考试时间：2018.12.27 7:00 PM-9:00 PM

考试地点：A1-302 A1-304

## 第一章Basic Concept

机器学习基础（3）

Linear Regression and Gradient Descent (3)

线性回归与梯度下降

Linear Classification and Stochastic Gradient Descent (3)

线性分类、支持向量机、随机梯度算法

Logistic Regression and Ensemble Methods (Decision Tree Adaboost) (3)

逻辑回归与集成学习算法

Overﬁtting, Underﬁtting and Cross-Validation

过拟合、欠拟合与交叉验证

Scientific Reading and Writing（3）

科技文阅读与写作

Multiclass Classification and Softmax (3)

多类分类

Clustering and Dimension Reduction (PCA, SVD, feature selection) (5)

聚类算法与维度约简

Recommendation Systems (3)

推荐系统

Image Processing Basics

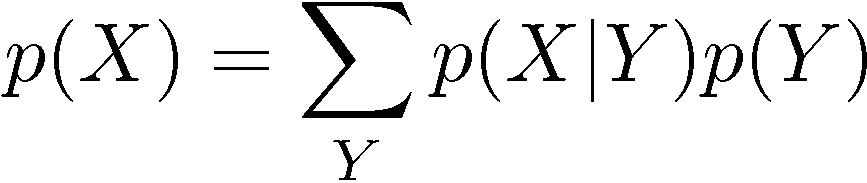
图像处理基础

Neural Networks and Deep Learning(Advanced topic) (3)

神经网络与深度学习

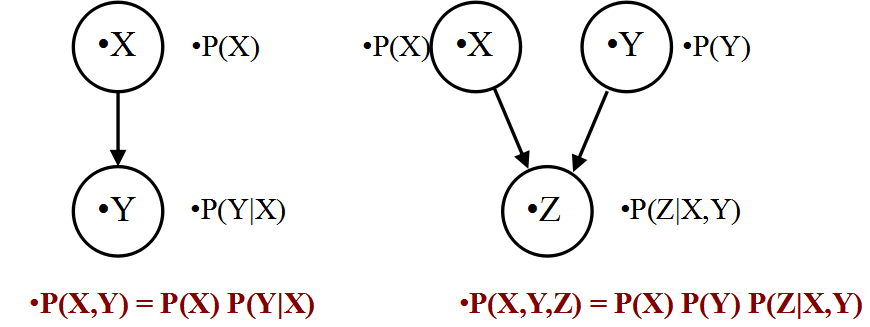
首先导论部分，将概率论的知识复习了一遍，讲解了有关边缘概率，贝叶斯公式等等

### 1.



1. Sum Rule
2. Product Rule

### TP_tmp.emfTP_tmp.emf2.



剩下部分主要是和概率论有关的知识点。主要是在第一章PPT内，请自己觉得比较薄弱的地方直接参考上述公式代入。

### 3.

最大似然估计：

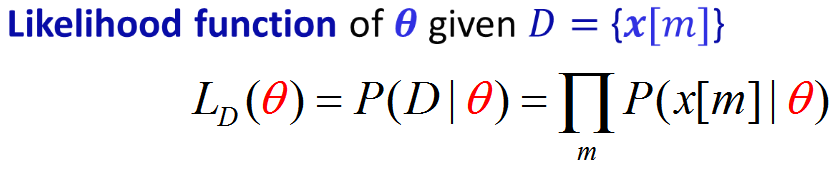
一个概率密度函数的参数有θ∈Ω：f（x|θ）

Θ是一个标量，或者向量（多维元素）

当给出数据D={x[m]}

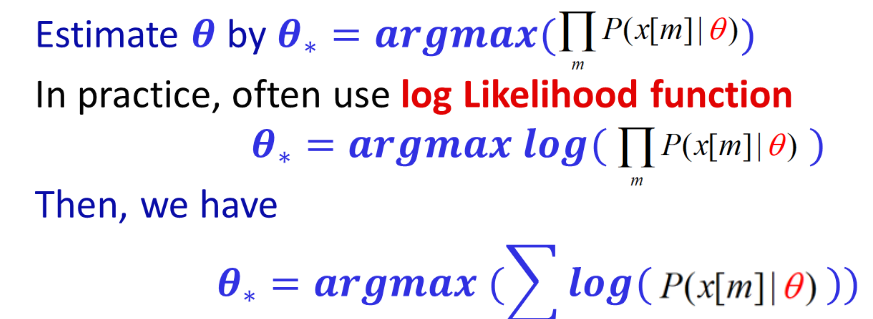
Θ的似然函数为：

L(θ|x)=p（x|θ）=P（X=x|θ）



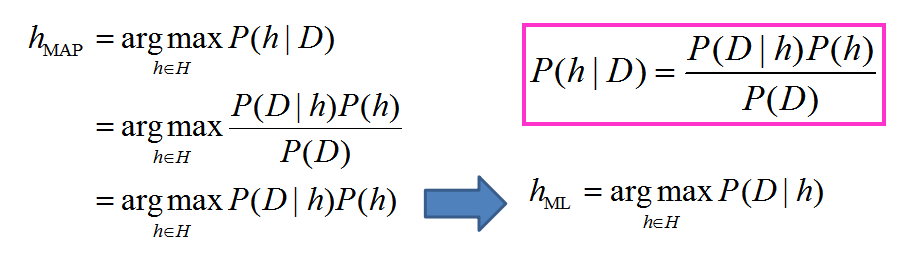
这部分的推导由1,2得知

通过对损失的一个估计和计算



上面是为了求一定的参数，使得目标近似达到最大同时近似可以等价转化为加上log值的内容

MAP VS MLE



第二部分引入信息论（Information Theory）和熵（Entropy）的概念，熵表示事物的混乱度以及信息量的情况

计算方法：

-plog（p）-（1-p）log（1-p） PS：一般以2为指数

应用场景会在后面有说明

## Linear Regression and Gradient Descent

### Basic Concepts about Machine Learning

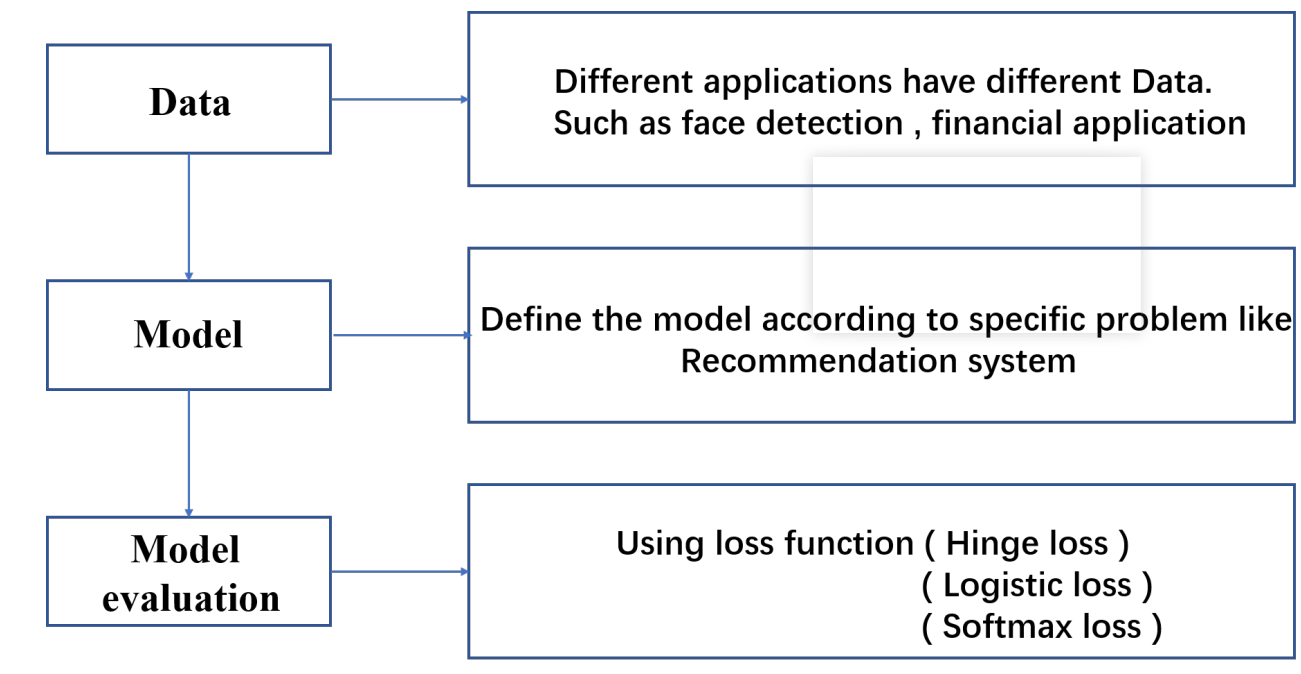
Machine Learning compose of three parts:

Data

Model(function)

Loss(prediction)

三者之间的关系：



我们使用函数去预测y

yˆ = f(x)



损失函数：



损失值越小越好

所以我们目标就是通过对于损失函数的一个简化和降低来去做整个机器学习的一个模型建立。我们的目标是找到一个预测函数使得损失函数最小

### Linear Regression线性回归

Simple linear regression describes the linear relationship between a variable, plotted on the x-axis, and a response variable y, plotted on the y-axis

嗯。上面就是讲一个简单的线性回归描述了在xy之间的一个线性

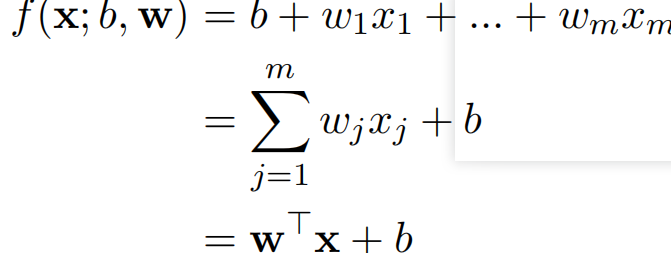
Learn f(x; w) with

Parameters:w ∈ R

m, b ∈ R

Input:x where xj ∈ R features for j ∈ 1, ...m

Model Function:

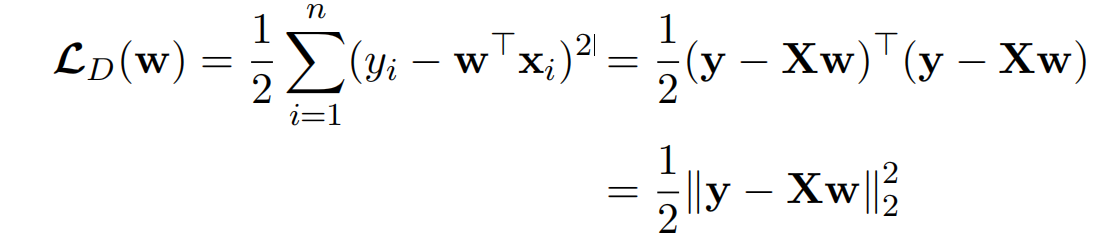


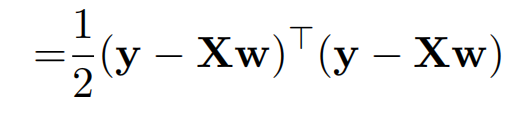
PS：一般我们会设置一个x0=1，直接做成WT\*x

下面就是对于其证明

### Closed—form solution闭式解

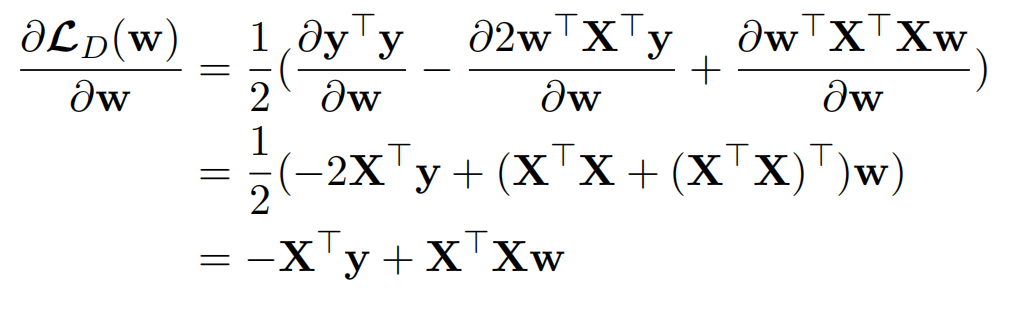
如果损失函数是这个：



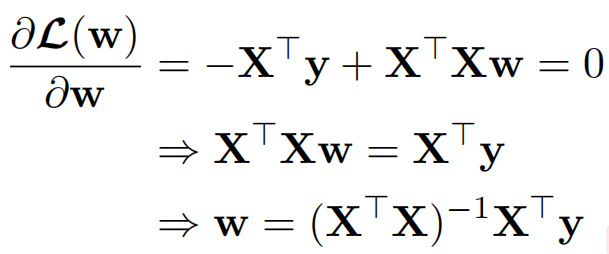


这个是矩阵的一个处理

损失函数对W进行求导：

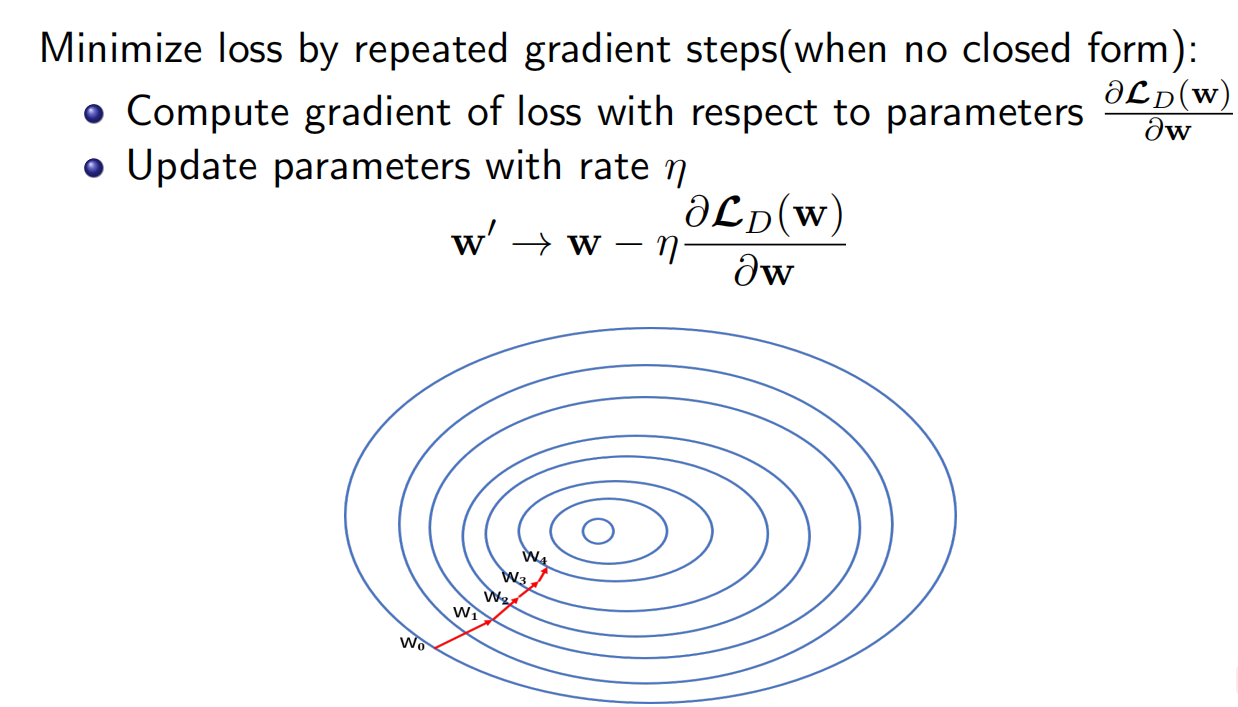


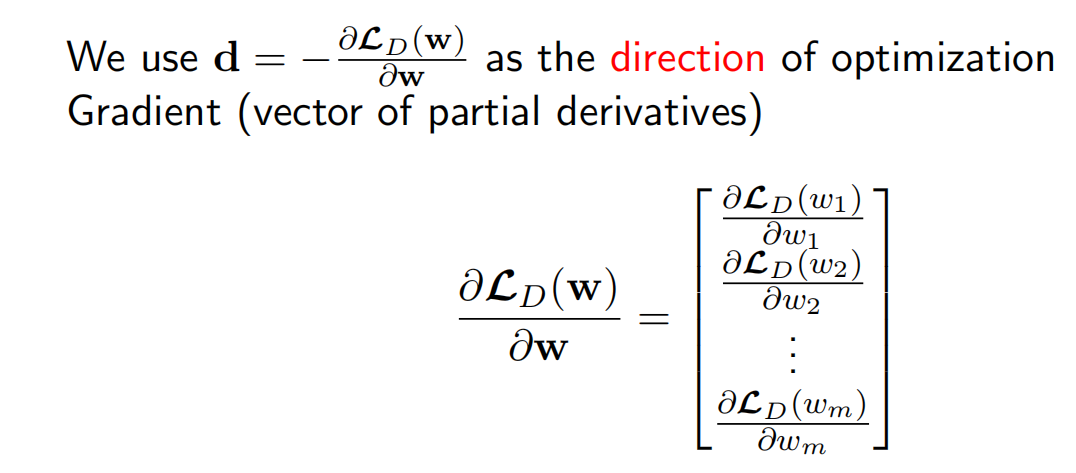
所以当偏导为0的时候，计算w的值：



### Gradient Descent递归下降

递归下降目标:



