武漢大學
WUHAN UNIVERSITY

自然语言处理基础

文本分类

2019.10.31

自强、弘毅、求是、拓新

情感分析-朴素贝叶斯算法

瓜蒂	形状	颜色	类别
脱落	圆形	深绿	熟
未脱	尖形	浅绿	生
未脱	圆形	浅绿	生
脱落	尖形	青色	熟
脱落	圆形	浅绿	熟
未脱	尖形	青色	生
脱落	尖形	深绿	熟
未脱	圆形	青色	熟
脱落	尖形	浅绿	生
未脱	圆形	深绿	熟

问题：瓜蒂脱落、形状圆形、颜色青色，判断生还是熟?

$$P(A_i|B) = ?$$

Ai代表生、熟，B代表给出的瓜的特征集合。

情感分析-朴素贝叶斯算法

问题：瓜蒂脱落、形状圆形、颜色青色，判断生还是熟?

A1代表瓜生，B代表特征(脱落，圆形，青色)，$P(A_i|B) = ?$

贝叶斯定理：$P(A_i|B) = \frac{P(B|Ai)\,P(A_i)}{\sum_j P(B|Aj)P(Aj)}$

瓜生 P(A1|B) = ?

瓜熟 P(A2|B) = ?

$$P(A_i|B) \propto P(B|Ai)P(Ai)$$

条件独立性假设：

$$P(\mathrm{B}|Ai) = P(B_0|Ai)P(B_1|Ai)P(B_2|Ai)$$

可得： $P(A_i|B) \propto P(Ai)\prod_{k=1} P(B_\mathrm{k}|Ai)$

瓜蒂	形状	颜色	类别
脱落	圆形	深绿	熟
未脱	尖形	浅绿	生
未脱	圆形	浅绿	生
脱落	尖形	青色	熟
脱落	圆形	浅绿	熟
未脱	尖形	青色	生
脱落	尖形	深绿	熟
未脱	圆形	青色	熟
脱落	尖形	浅绿	生
未脱	圆形	深绿	熟

情感分析-朴素贝叶斯算法

瓜生 P(A1|B) = ?

瓜熟 P(A2|B) = ?

问题：瓜蒂脱落、形状圆形、颜色青色，判断生还是熟?

$P(B_0|A_1) = 1/4$ $P(B_1|A_1) = 1/4$ $P(B_0|A_1) = 1/4$ $P(A_1) = 2/5$

$P(B_0|A_2) = 2/3$ $P(B_1|A_2) = 2/3$ $P(B_0|A_2) = 1/3$ $P(A_2) = 3/5$

$$P(A_1|B) = 0.25^3 * 0.4 = 0.00625$$

$$P(A_2|B) = 0.67 * 0.67 * 0.33 * 0.6 = 0.08889$$

$P(A_1|B) < P(A_0|B)$ 所以瓜熟可能性较大

瓜蒂	形状	颜色	类别
脱落	圆形	深绿	熟
未脱	尖形	浅绿	生
未脱	圆形	浅绿	生
脱落	尖形	青色	熟
脱落	圆形	浅绿	熟
未脱	尖形	青色	生
脱落	尖形	深绿	熟
未脱	圆形	青色	熟
脱落	尖形	浅绿	生
未脱	圆形	深绿	熟

情感分析-朴素贝叶斯算法

武漢大學 WUHAN UNIVERSITY

Text	Class	Doc	
Chinese Beijing Chinese	ZH	1	Train set
Chinese Chinese Shanghai	ZH	2	
Chinese Macao	ZH	3	
California LA Chinese	US	4	
Chinese Chinese California LA	?	5	Test set

$$P(ZH|B) = P(B_0|ZH)P(B_1|ZH)P(B_2|ZH) \ldots P(B_3|ZH)P(ZH)$$

$P(B_0|ZH) = 5/8$， $P(B_1|ZH) = 5/8$， $P(B_2|ZH) = 0/8\ldots?$

B0 Chinese, B1 Chinese

B2 California B3 LA

情感分析-朴素贝叶斯算法

Text	Class	Doc	
Chinese Beijing Chinese	ZH	1	Train set
Chinese Chinese Shanghai	ZH	2	
Chinese Macao	ZH	3	
California LA Chinese	US	4	
Chinese Chinese California LA	?	5	Test set

$$P(B_0|\mathrm{A}_i) = \log\left(\frac{num\ of\ B_0\ in\ A_i + 1}{num\ of\ A_i + Total\ num}\right)$$

