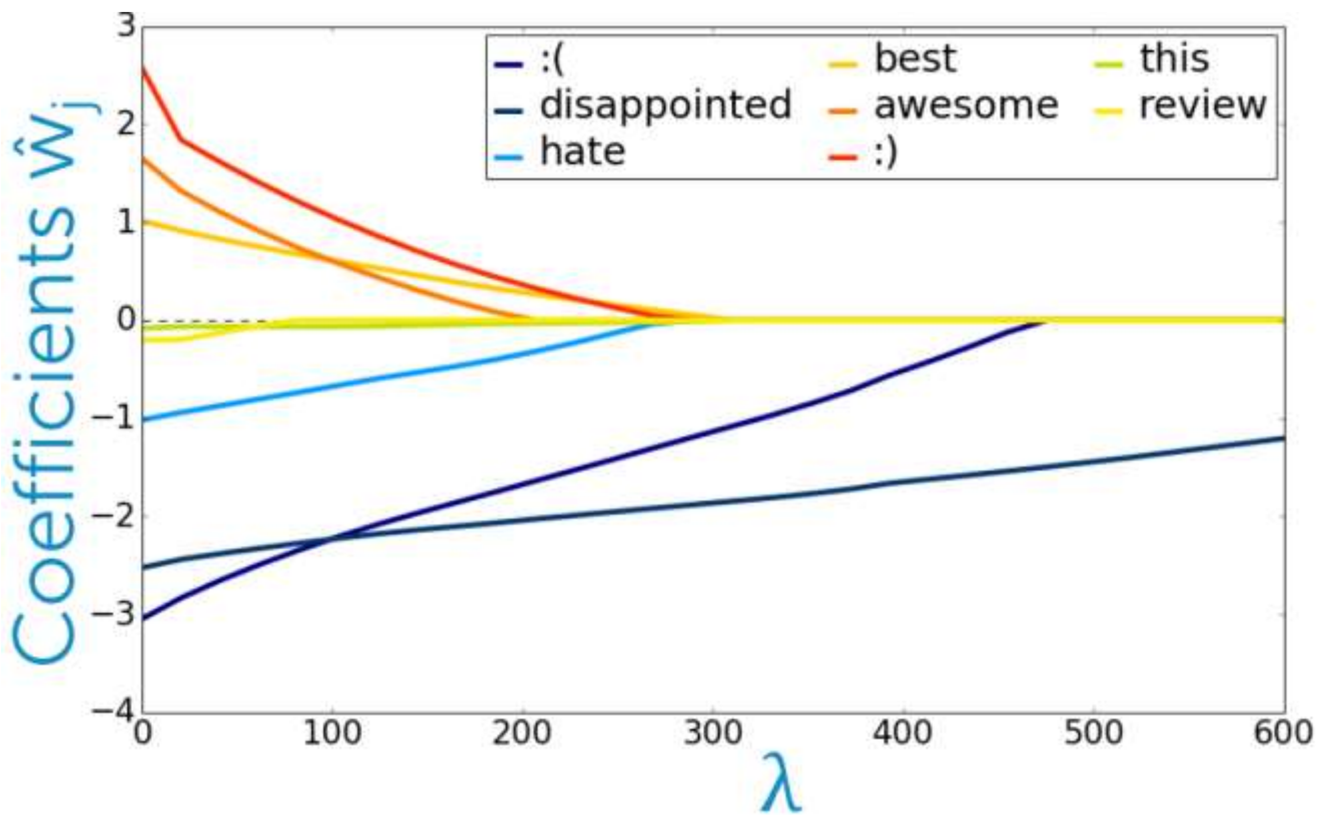
L_1 regularized logistic regression

Pick λ using:
- Validation set (for large datasets)
- Cross-validation (for smaller datasets)

(as in ridge/lasso regression)