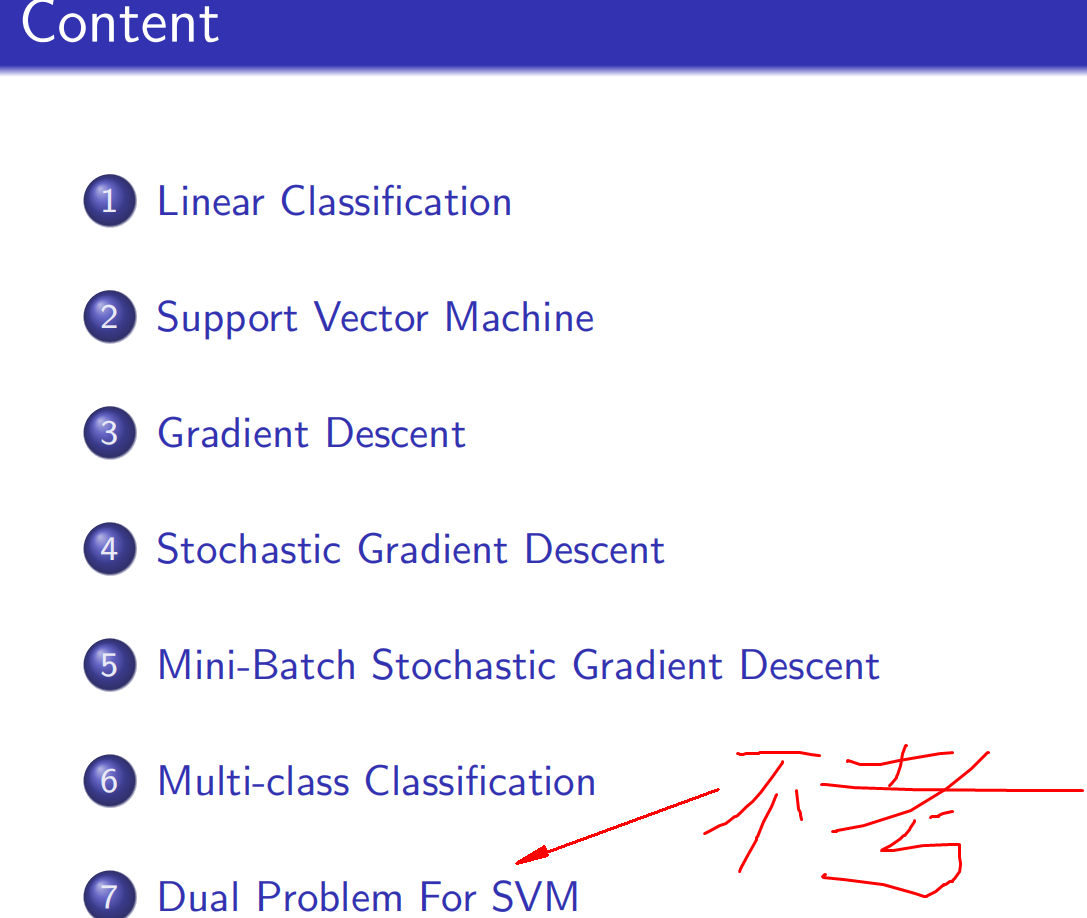


递归的结果及其方法，请查看第一次实验

## Linear Classification and Support Vector Machine and

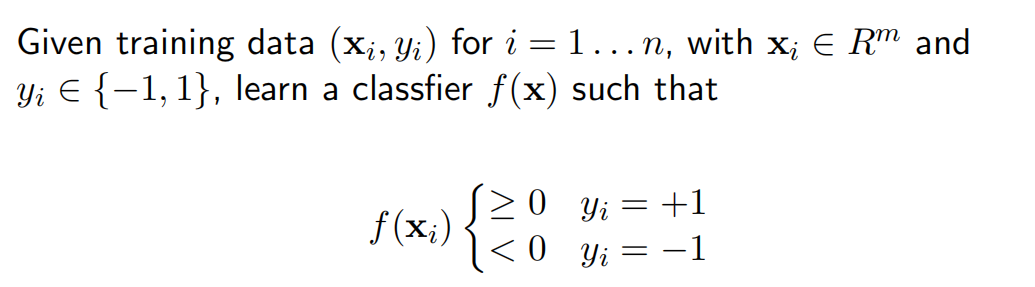
## Stochastic Gradient Descent and Multi-class Classification

线性分类，支持向量机以及随机梯度下降和多类识别

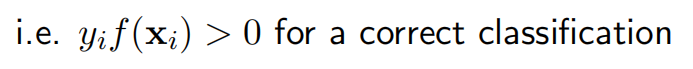


### Linear Classification

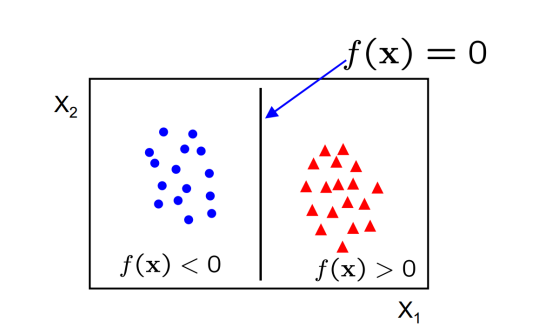
给出训练数据data，y是在{-1,1}



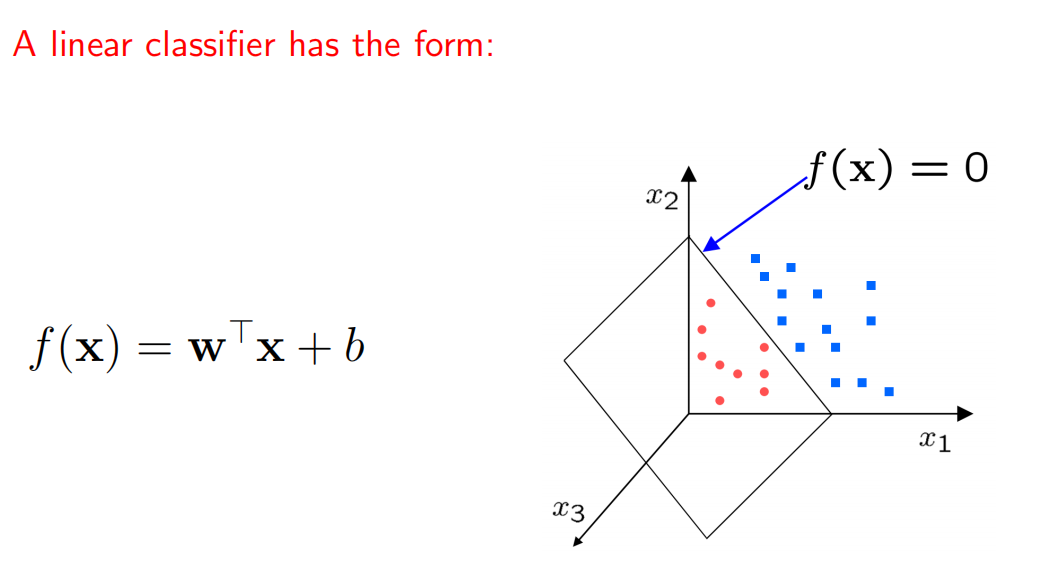
所以可以得出下面的情况



f(x) = wTx + b

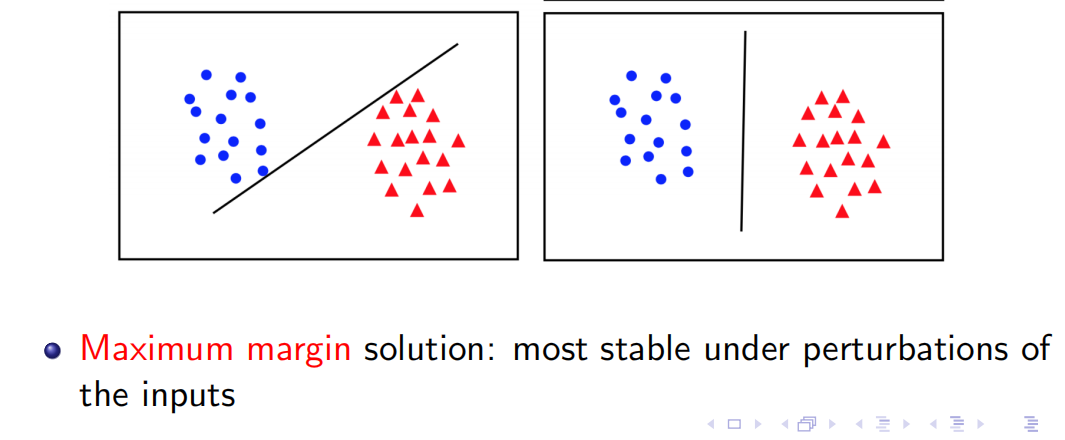


这个是二维的结果



这个是三维的结果，而还是可以向上不断进行

### Support Vector Machine



最大边缘解决，这两个图明显是第二个最有效

：

Select two parallel hyperplanes that separate the two classes

of data and let the distance between them as large as possible

选择两个分隔这两个类的并行超平面

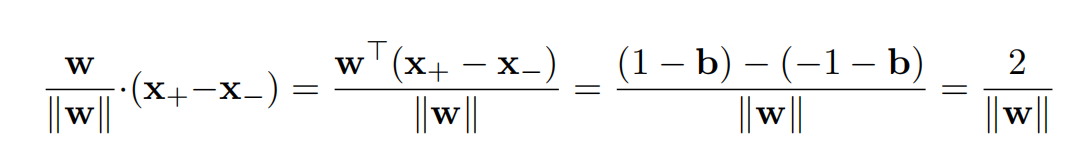
并让它们之间的距离尽可能大

The region bounded by these two hyperplanes is called the

”margin”

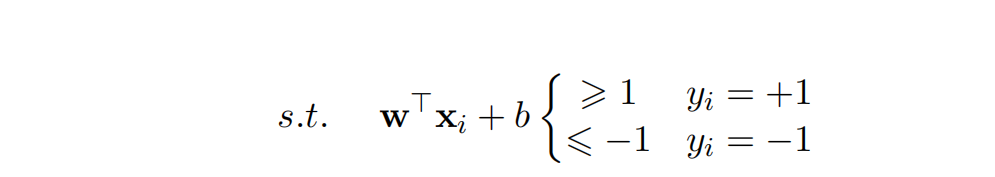
由这两个超平面包围的区域称为“margin”

计算得到最后的距离

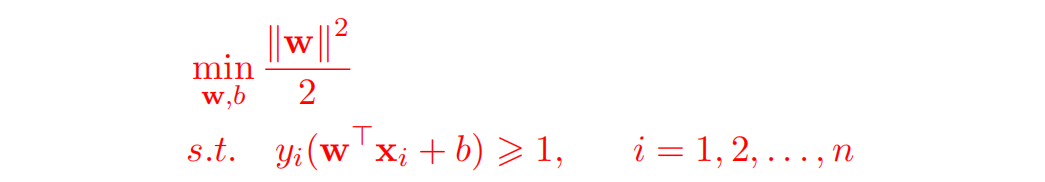


我们使用SVM是为了得到一个优化结果使得：

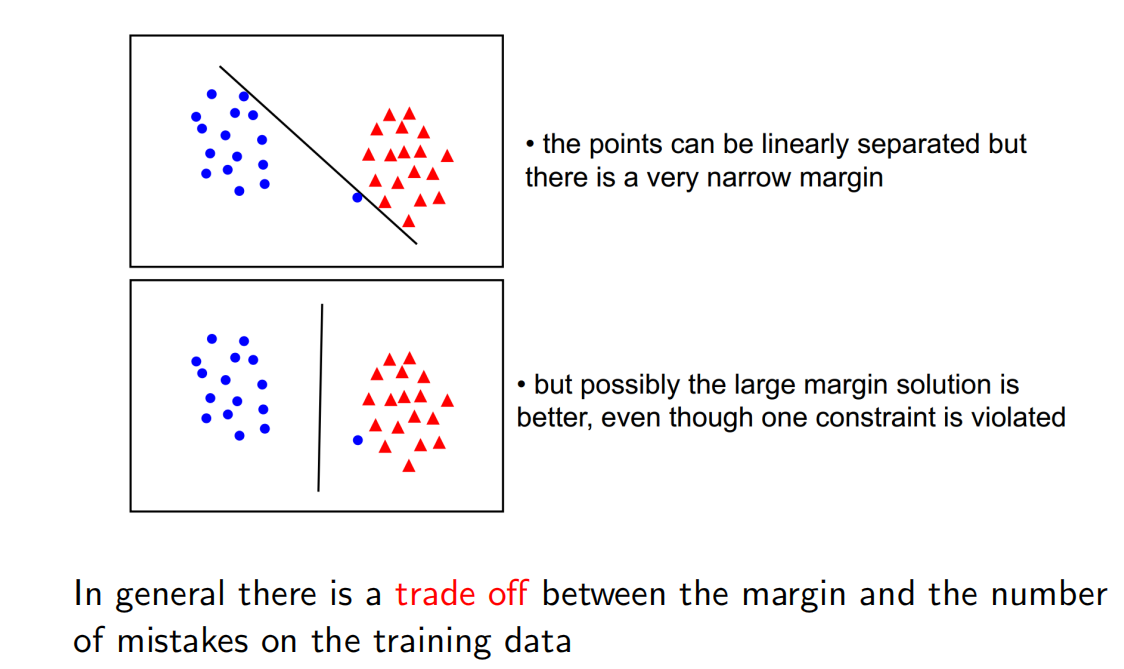
2/||w||最大



这个的解决结果就是会等价于这样的情况：



通常来说在margin与错误率之间有个平衡点：



在一些情况下，训练数据可能不会被线性的分离：

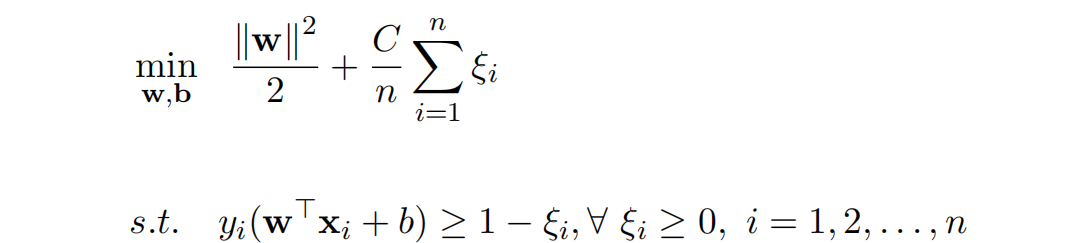
引入变量ξi > 0，对于每个i，每一个代表着样本i在margin边界的错误边界的数量

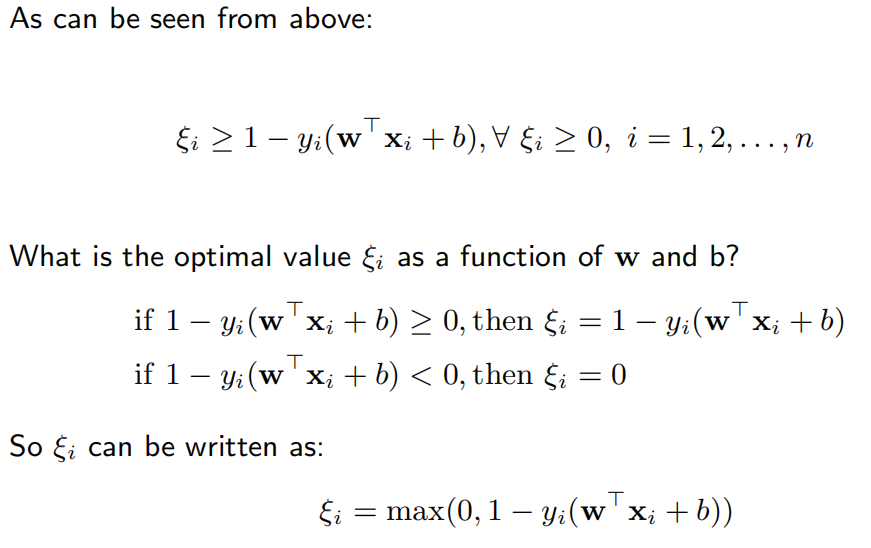
如果ξi =0，还是可行的

如果ξi <1，这是在被正确的分类，有个最小的margin小于1/||w||

如果ξi >1，分类不正确

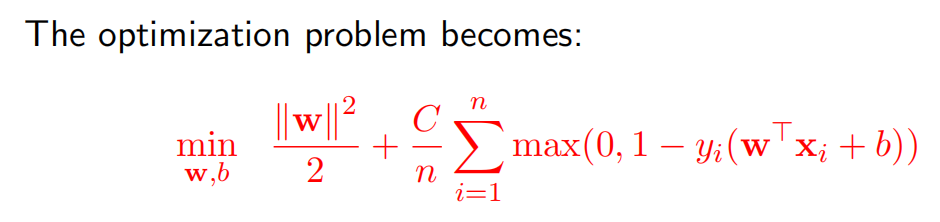
优化方案就变成了：





ξi = max(0, 1 − yi(wTxi + b))