平滑处理

情感分析-朴素贝叶斯算法

Text	Class	Doc	
Chinese Beijing Chinese	ZH	1	Train set
Chinese Chinese Shanghai	ZH	2	
Chinese Macao	ZH	3	
California LA Chinese	US	4	
Chinese Chinese California LA	?	5	Test set

$$P(ZH|B) = P(B_0|ZH)P(B_1|ZH)P(B_2|ZH) \ldots P(B_3|ZH)P(ZH)$$

$$P(B_0|ZH) = \log(\frac{5+1}{8+3}), \quad P(B_1|ZH) = \log(\frac{5+1}{8+3}), \quad P(B_2|ZH) = \log(\frac{1}{8+3}) \ldots ?$$

B0 Chinese, B1 Chinese

B2 California B3 LA

情感分析

输入：章子怡宣布了二胎喜讯。

输出：情感倾向，正面 | 中性 | 负面

第一步：分词

→ 章子怡 宣布 了 二胎 喜讯 。

中文可用jieba 实现分词，英文直接按照空格切分

情感分析

特征提取：

词袋法：

S1　不　知道　你　在　说　什么　。

S2　我　就　知道　你　不　知道　。

词表：	不	就	你	什么	我	说	知道	在	。
S1	[1	0	1	1	0	1	1	1	1]
S2	[1	1	1	0	1	0	2	0	1]

情感分析-特征提取

TF-IDF:

TF- term frequency： $tf(x,w)=\frac{\text{单词x在文章w中出现的次数}}{\text{文章w中包含的单词个数}}$

	不	就	你	什么	我	说	知道	在	。
doc1	[1	0	1	1	0	1	1	1	1]
doc2	[1	1	1	0	1	0	2	0	1]

tf(不，s1) = 1/7， tf(就，s1) = 0/ 7 … tf(知道，s1) = 1/7

tf(不，s2) = 1/7, tf(就，s2) = 1/ 7 … tf(知道，s2) = 2/7

情感分析-特征提取

TF-IDF:

inverse doc frequency： $\mathrm{idf}(x) = \log\left(\frac{\text{文章总数}+1}{\text{包含}x\text{的文章个数}+1}\right)$

	不	就	你	什么	我	说	知道	在	。
doc1	[1	0	1	1	0	1	1	1	1]
doc2	[1	1	1	0	1	0	2	0	1]

idf(不) =log(3/3) = 0, idf(就) = log(3/2)=0.176 … idf(知道) = log(3/3) = 0

情感分析-特征提取

TF-IDF:

	不	就	你	什么	我	说	知道	在	。
doc1	[1	0	1	1	0	1	1	1	1]
doc2	[1	1	1	0	1	0	2	0	1]

$$tf\text{-}idf(x,w) = tf(x,w) * idf(x)$$

Doc1 = [0, 0, 0, 0.025, 0, 0.025, 0, 0.025, 0]

Doc2 = [0, 0.025, 0, 0, 0.025, 0, 0 ,0 , 0]

Logistics Regression

$$y = \frac{1}{e^{-(Wx+\mathrm{b})} + 1},$$

$$\mathrm{W} \in (1 \ * N), x \in (N,), b \in (1,), y \in (1,)$$

x为输入特征(N维向量，每一维度都是浮点数)，y为标签(一般取值(0，1))