L1 regularized logistic regression
L1正则化的Logistic回归

Coefficient path – L1 penalty　　系数路径 — L1惩罚

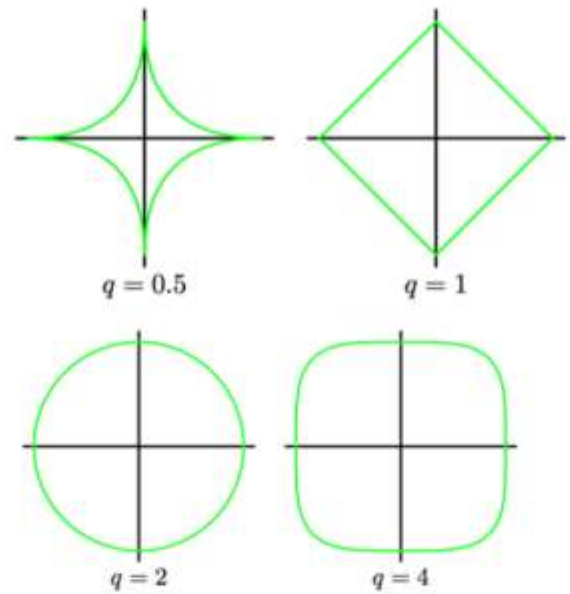


General form of regularization
正则化通用形式

- L_q 正则化的通用形式：

$$\sum_{i=1}^{n}(y_1 - w^T x_i)^2 + \lambda||w||^q$$

- q=2：岭回归(Ridge Regression) ⇔ loss + L_2 惩罚项
- q=1：LASSO ⇔ loss + L_1 惩罚项

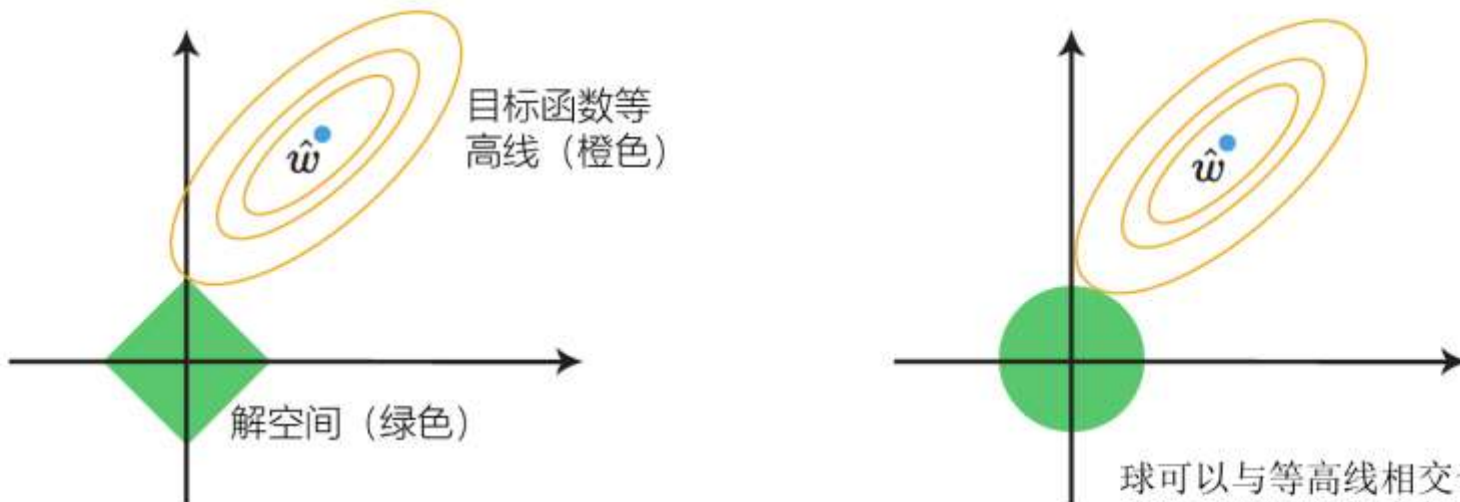


$||w||^q$=1示意图

Why does LASSO produce sparse solutions
为什么LASSO产生稀疏解

最优解：发生在目标函数的等高线和可行区域的交集处.

LASSO　　　　　　　　　　　　岭回归



角比边更有可能与等高线相交，这个现象在高维的情况下尤其明显，因为高维的角更加“凸出”=> 产生稀疏解

球可以与等高线相交于任何一点，获得稀疏解的概率低

Summary of overfitting in logistic regression
Logistic回归过拟合的总结

Describe symptoms and effects of overfitting in classification

- Identify when overfitting is happening
- Relate large learned coefficients to overfitting
- Describe the impact of overfitting on decision boundaries and predicted probabilities of linear classifiers

Summary of overfitting in logistic regression
Logistic回归过拟合的总结

Use regularization to mitigate overfitting

- Motivate the form of L2 regularized logistic regression quality metric
- Describe the use of L1 regularization to obtain sparse logistic regression solutions
- Describe what happens to estimated coefficients as tuning parameter λ is varied
- Estimate L2 regularized logistic regression coefficients using gradient ascent
- Interpret coefficient path plot