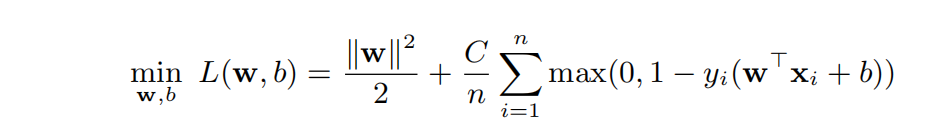
最优化问题可以被看成：



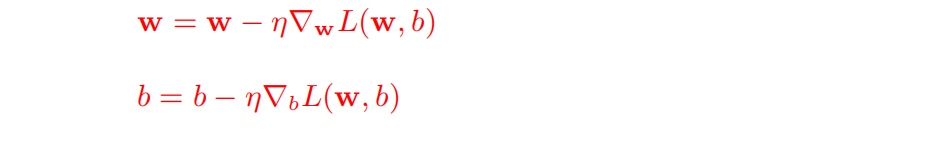
### Gradient Descent

递归下降

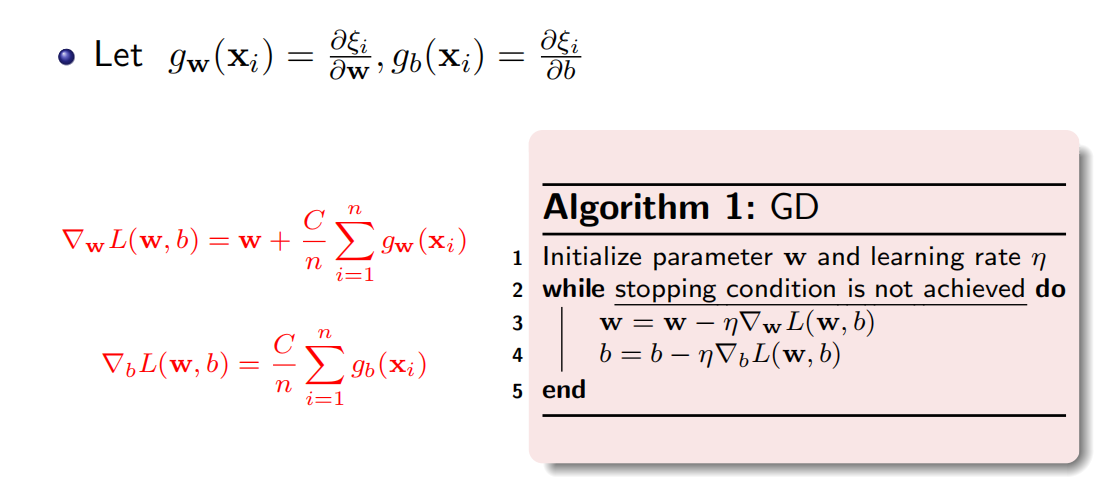
损失函数：

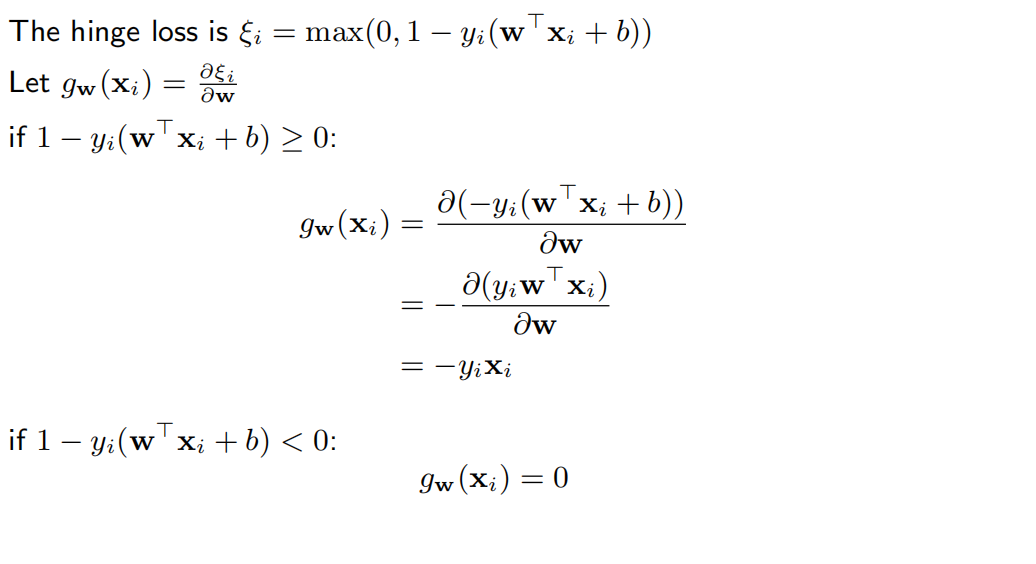


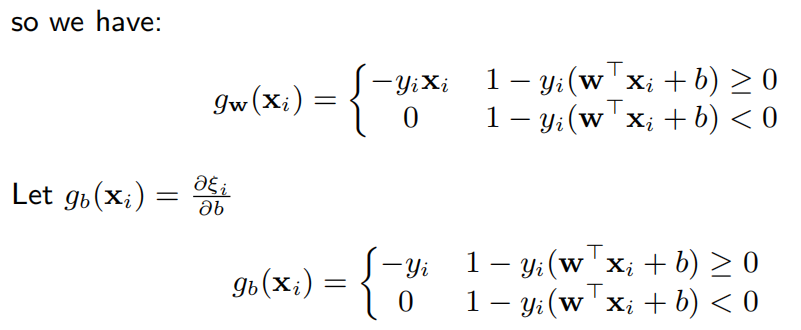
这个是更新参数的情况



这个是对于递归下降的一个算法和计算情况





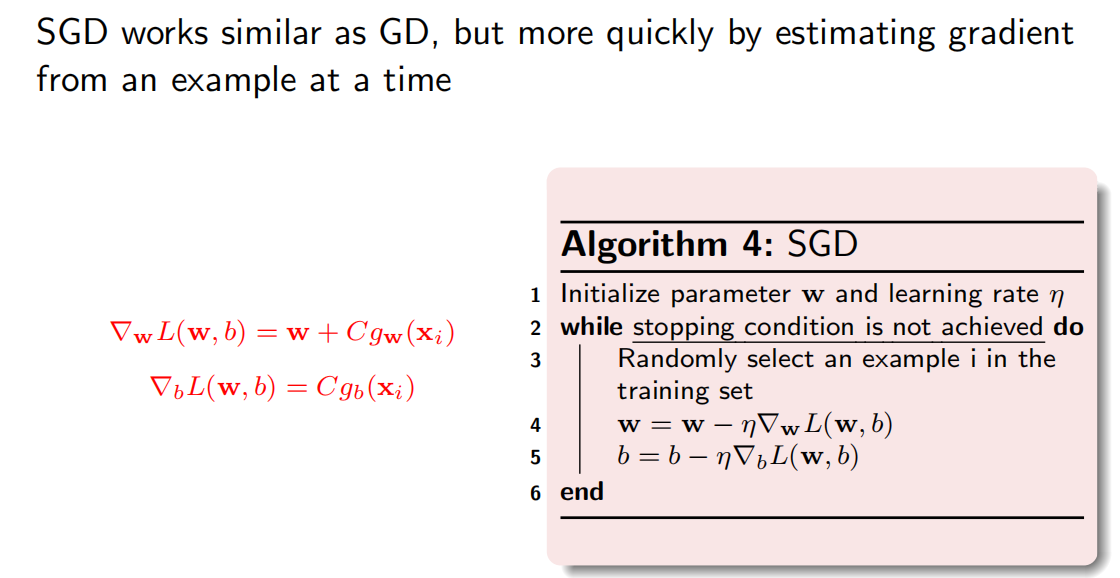


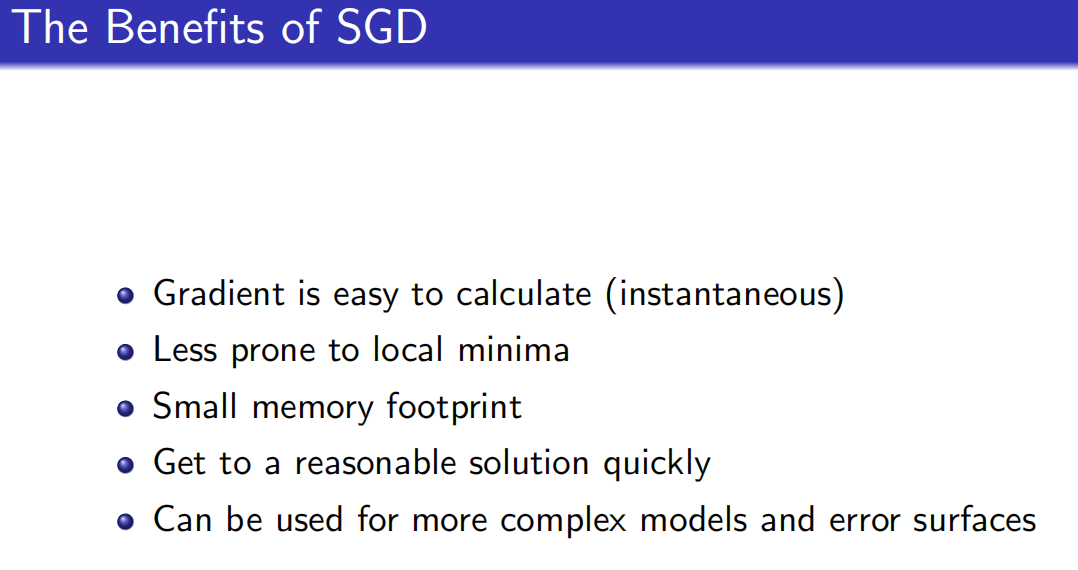
### Stochastic Gradient Descent

随机梯度下降：

Key points：从一个小的，当前数据集的子样本中有足够多的梯度迭代帮助我们收敛到全局最小值

随机梯度下降：



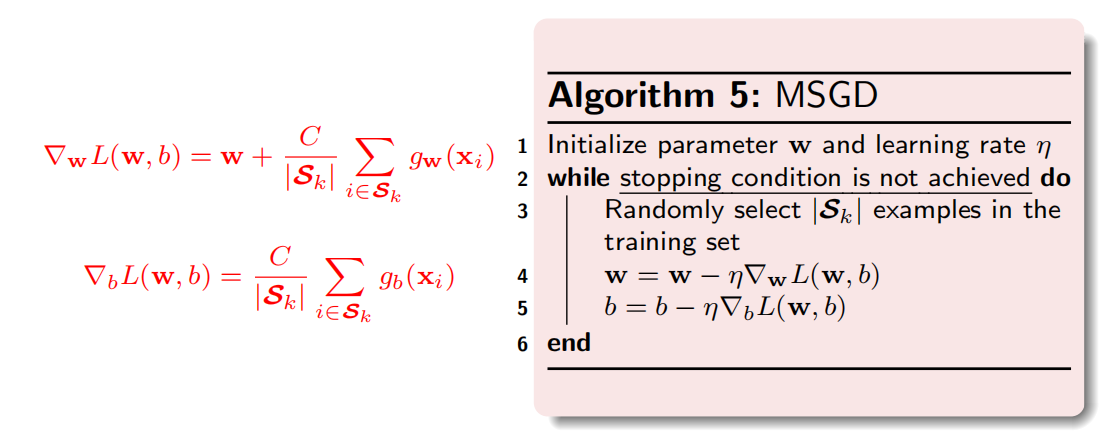


## Mini-Batch Stochastic Gradient Descent

批量随机梯度下降（MSGD）

MSGD的工作原理与SGD相同，只是我们使用的比SGD多

一个训练的例子，使每一个估计梯度



## Multi-class Classification

Three common classification problems：

Binary classification

Multi-class classification

Multi-label classification

多类分类是一种常见的分类问题，它将实例分类为多个类中的一个

多类分类的三个常用的策略：

Transformation to binary classification

Extension from binary classification

Hierarchical classification

二分类转换

二进制分类的扩展

分层分类

二分类转换：

这个策略就是将一个多类分解变成多个二类分解的问题

1. 一对rest
2. 一对一
3. Decision Directed Acyclic Graph, DDAG（决策有向无环图）

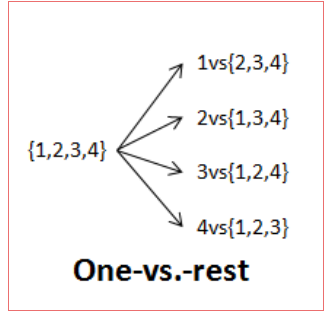
One VS rest方法：

1.方法

训练：

对于每个类：

训练一个二分类器，将该内容作为正分类，其余的变成负分类

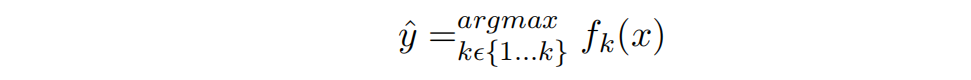


1. 预测：

对于每个二分类器：

产生一个真值信心得分（confidence score）

预测最高得分下的那个label



1. 优缺点

Advantage：

Trains K binary classifiers for a K-way multi-class problem.

Disadvantage：

1. The distributions of the binary classifications are unbalanced.

(The set of negatives is much larger than the set of positives.)

1. The scale of the confidence values may differ between the

binary classifiers.

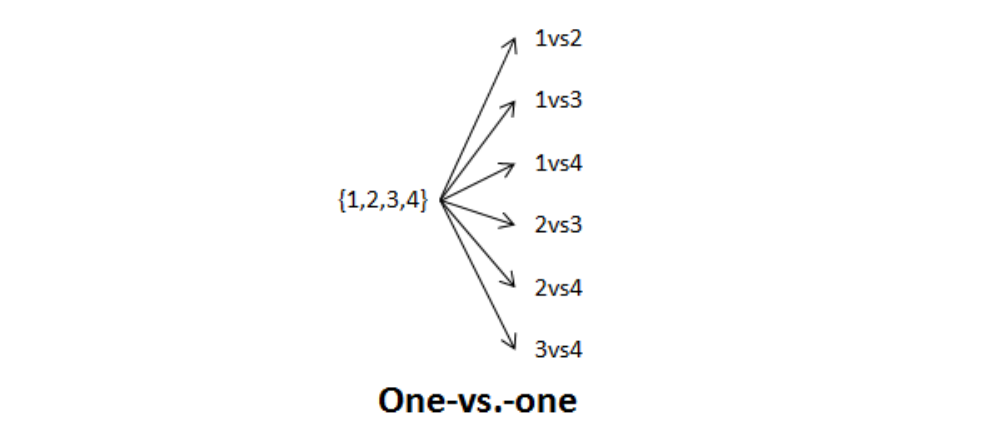
One VS one 方法：

1. 方法

训练：

对于每个类的pair：

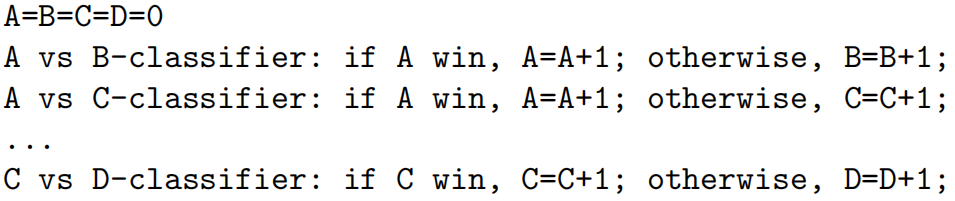
训练一个二分类器来去分辨两个类



1. 预测：

对于每个二分类器：

比较两个目录内容并且进行投票:



Predict the label with the maximum number of votes wins.预测这个计算的内容

yˆ=Max(A,B,C,D)

1. 优缺点：
   1. 优点：

The distributions of the binary classifications are balanced.

二元分类器的分离是平衡的

* 1. 缺点：
     1. Train K(K − 1)/2 binary classifiers for a K-way multi-class

problem, which has high computed complexity.

对K路分类问题，训练 K(K − 1)/2二分类器，这个能有高计算复杂度

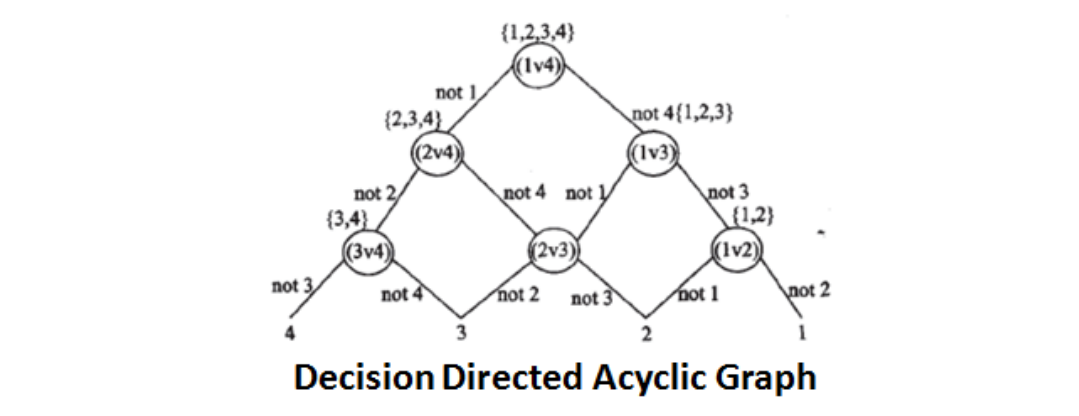
* + 1. Suffer from ambiguities when receive the same number of

votes.