W, b为模型参数

Logistics Regression

$$y = \frac{1}{e^{-(Wx+\mathrm{b})} + 1},$$

$$e^{-(Wx+\mathrm{b})} \in (+\infty, 0)$$

Logistics Regression

$$y = \frac{1}{e^{-(Wx+\mathrm{b})} + 1},$$

$$e^{-(Wx+\mathrm{b})} \in (+\infty, 0)$$

$$e^{-(Wx+\mathrm{b})} + 1 \in (+\infty, 1)$$

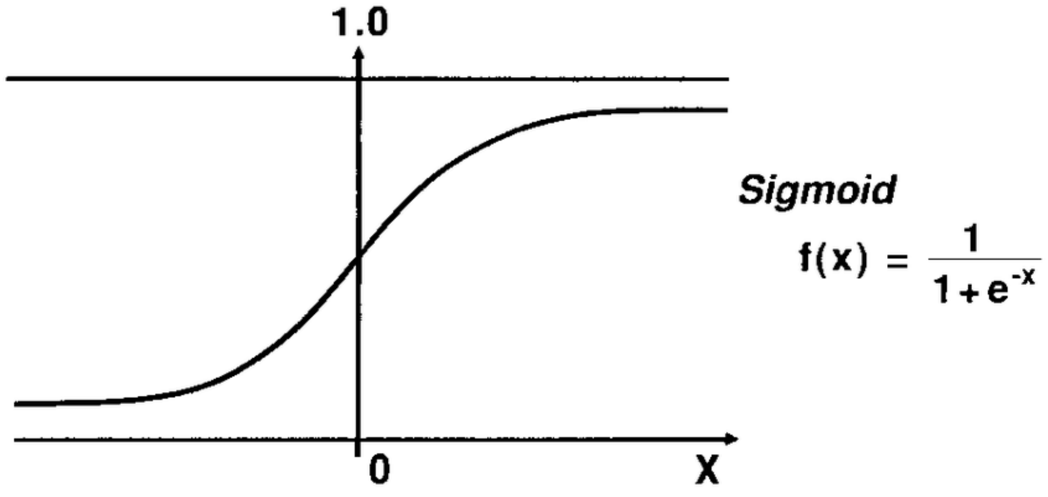
Logistics Regression

$$y = \frac{1}{e^{-(Wx+\mathrm{b})} + 1},$$

$$e^{-(Wx+\mathrm{b})} \in (+\infty, 0)$$

$$e^{-(Wx+\mathrm{b})} + 1 \in (+\infty, 1)$$

$$\mathrm{y} = \frac{1}{e^{-(Wx+\mathrm{b})}+1} \in (0,1)$$



输入x， 如果y > 0.5, 结果为正例，反之为负例

机器学习-梯度下降法

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

机器学习-梯度下降法

$$y = \frac{1}{e^{-(Wx+\mathrm{b})} + 1},$$

$(x0, y0)$为一个样本，$\hat{y}$为模型的输出结果，

损失函数$Loss = (y_0 - \hat{y})^2$

也可以用 交叉熵$Loss = -\sum_{\mathrm{k}=1}^{N} p_k log q_k$ ，当$\mathrm{p_k} = q_k$时，Loss最小

$\mathrm{p_k}$为真实标签分布，q_k为预测的结果分布

机器学习-梯度下降法

$$y = \frac{1}{e^{-(Wx+\mathrm{b})}+1},$$

(xi, yi)为一个样本，$\hat{y}$为模型的输出结果，

损失函数$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i log(\hat{y}_i) + (1-y_i)log(1-\hat{y}_i) = -\frac{1}{N}\sum_{i=1}^{N}(y_i w x_i - log(1+e^{wx_i}))$

$\mathrm{Loss} = \mathrm{f(w)}$ $\rightarrow$ $\min_{w \in R} f(w)$

$$f(w) = f(w_0) + f'(w_0)(\mathrm{w} - w_0) + \frac{1}{2} f''(w_0)(w - w_0)^2 + o(w^2)$$

机器学习-梯度下降法

$$y=\frac{1}{e^{-(Wx+\mathrm{b})}+1},$$

(xi,yi)为一个样本，$\hat{y}$为模型的输出结果，

损失函数$Loss=-\frac{1}{N}\sum_{i=1}^{N}(y_i w x_i-log(1+e^{wx_i})$

$\mathrm{Loss}=\mathrm{f(w)}$ → $\min\limits_{w\in R} f(w)$

$$f(w)=f(w_0)+f'(w_0)(w-w_0)+o(w)$$

机器学习-梯度下降法

$$y = \frac{1}{e^{-(Wx+\mathrm{b})}+1},$$

(xi, yi)为一个样本，$\hat{y}$为模型的输出结果，

损失函数$Loss = -\frac{1}{N}\sum_{i=1}^{N}(y_i w x_i - log(1+e^{wx_i})$

$\mathrm{Loss} = \mathrm{f(w)} \quad \rightarrow \quad \min_{w \in R} f(w)$

$$f(w) = f(w_0) + f'(w_0)(w-w_0) + o(w) \approx f(w_0) + f'(w_0)(w-w_0)$$

$$= f(w_0) + f'(w_0)\Delta w \ \ldots . \Delta w = (w-w_0)$$

$$f(w) = f(w_0) + f'(w_0)\Delta w$$

机器学习-梯度下降法

$$y=\frac{1}{e^{-(Wx+\mathrm{b})}+1},$$

(xi, yi)为一个样本，$\hat{y}$为模型的输出结果，

损失函数$Loss=-\frac{1}{N}\sum_{i=1}^{N}(y_i wx_i-log(1+e^{wx_i})$

$\mathrm{Loss}=\mathrm{f(w)}$ → $\min\limits_{w\in R} f(w)$

$$f(w)=f(w_0)+f'(w_0)(w-w_0)+o(w)\approx f(w_0)+f'(w_0)(w-w_0)$$

$$=f(w_0)+f'(w_0)\Delta w \ldots.\Delta w=(w-w_0)$$

$$f(w)=f(w_0)+f'(w_0)\Delta w$$

要使$f(w)<f(w_0)$, 令 $\Delta w=-\alpha f'(w_0)$

机器学习-梯度下降法

$$y = \frac{1}{e^{-(Wx+\mathrm{b})}+1},$$

(xi, yi)为一个样本，$\hat{y}$为模型的输出结果，