当得到相同的投票的时候，需要忍受模棱两可的选择

#### DDAG：

1. 比起来一对一方法，此方法使用了一个根基础的二分有向无环图，这个有着内节点和叶子节点：



1. 预测：

从根节点开始

在到达叶子节点之前：

评估二元决策函数

根据输出值向左或者向右移动

比起来One VS one 方法：

每个二元分类器的相关性减少了预测的难度和代价

#### Extension from binary classification二元分类器的延伸

该策略扩展了现有的二元分类器机制，能够解决多类分类问题：

Neural networks

Decision trees

K-nearest neighbours

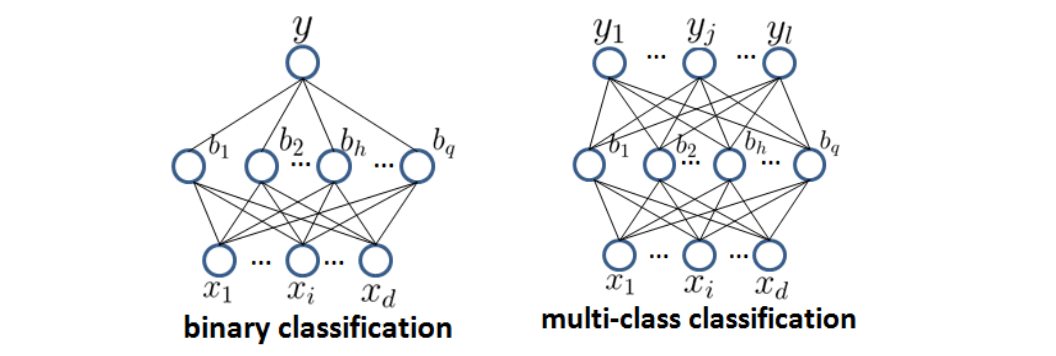
Softmax function

#### NN：

1. 介绍：

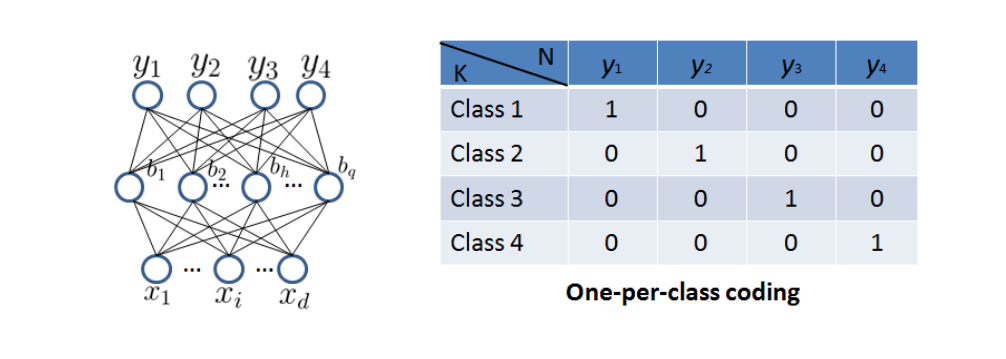
Instead of just having one neuron in the output layer, the network could have N binary neurons leading to multi-class classification.

不是在输出层仅仅只有一个神经元，这个网络有n个二元神经元从而实现多类分类



1. 进一步的认识：

每个输出神经元被指定辨识一个给出的类。N=K



Train：

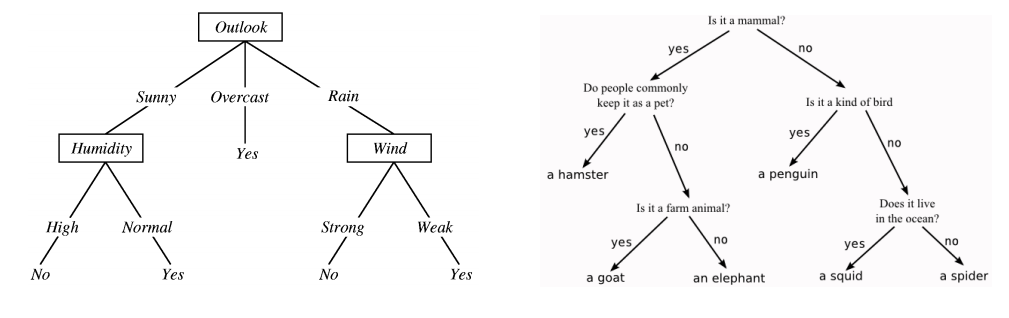


Predict：

输出最大的神经元被认为是例子中的那个类

#### Decision Tree

决策树试图根据特征值分割训练数据，以产生良好的generation



这个算法能够自然的处理二分类和多类分类

#### nearest neighbours：

算法：

1. 计算距离

计算在测试对象和训练集的每个对象

1. 找到邻居（neighbors）

找到K个最近的邻居作为neighbors

1. 投票标注

基于邻居最常用的类的邻居分类

Advantage

The method is a non-parametric classification algorithms.

The algorithm can naturally handle binary and multi-class

classification.

方法是无参数分类算法，这个算法能自然地处理二分和多类分类

Disadvantage

The computational and memory requirements are high.

Finding good representations and distance measures between

objects is hard.