损失函数$Loss = -\frac{1}{N}\sum_{i=1}^{N}(y_i w x_i - log(1+e^{wx_i})$

$$\mathrm{Loss} = \mathrm{f(w)} \quad \rightarrow \quad \min_{w \in R} f(w)$$

$$f(w) = f(w_0) + f'(w_0)(w-w_0) + o(w) \approx f(w_0) + f'(w_0)(w-w_0)$$

$$= f(w_0) + f'(w_0)\Delta w \ldots . \Delta w = (w - w_0)$$

$$f(w) = f(w_0) + f'(w_0)\Delta w$$

要使$f(w) < f(w_0)$, 令 $\Delta w = -\alpha f'(w_0)$

则有：$f(w) = f(w_0) - \alpha(f'(w_0))^2$

机器学习-梯度下降法

$$y = \frac{1}{e^{-(Wx+\mathrm{b})}+1},$$

(xi, yi)为一个样本，$\hat{y}$为模型的输出结果，

损失函数$Loss = -\frac{1}{N}\sum_{i=1}^{N}(y_i w x_i - log(1+e^{wx_i})$

$\mathrm{Loss} = \mathrm{f(w)}$ → $\min\limits_{w \in R} f(w)$

$$f(w) = f(w_0) + f'(w_0)(w - w_0) + o(w) \approx f(w_0) + f'(w_0)(w - w_0)$$

$$= f(w_0) + f'(w_0)\Delta w \dots . \Delta w = (w - w_0)$$

$$f(w) = f(w_0) + f'(w_0)\Delta w$$

要使$f(w) < f(w_0)$, 令 $\Delta w = -\alpha f'(w_0)$

则有：$f(w) = f(w_0) - (f'(w_0))^2$

所以，只要$w - w_0 = -\alpha f'(w_0)$ 则 $w = w_0 - \alpha f'(w_0)$ 即可每次让f(w)更小

机器学习-梯度下降法

$$y = \frac{1}{e^{-(Wx+\mathrm{b})} + 1},$$

(xi, yi)为一个样本，$\hat{y}$为模型的输出结果，

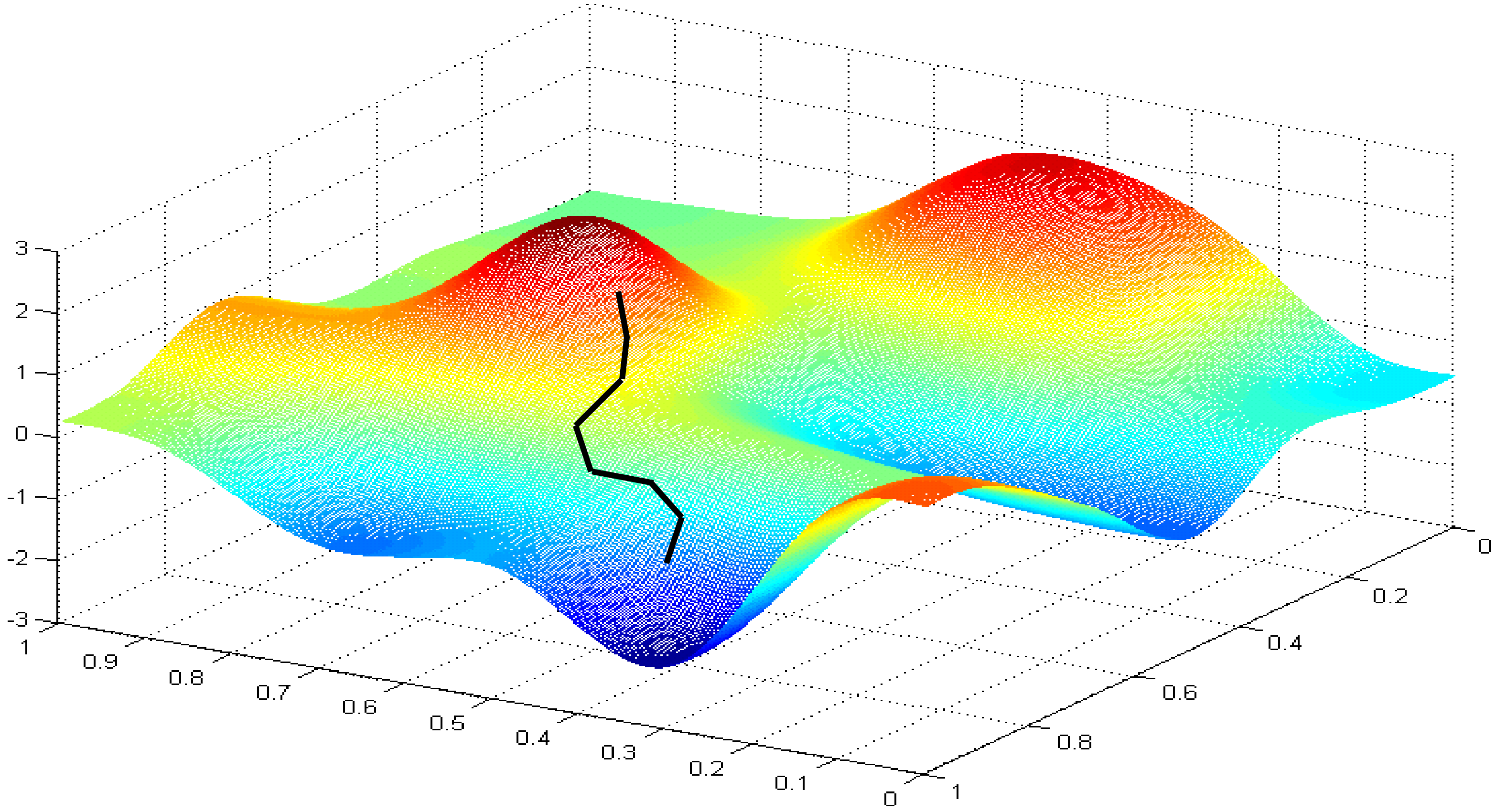
损失函数$Loss = -\frac{1}{N}\sum_{i=1}^{N}(y_i w x_i - log(1 + e^{w x_i})$

$\mathrm{Loss} = \mathrm{f(w)} \quad \rightarrow \quad \min_{w \in R} f(w)$

则 $w = w_0 - \alpha f'(w_0)$ 即可每次让f(w)更小

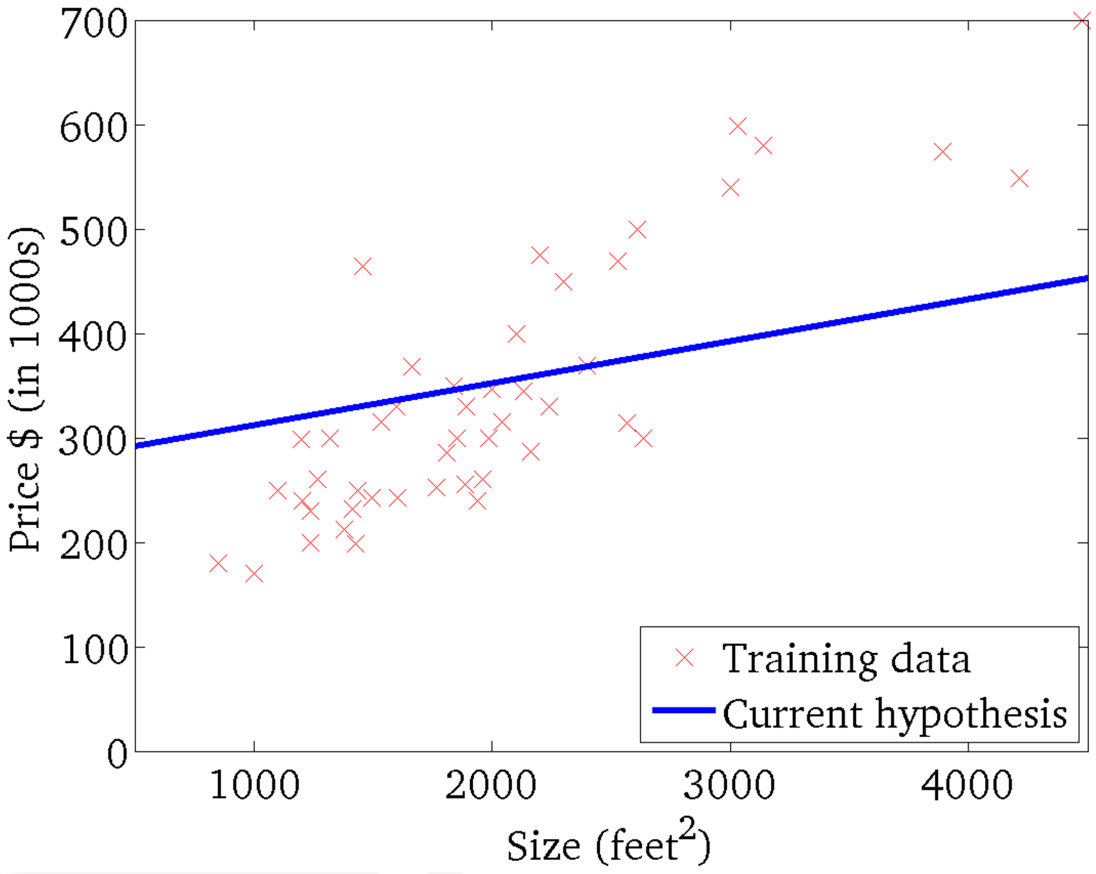
$$f'(w) = \frac{\partial \mathrm{Loss}}{\partial \mathrm{w}} = \frac{\partial(-\frac{1}{N}\sum_{i=1}^{N}(y_i w x_i - log(1 + e^{w x_i}))}{\partial w} = -\frac{1}{N}\sum_{i=1}^{N}(x_i(y_i - \hat{y}_i))$$