计算和内存需求都特别高，找到在物体之间的好的展示和距离度量是困难的

#### Hierarchical classification不考

层次分类：

这个策略解决多类分类问题，方法是将输出空间变成一个树

Label Tree

Train：

在叶子节点之前，只包含一个单独的类

每个父节点被分为大量的聚类，每个聚类对应一个子节点

在每个节点，一个简单的分类器被训练去辨识不同的子类聚类

Predict：

从根节点开始

Travel到每个叶子节点，这个叶子节点就是一个标签

优缺点：

优点：

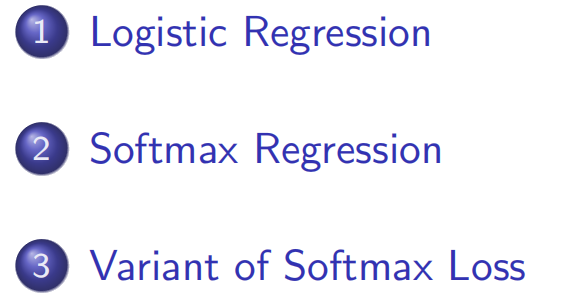
这个树方法能够降低预测的代价

缺点：

找到好的聚类方法很重要

## 第四章：Logistic Regression and Softmax Regression

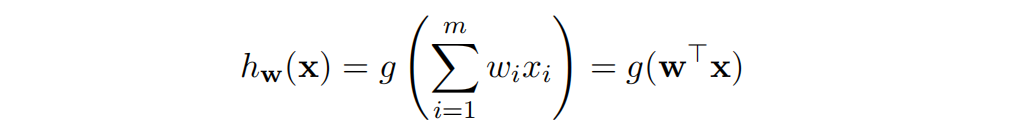
逻辑回归和归一化回归



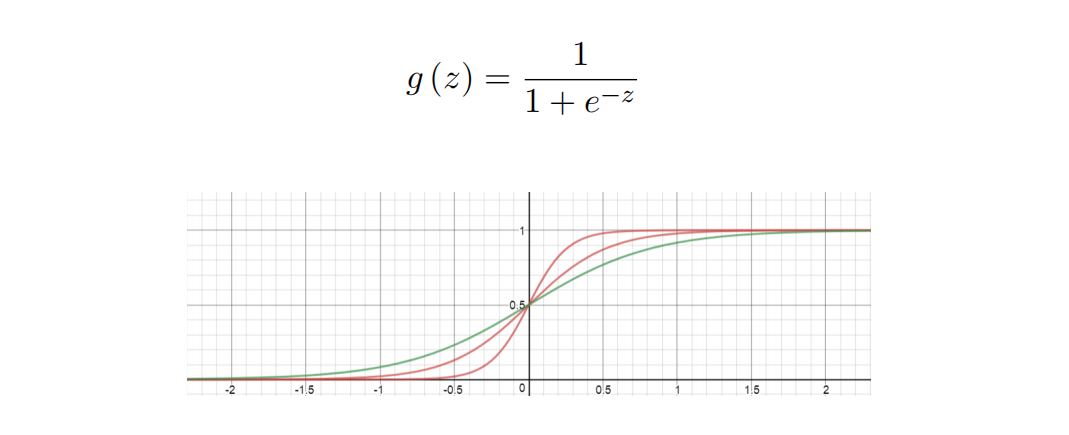
### Logistics Regression

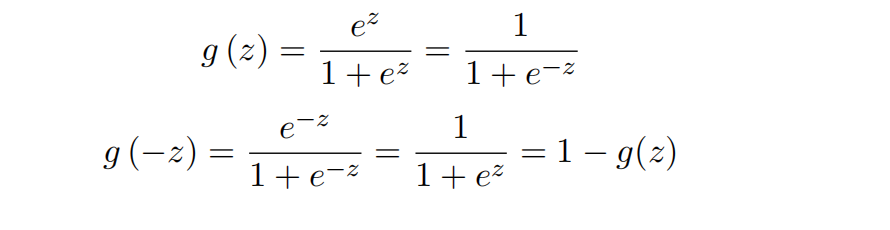
逻辑回归

At first，给出一个似然函数：

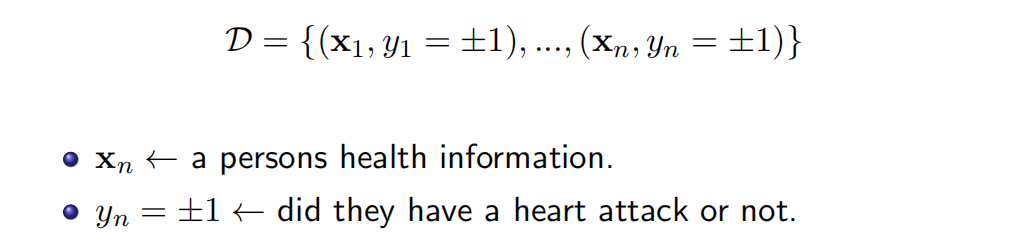


逻辑函数：

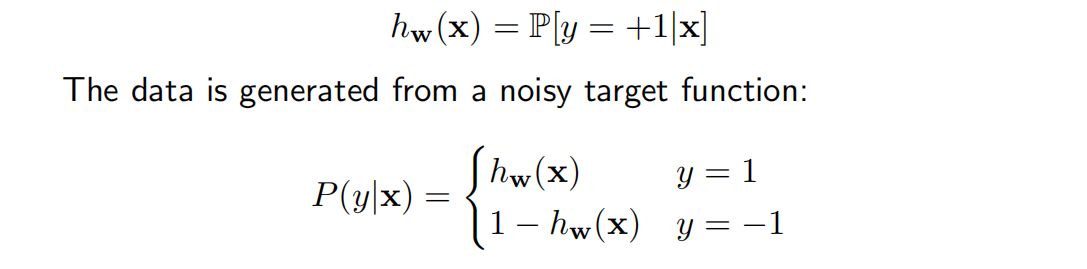
有一个函数关系：



数据情况：

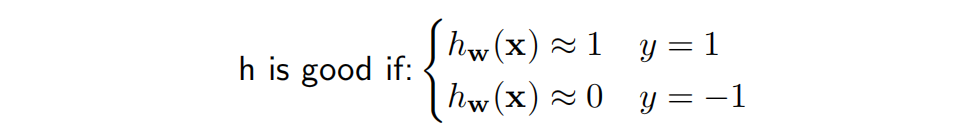


目标函数：

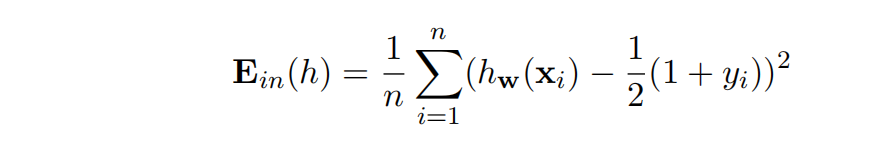


Hw（x）意味着当为x的时候y为+1的概率

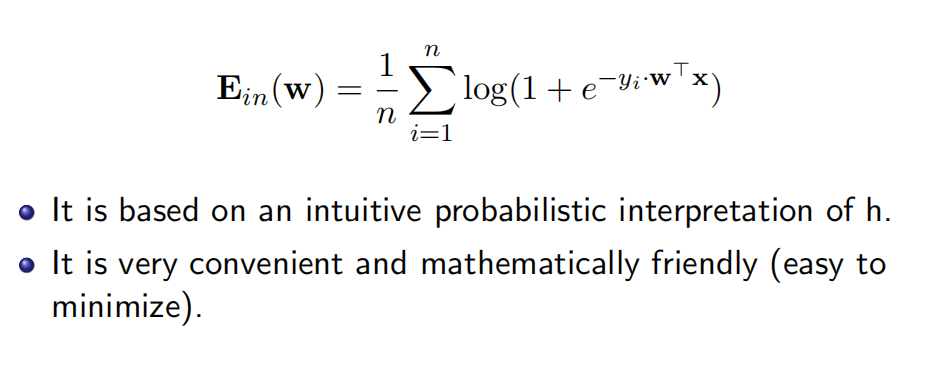
什么时候h是好的？



错误率计算：



同时对于这个计算比较复杂，所以简化了：

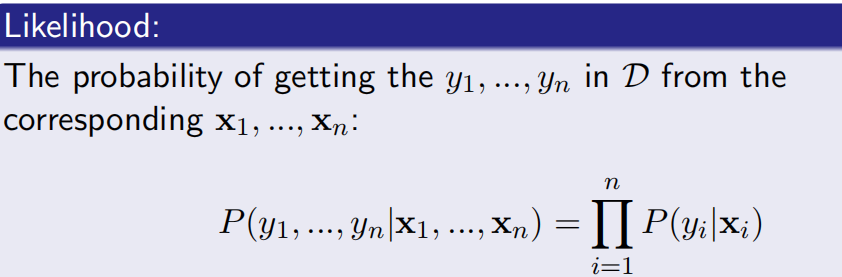


PPT13页

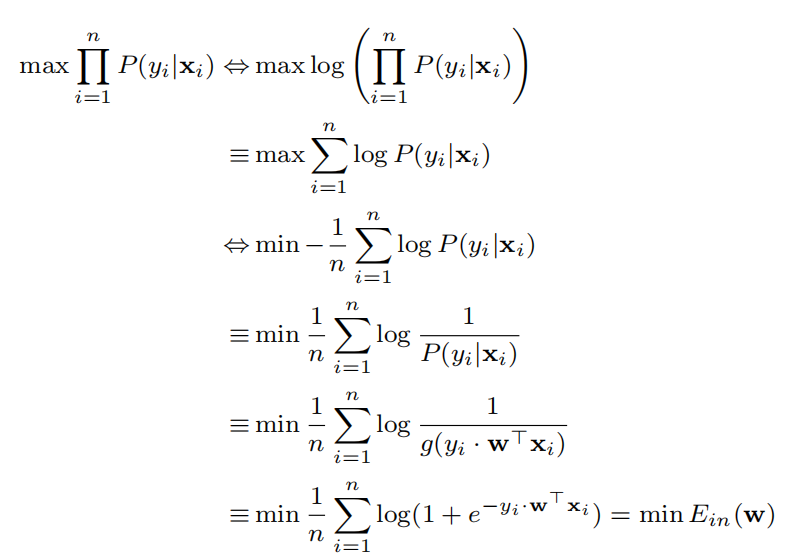
一系列推导让我们得到一个推测函数：



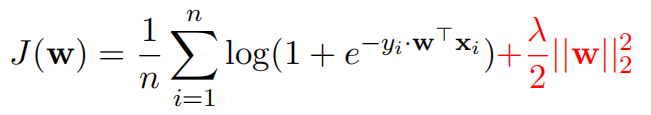
同时我们可以回顾第一章的一个似然计算：



对于最大似然和减小错误率的一个估算：



正则化：



W的取值如果偏小

模型会简单些

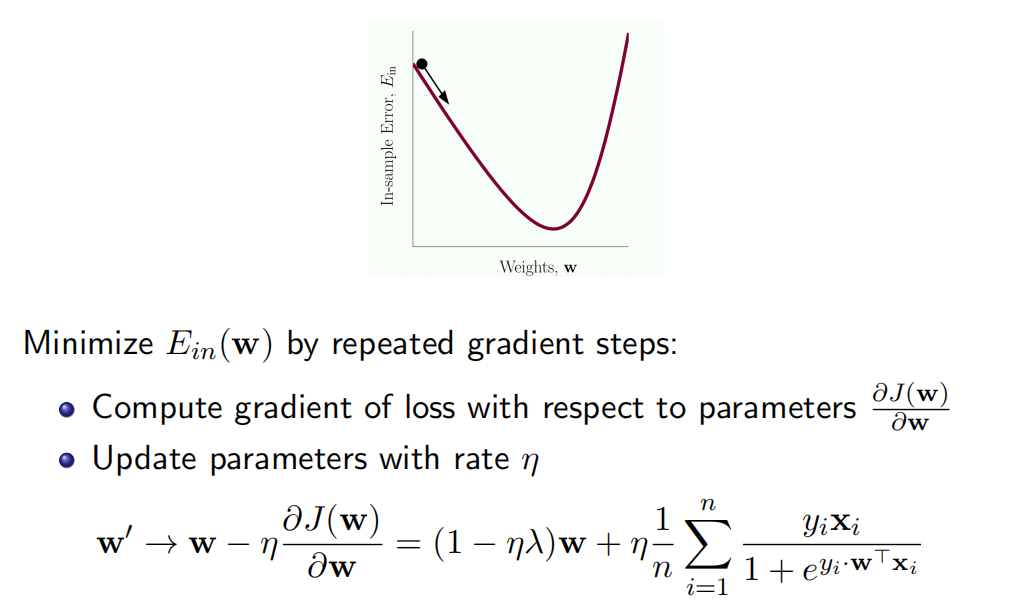
更小的可能会过拟合

正则参数λ

在保持模型相对简单和足够拟合训练集之间进行权衡

找到最好的权重值：

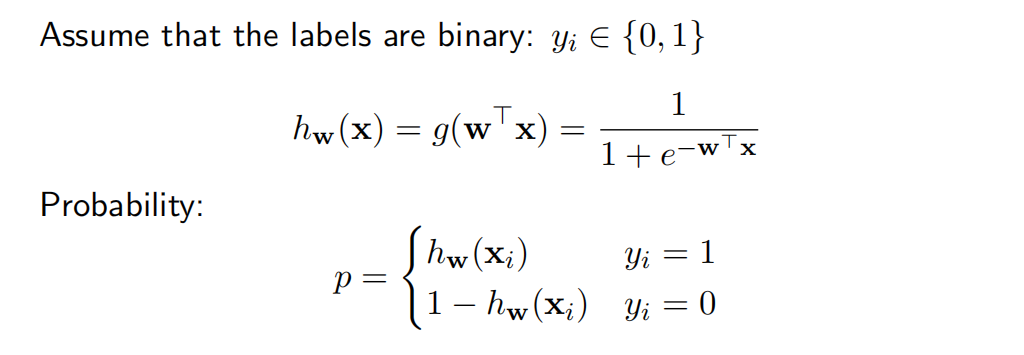
使用递归下降的方法



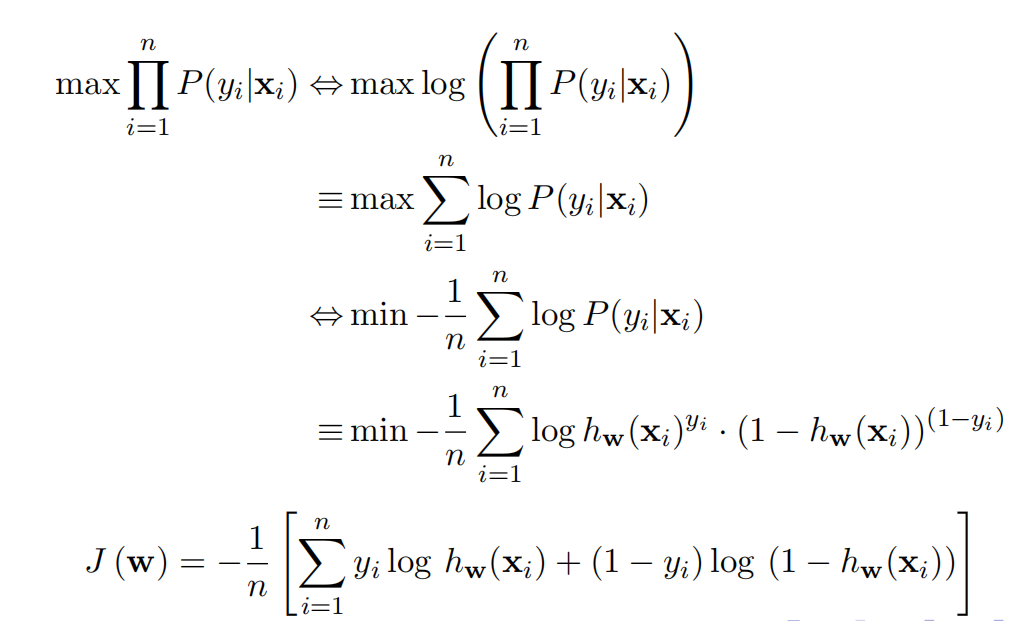
ɳ实际上是学习率

以上为分类结果为{+1，-1}的时候

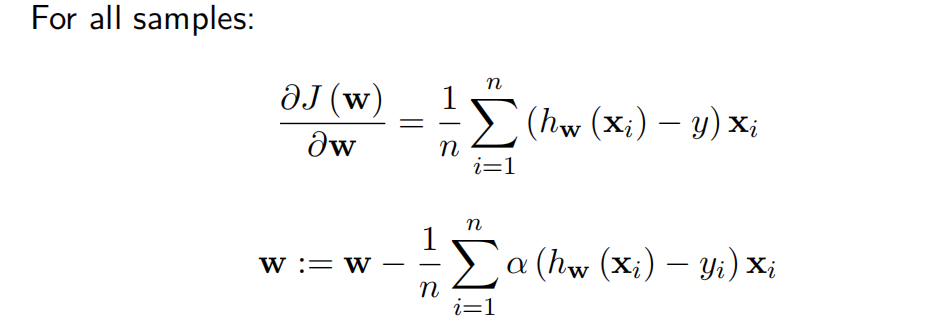
下面为分类结果为{+1, 0}的时候情况



结果损失函数变成了：



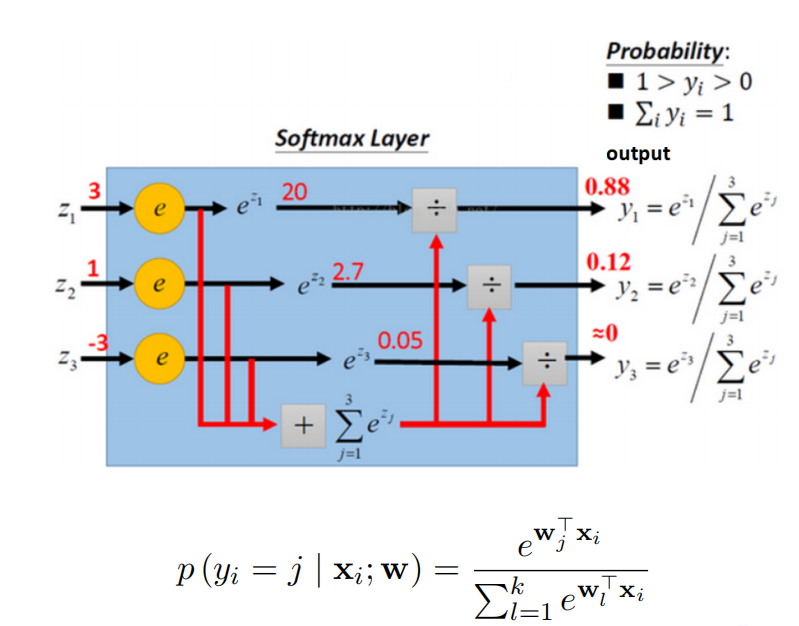
就计算可以得到权重的更新w’



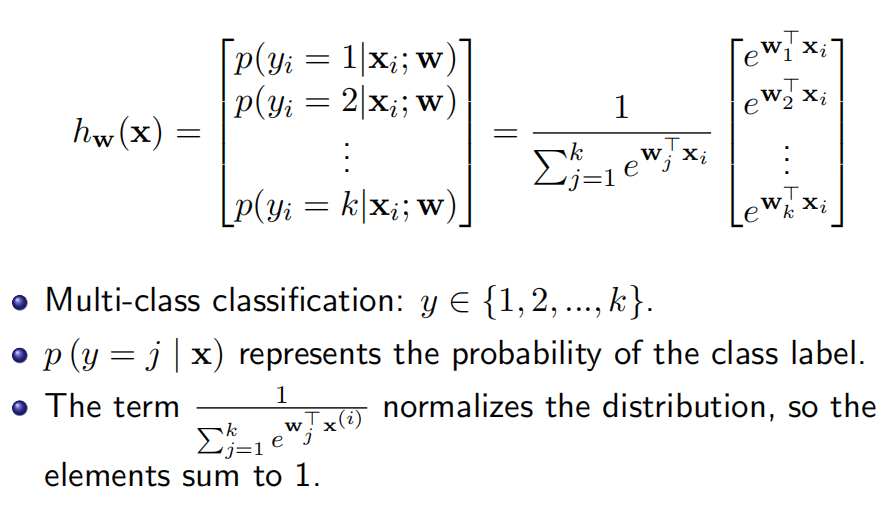
### SoftMax Regression

正则化

多类分类器

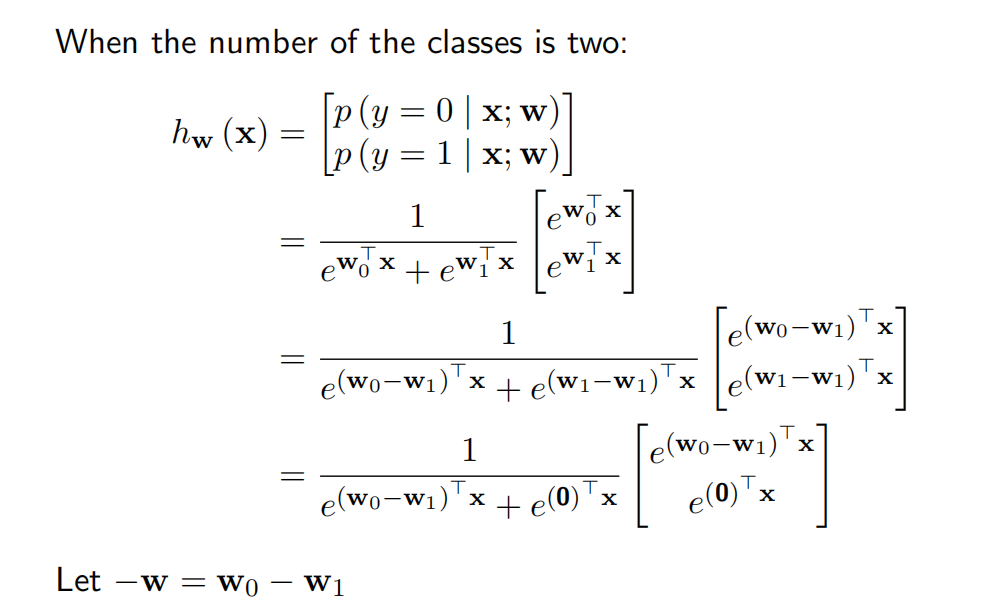


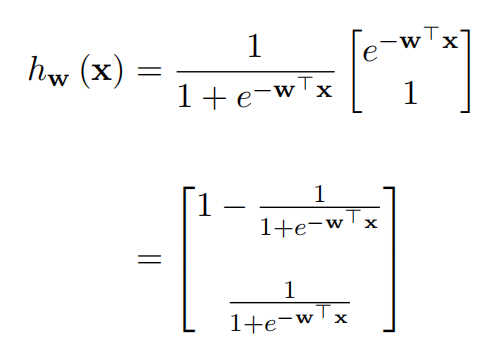
突然出现了这张图，无言说



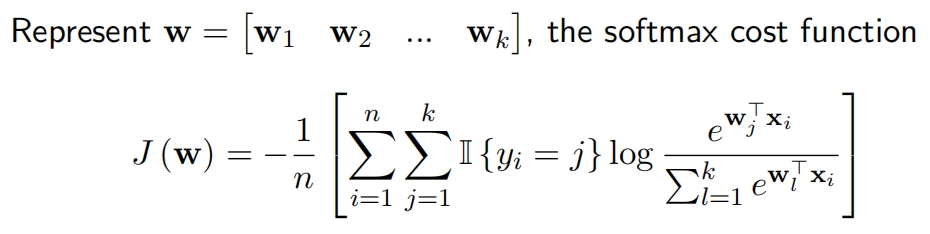
这个部分是对于一定程度上正则化的一个处理

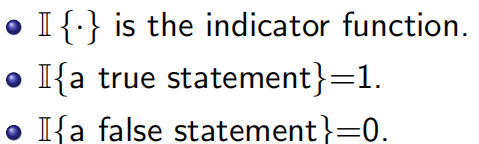
如果是两类分解问题，而且进行一个处理



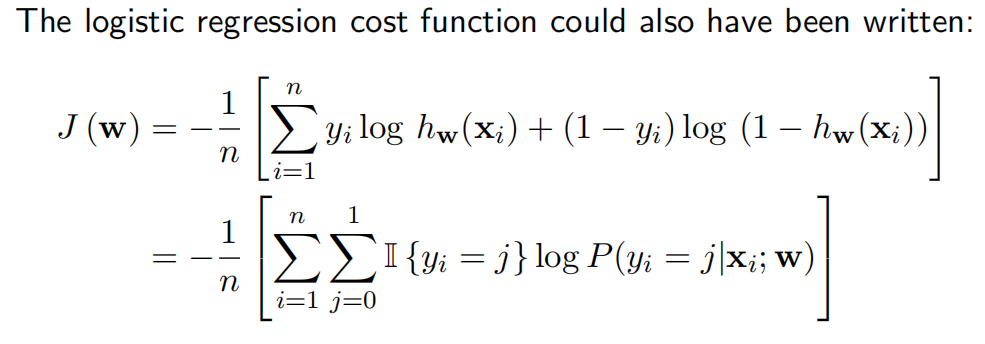


这个说明了softmax的一个作用，就是将逻辑递归的一个实现

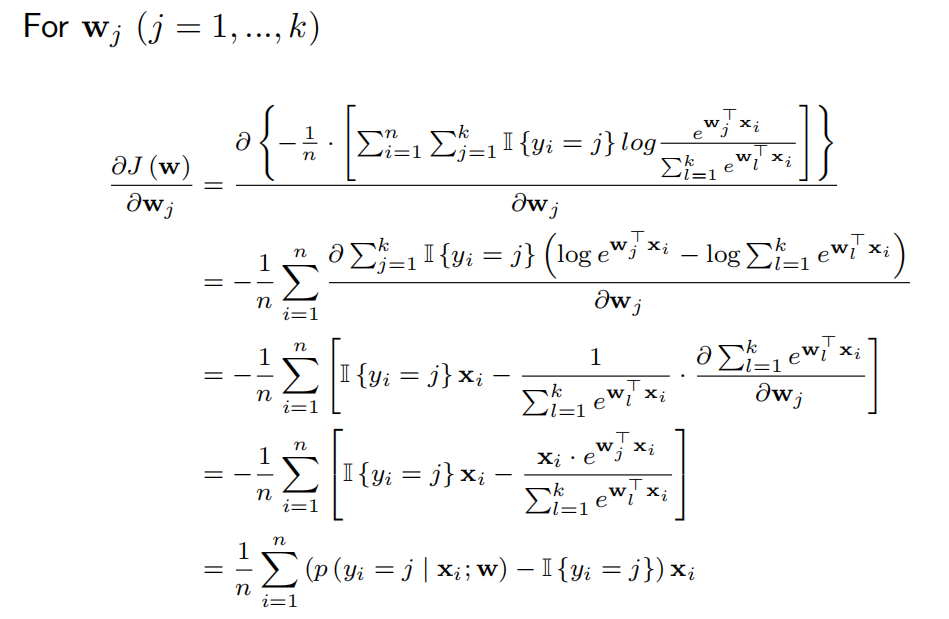


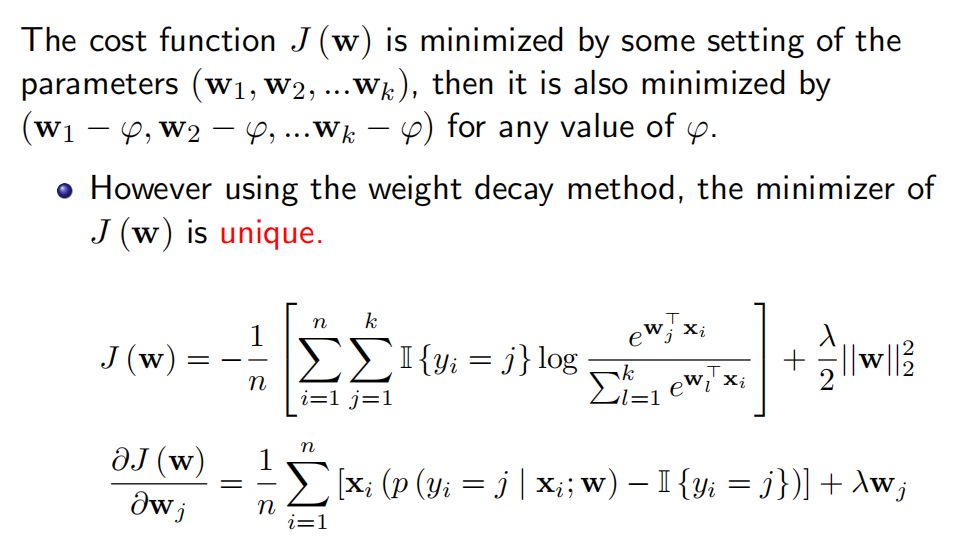


逻辑回归的方程式：

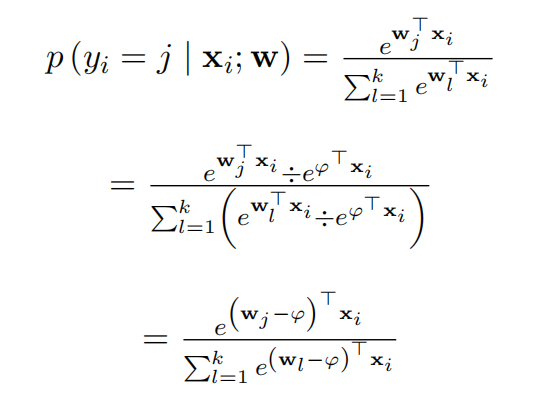


最后是关于softmax函数对w的一个求导





接着加上一个函数的冗余φ



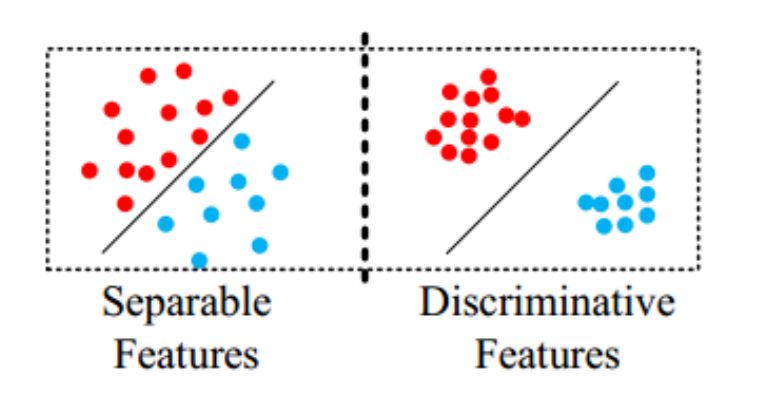
### Variant of Softmax Loss

Large-Margin Softmax Loss

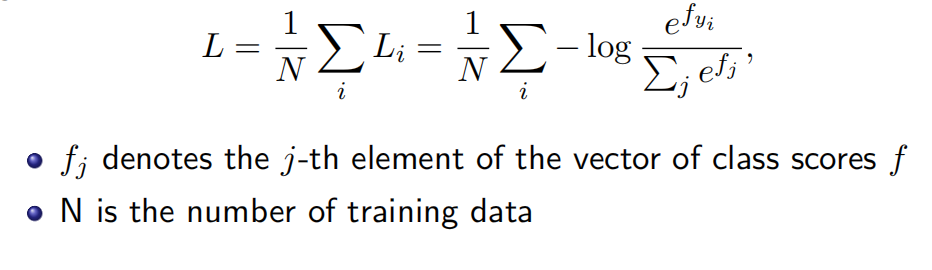
Angular Softmax Loss

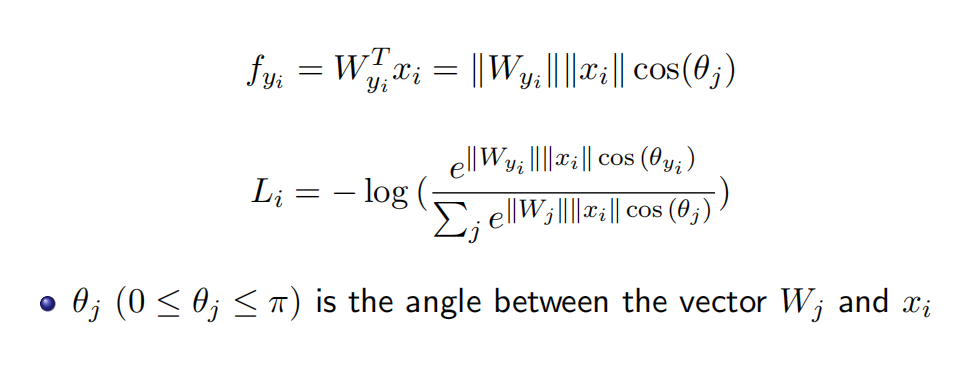
动机：学习一个识别的特征

使得左边的变成右边的：



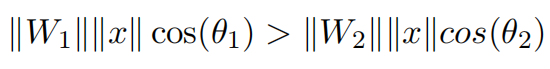
鉴于给出的特征xi和标签yi，softmax损失函数的表现为：



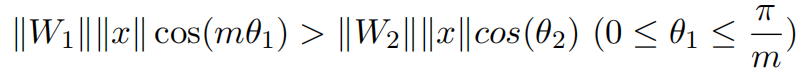


参考二元分类器，我们有来自一个1类的样本

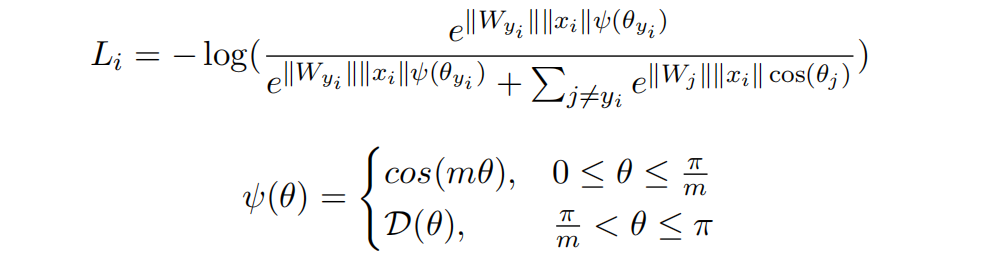
原softmax：

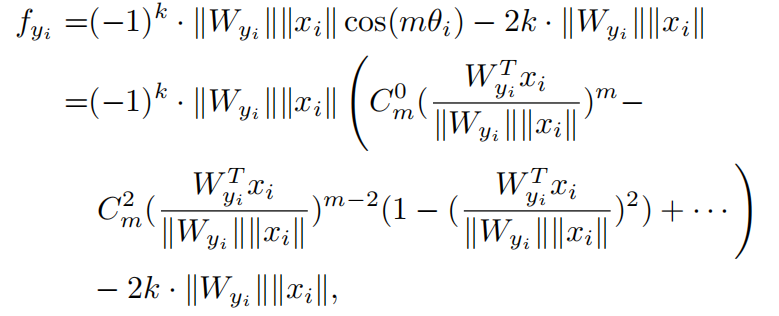


Large-margin softmax：

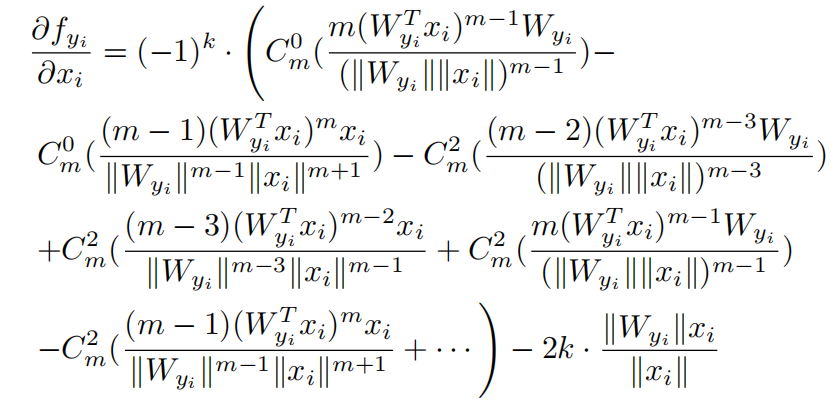


Large-margin softmax Loss：



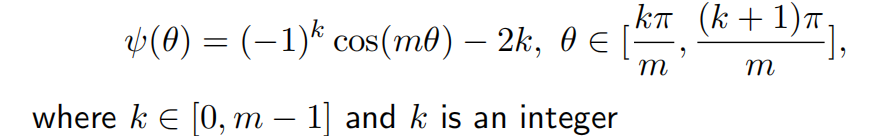


以上公式为更改之后的内容

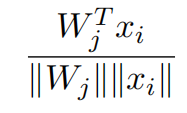


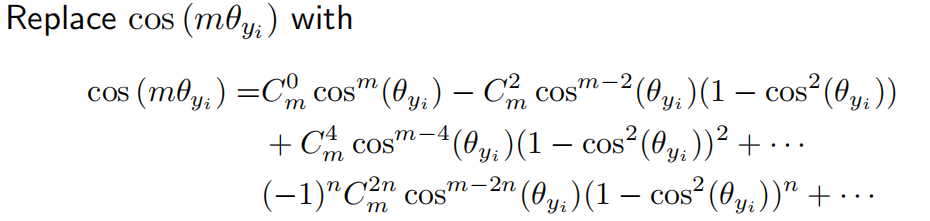
第二种：Angular Softmax Loss

构建一个特殊的θ函数

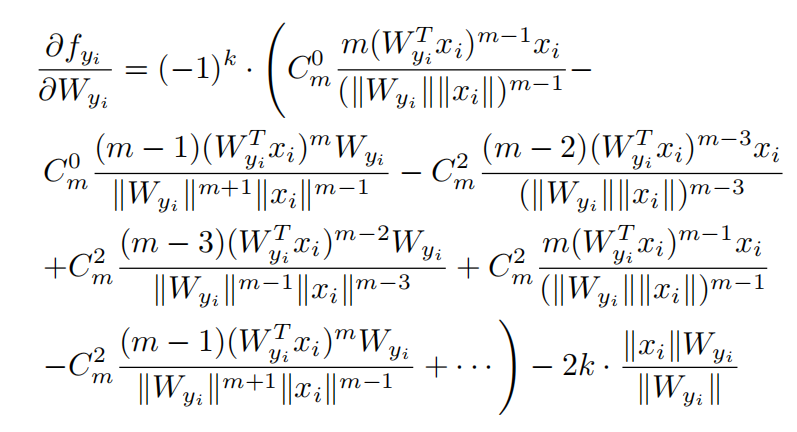


重置cos(θi)用：





请不要问我为什么要使用这个



## 第五章Cross-Validation

交叉验证

内容有：

1. 模型验证的问题：

欠耦合觉过耦合

Bias-varience 权衡

1. 交叉验证

训练集数据，验证集数据和测试集数据

性能报告

参数调整

K-fold交叉验证

1.

为什么我们需要验证？

1. 商业原因
   1. 需要选择最佳模型
   2. 衡量所选择模型的好坏（accuracy/power）
   3. 足够好来去衡量建模项目的ROI
2. 统计原因
   1. 模型创建技术被天然地设计去减少或最小化bias或者loss
   2. 对于一个范围，一个模型总是和信号一样去适应噪声
   3. 如果你只去适应在一个数据集上的一些模型，并且选择最优的，可能会有过优化的结果

在训练集上面的误差有时候可能会被误导

不要预测未来的表现

太复杂的模型可能会导致未来数据的不准确性

有时候被称作这个被称作偏差变量权衡

复杂的模型：

低“bias”：

模型在训练集上面很好

模型的值和数据的期望值很接近

高“Varience”：

模型更可能是一个错误的预测

欠拟合和过拟合的标志：

1. 如果通过增加容量我们降低实现错误，那么我们在欠拟合，否则就是过拟合
2. 如果测试集展示的错误相当大而且实现错误也是很大，那么就是欠拟合

需要增加容量：

如果展示在训练集的错误很小而实现错误相当大，那么就是过拟合

需要降低容量或者增加训练集：

有很大特征和相当小的训练集

如果你已经选择一个大的容量来去实现这些很多的特征，你可能过度拟合这个数据

需要降低容量：

2.

数据分离

最简单的想法：把数据分为两片

训练数据：用来去拟合模型的数据

测试数据：用来去评估模型的“新鲜”数据

测试数据包括：

真实目标值y

预测目标值y\*

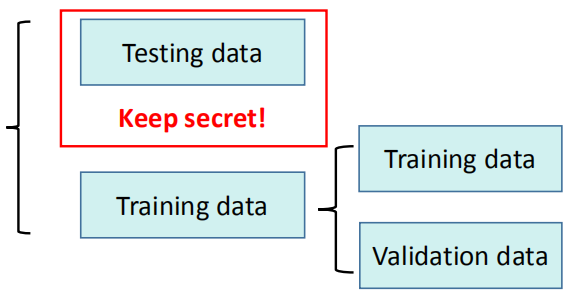
我们能够找到展示y和y\*关系的聪明的方法：

升力曲线

增长图表

ROC曲线

......