机器学习-梯度下降法



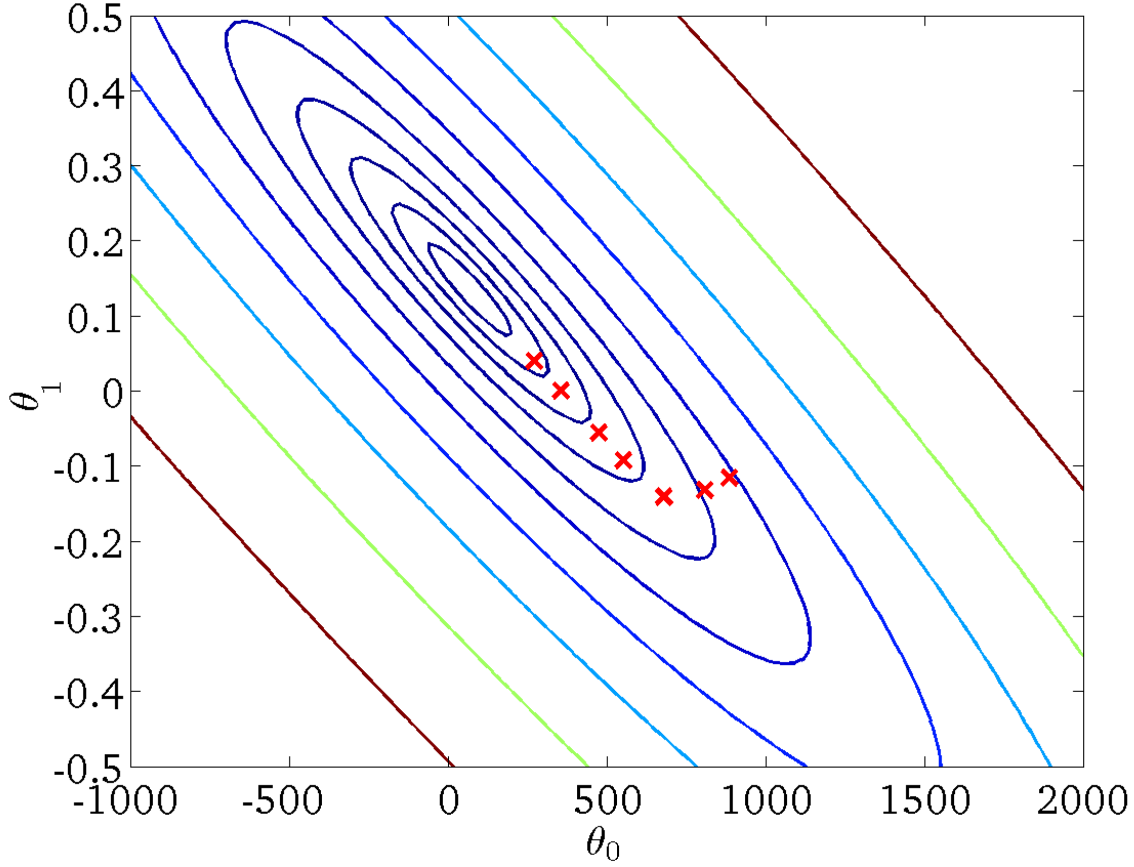
$$h_\theta(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



LogisticsRegression-梯度下降法

机器学习（Logistics Regression）进行文本分类：

Step1:数据 （文本，标签）→向量化→（X, y）

Step2:选择模型： $y=\frac{1}{e^{-(wx+\mathrm{b})}+1}$

Stop 为False

While Stop is False：

计算将(x_i, y_i)代入计算$\hat{y}_k=\frac{1}{e^{-(Wx_i+\mathrm{b})}+1}$

得到损失： $Loss$

求导： $\mathrm{f}'(w_k)=\frac{\partial \mathrm{Loss}}{\partial w_k}$

优化： $w_{k+1}=w_k-\alpha f'(w_\mathrm{k})$

如果$|f(w_{k+1})-f(w_k)|<\varepsilon$

set Stop True

否则： $\mathrm{k}=\mathrm{k}+1$

最终的到模型参数(W_{k+1}, b)

示例Demo


```
import ...

"""..."""

def read_train_valid(filname):...

def read_test(filename):...

def split_text(text_data):...

def vectorizer(train_data, valid_data, test_data):...

def train_valid(train_data, train_label, valid_data, valid_label):...

def predict(mode, test_data):...

def run_step():
    """
    选择相应的任务和文件
    读文件， train_data, train_label = some_function(filename='')
               valid_data, validlabel = some_function(filename='')
              test_data, test_ids = some_function(filename='')
    将原始文本分词：
             train_data = split_function(train_data)
             valid_data = split_function(valid_data)
             test_data = split_function(test_data)
    将分词后的文本变成向量：
```

主要内容

参考资料

- 李航 《统计学习方法》
- 周志华《机器学习》

参考工具

- Scikit-Learn(机器学习包)
  + Logistics Regression
  + Naïve Bayes
- Jieba分词

武漢大學
WUHAN UNIVERSITY

谢谢大家

T h a n k s f o r Y o u r A t t e n t i o n

自强、弘毅、求是、拓新