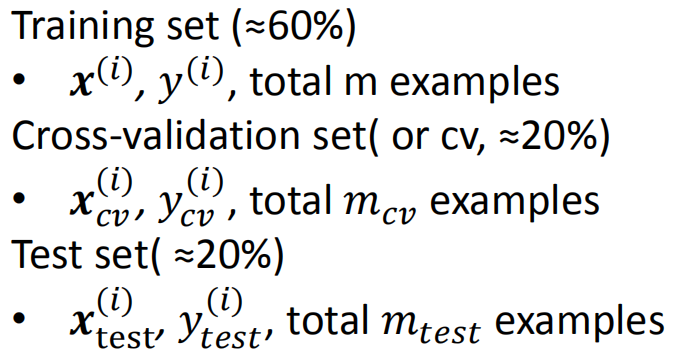


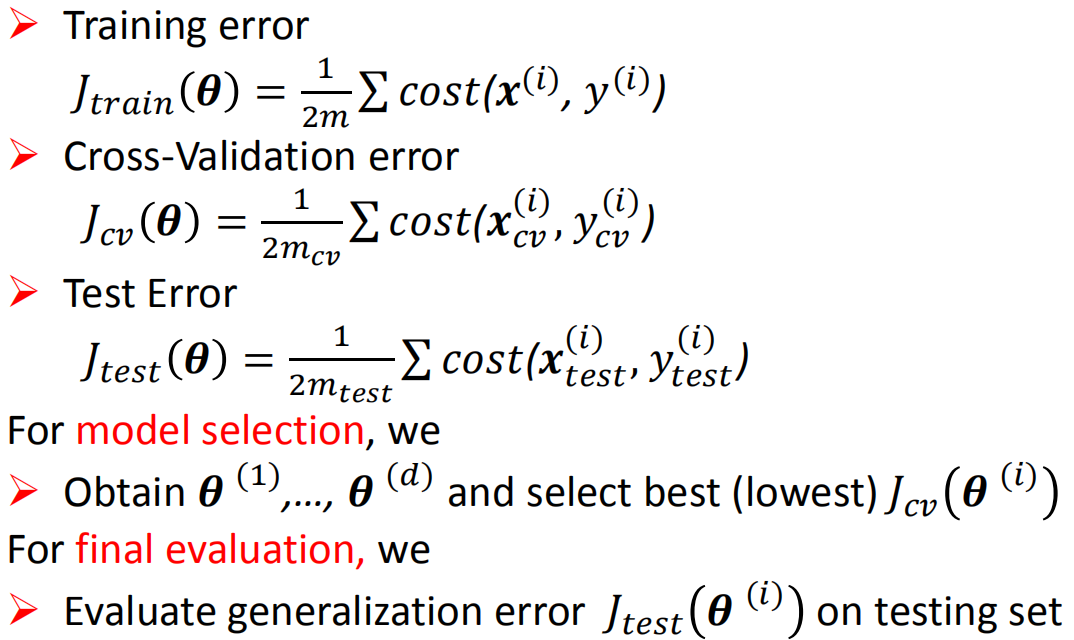
通常来说交叉评估被用来寻找一些参数的最好的值

我们仍然有训练集和测试集

但是此外我们有交叉评估集合来去测试我们模型的性能表现这个依赖于参数

不是将数据集分为两部分我们是做成了三部分：





调参：

集合分为：

训练集

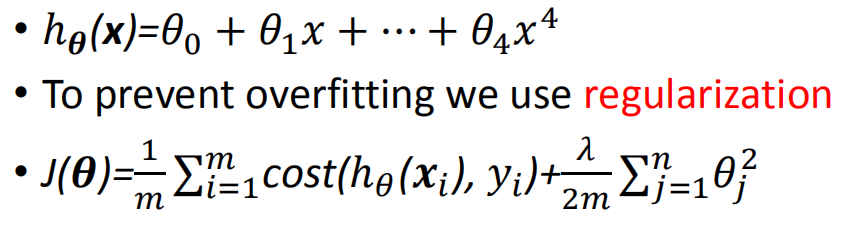
验证集

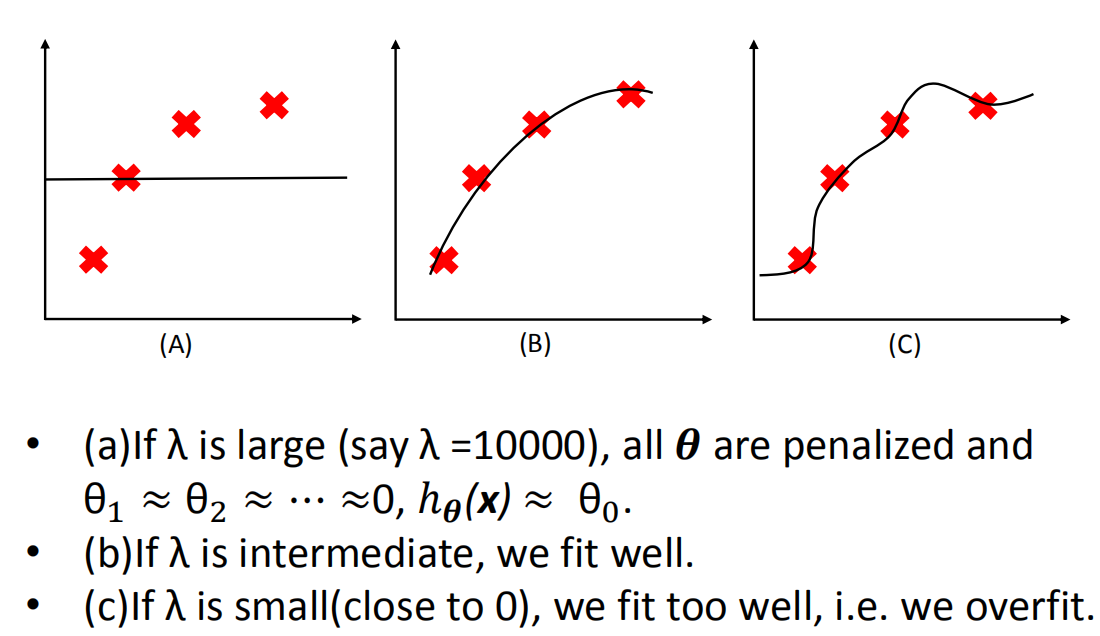
测试集

使用验证集调整超参数

使用测试集来做最后的评估

假设我们用一个高序的多项式去拟合一个模型





在验证集上选择好的验证集：

选择一个λ的区间可能性值（0.02，0.04，0.06，......，0.24）

这个给我12个模型来去检查

对于每个λi：

计算θi

计算Jcv（θi）

用最小的Jcv（θi）代入λi

最后我们汇报测试集的错误率

K-Fold Cross-Validation

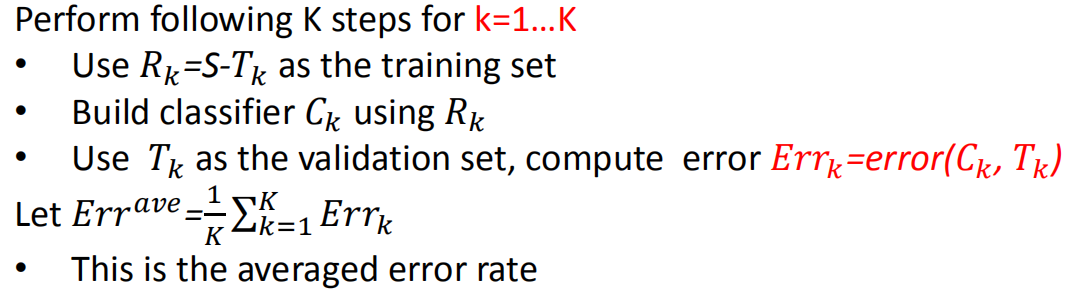
如果我们想减少数据的可变性

我们可以使用不同分区进行多轮交叉验证

然后在所有轮中平均结果

我们从总体D中采样样本S

分区数据S被分为K个等价的连接子部分



Choosing best parameter λ with K-Fold Cross-Validation

Split your data into training set and validation set

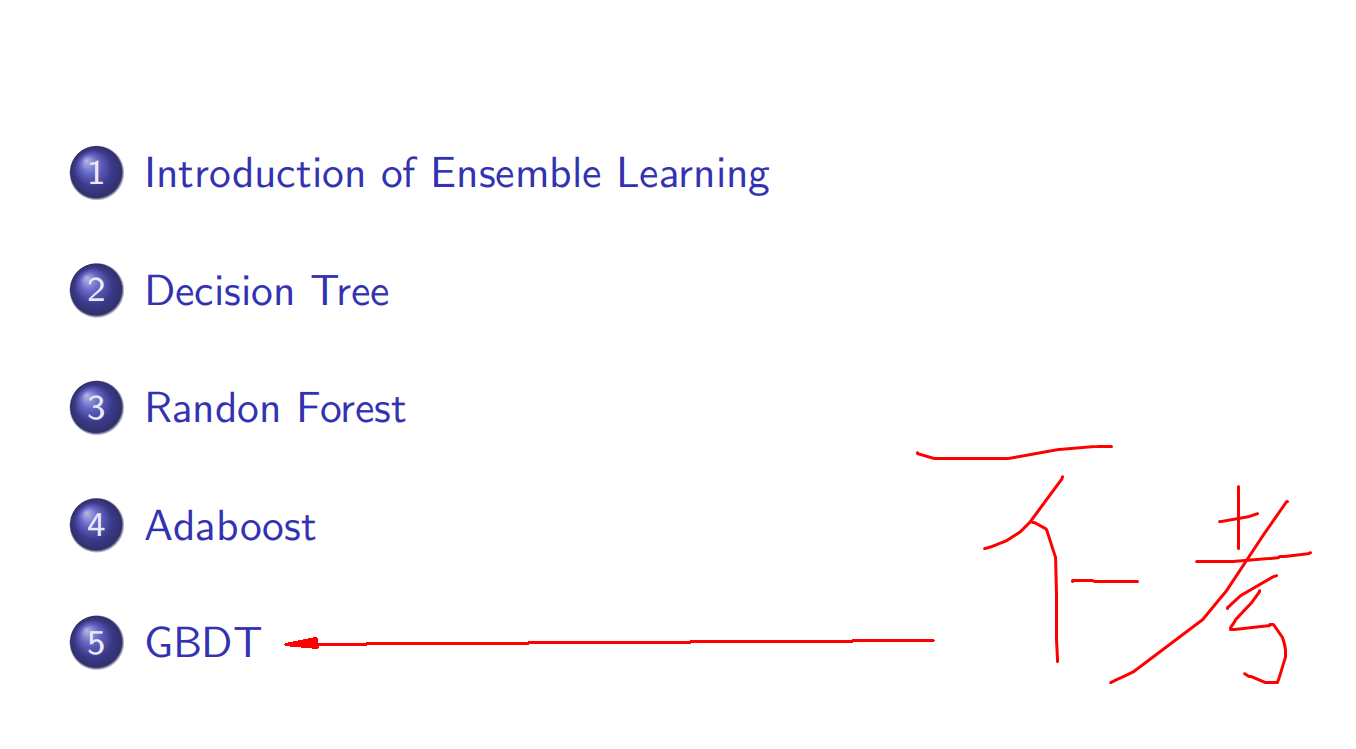
For every possible value λ, estimate the error rate

Select λ with least average error rate

最终找到测试集上面的评估

## Ensemble Methods

集群算法：

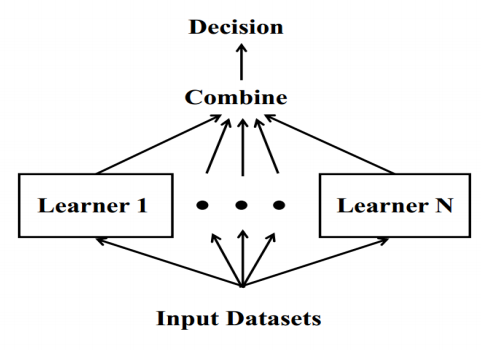


### Introduction of Ensemble Learning

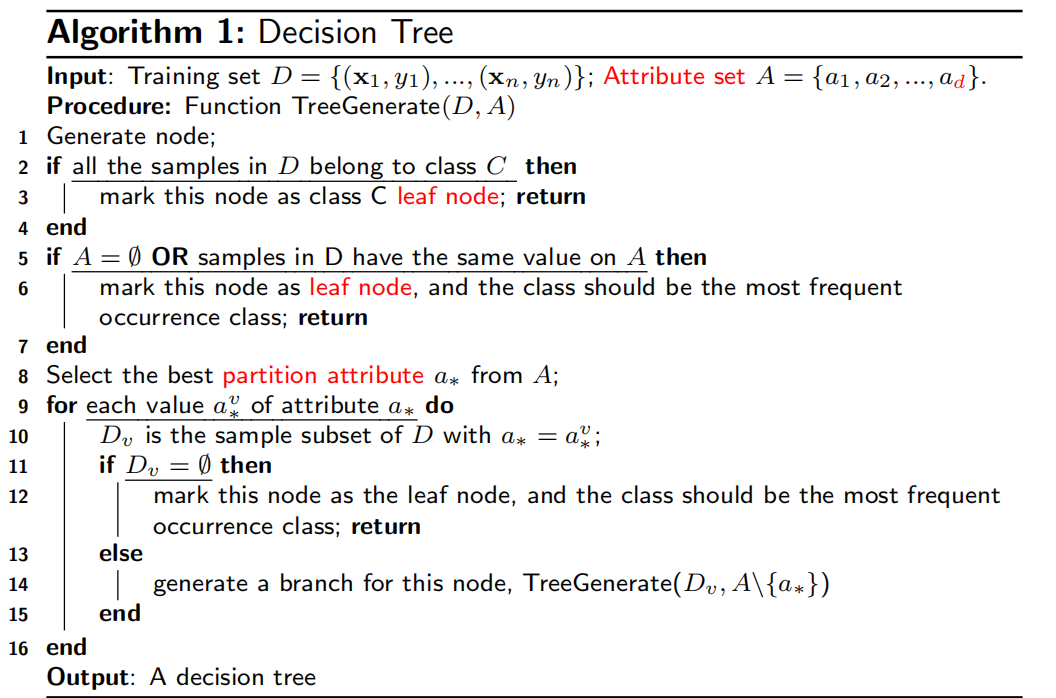
#### 集成学习

将多个弱分类器结合起来变成强分类器

主要方法：bagging，boosting



#### Decision Tree

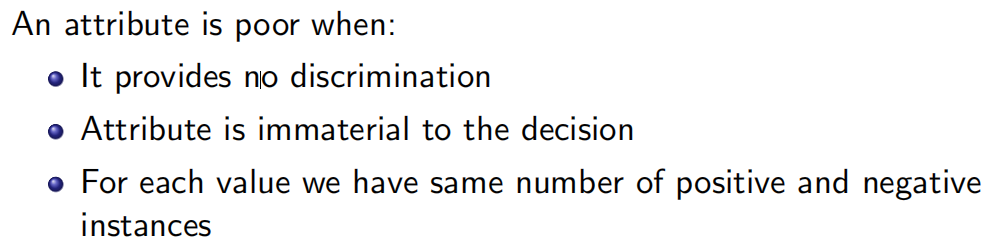


A good Attribute：

一个好的属性是：

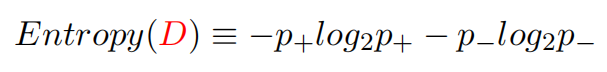
一个实例的所有值都是正的

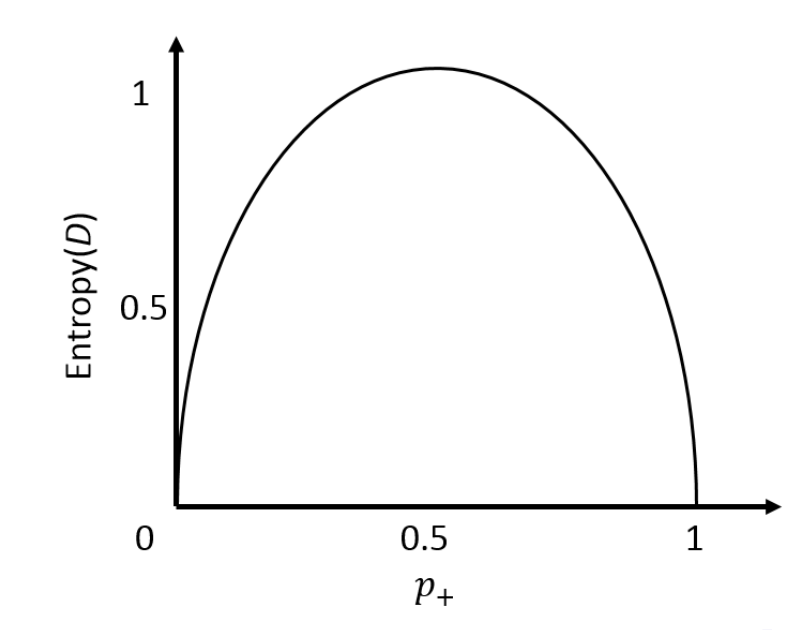
别的都是负的

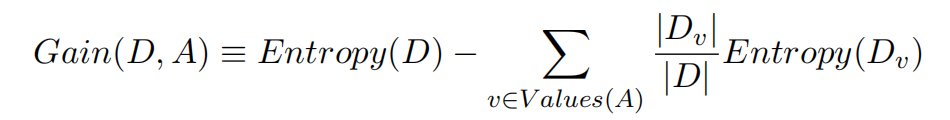


Entropy: Characterizes the (im)purity of an arbitrary collection of examples

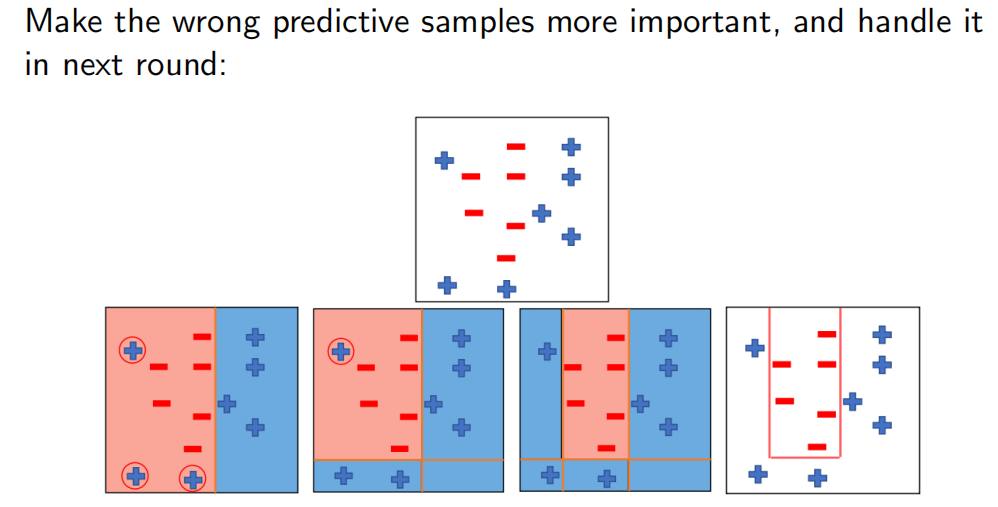
Given a collection D of positive and negative examples, entropy of D relative to boolean classification is

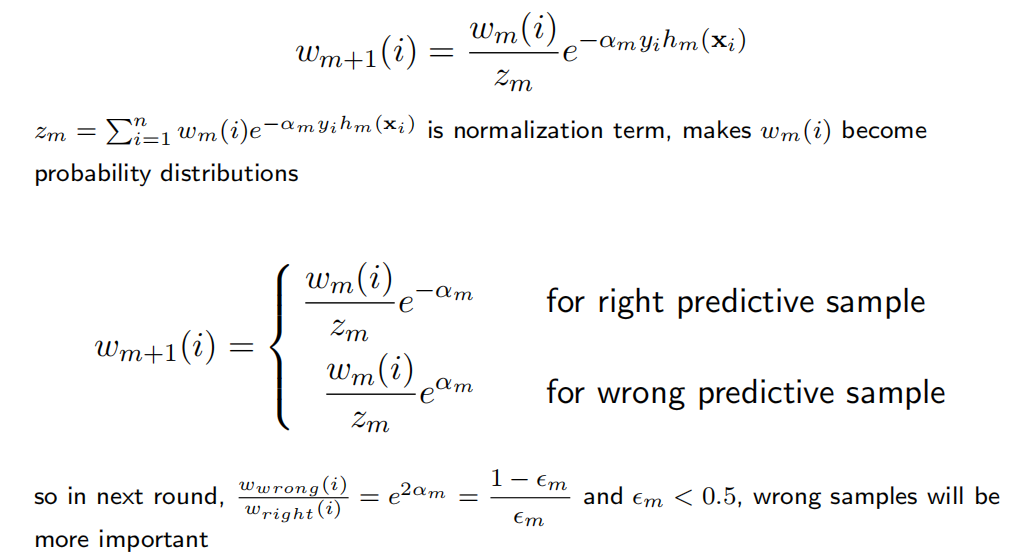






### Adaboost



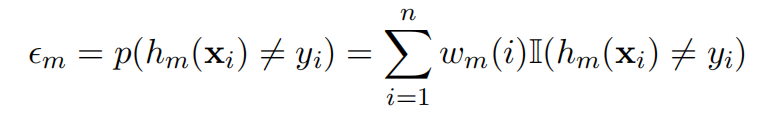


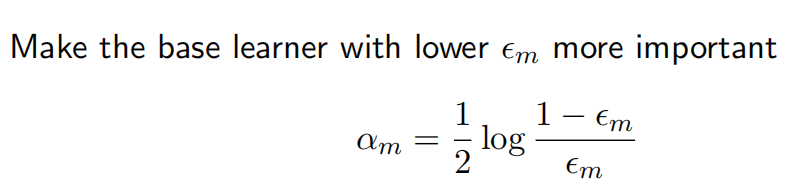
算法计算和更新

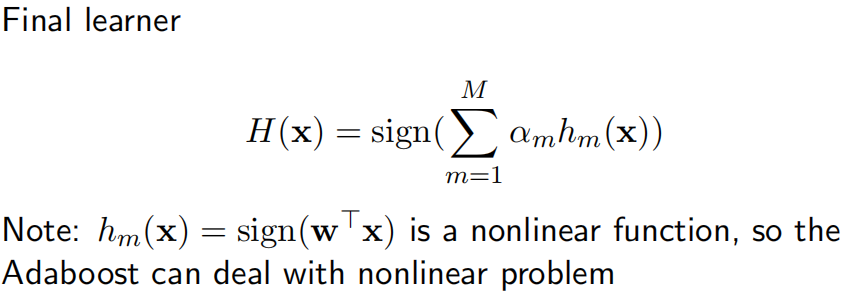
基学习器：

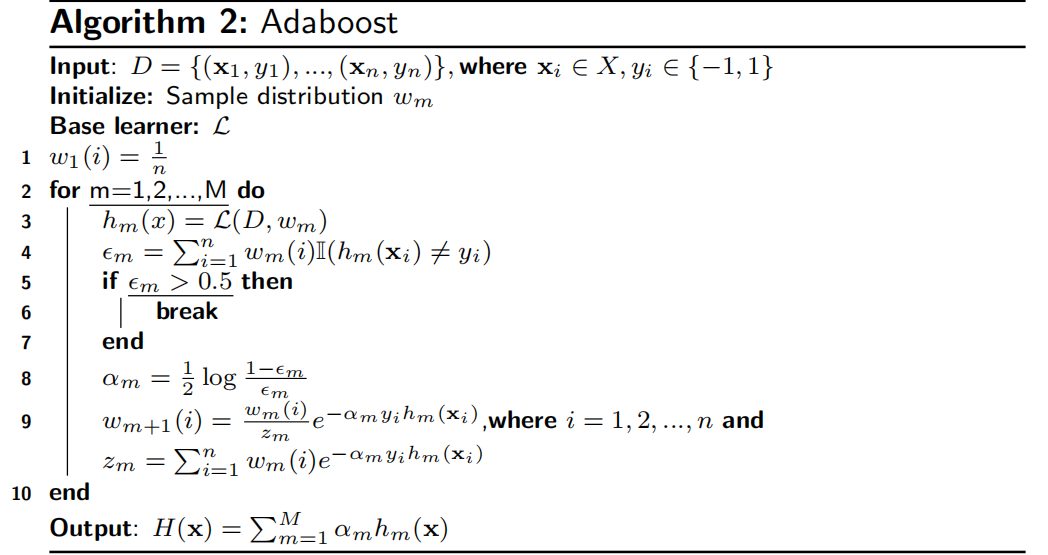


错误率：



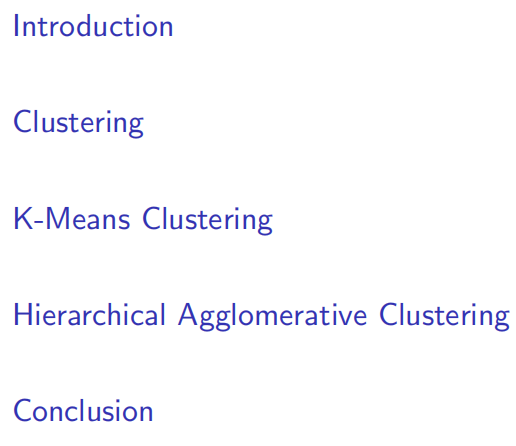






## Clustering

聚集



有监督的学习：

Data D = {(x1, y1), . . . ,(xn, yn)}



### Clustering

Simplest idea for discovering structure.

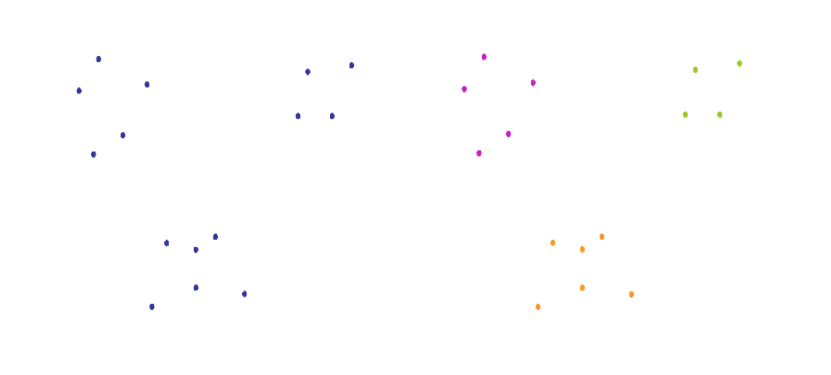
Find groups of similar examples:

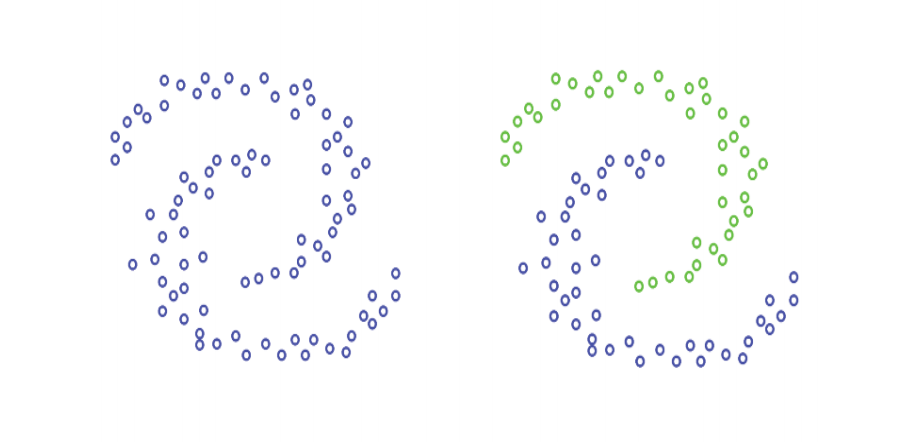
To understand the data.

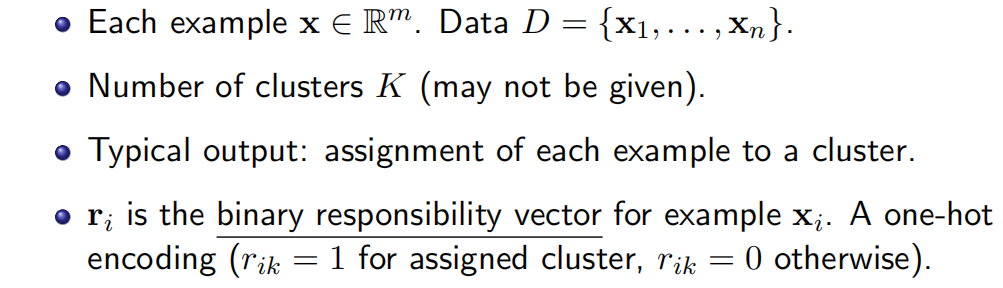
For dimensionality reduction.

To preprocess a lot of unlabeled data, find concepts to use for

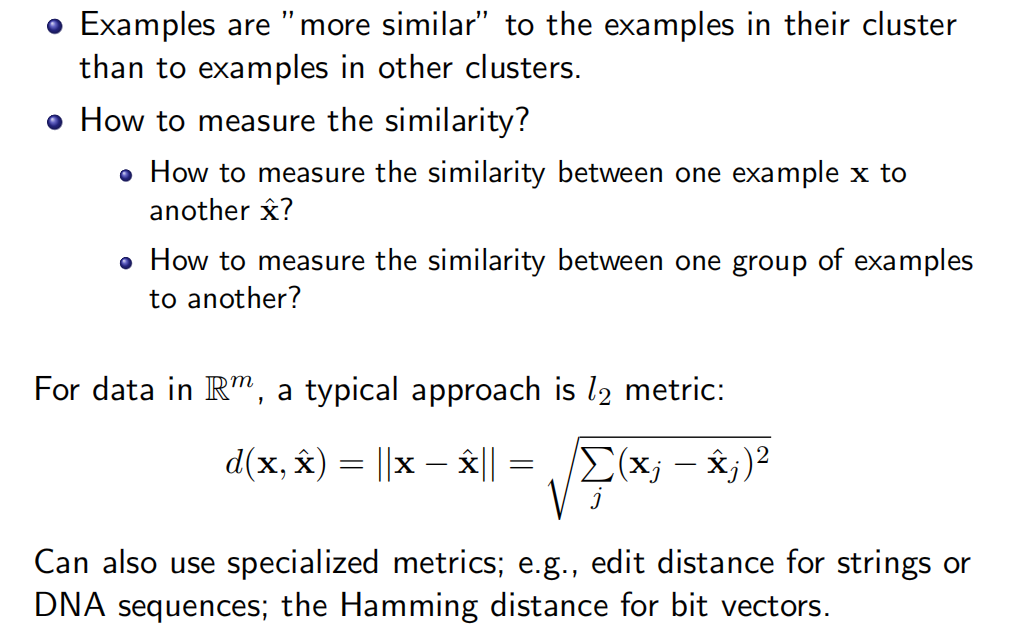
supervised learning



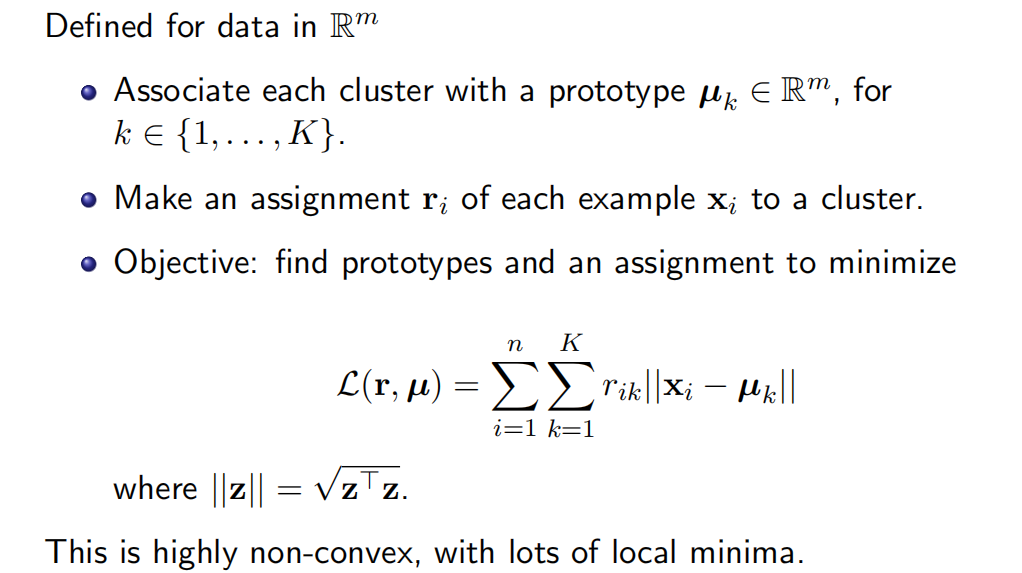


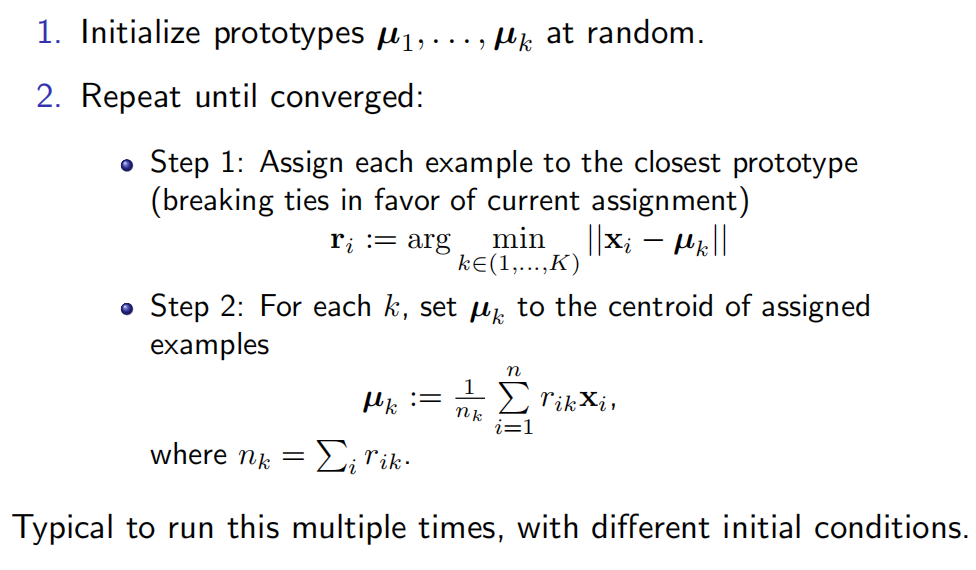


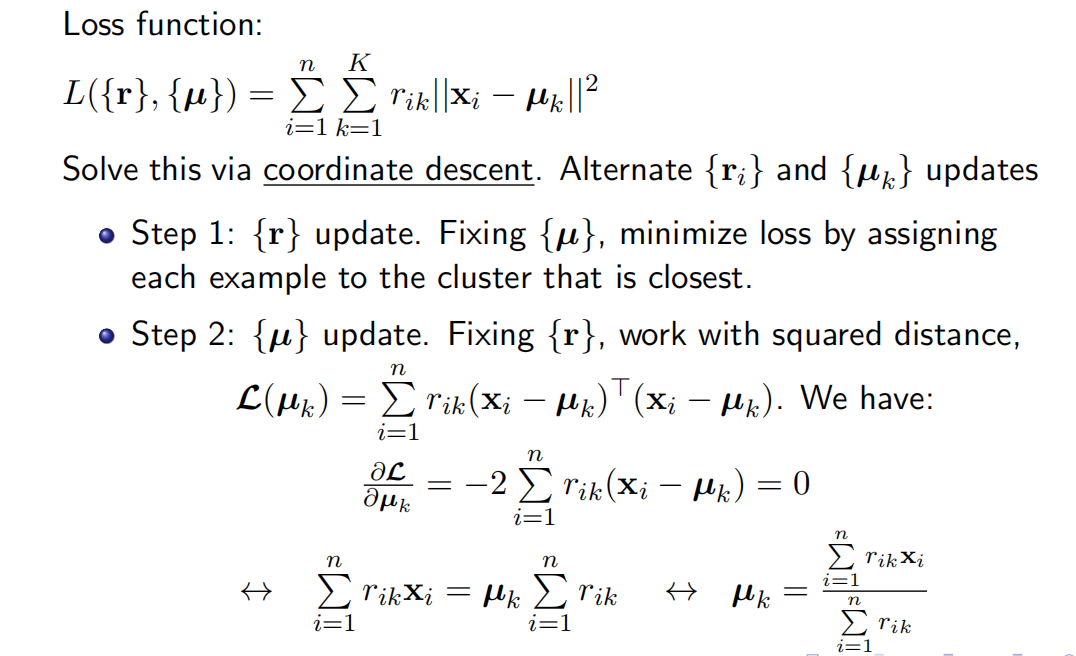
What is a good Clustering?



### K-Means Clustering





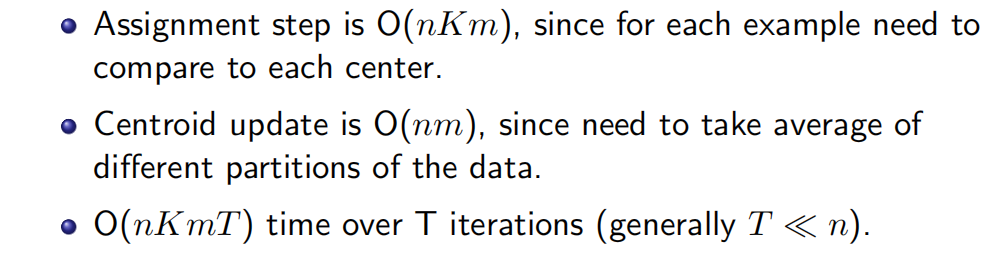


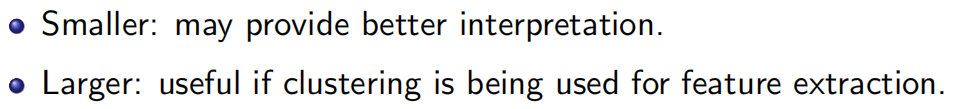
简单又流行的方法

参数

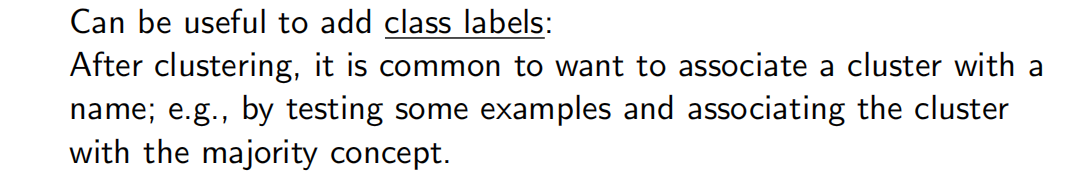
如果线性限制失败了就没有用了

计算复杂度：

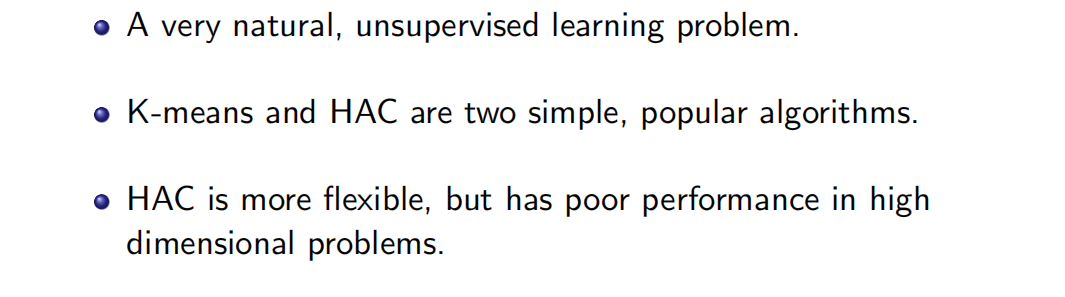




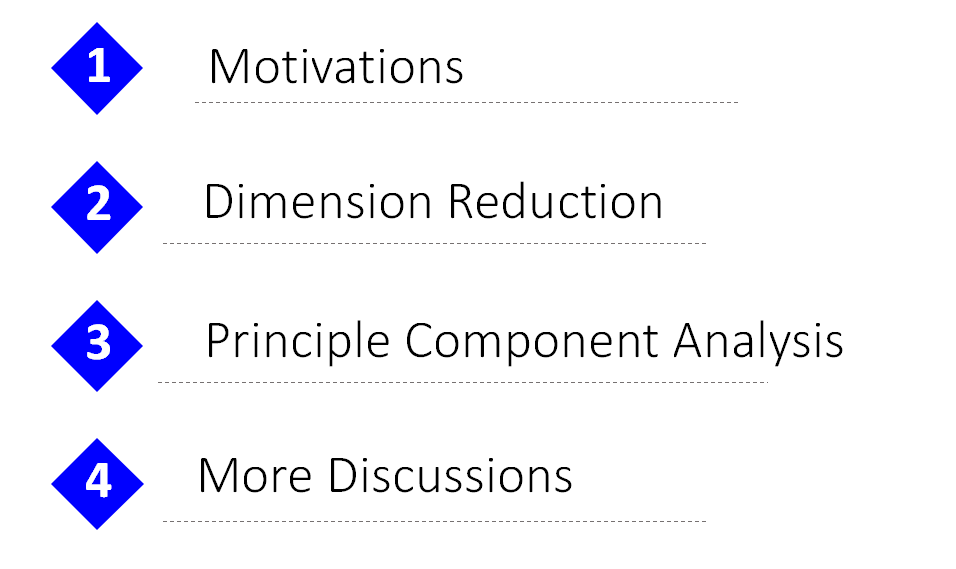
没有做这个的标准的方法，一个启发式方法是把K和损失向对抗，并且寻找一个“knee”在plot上

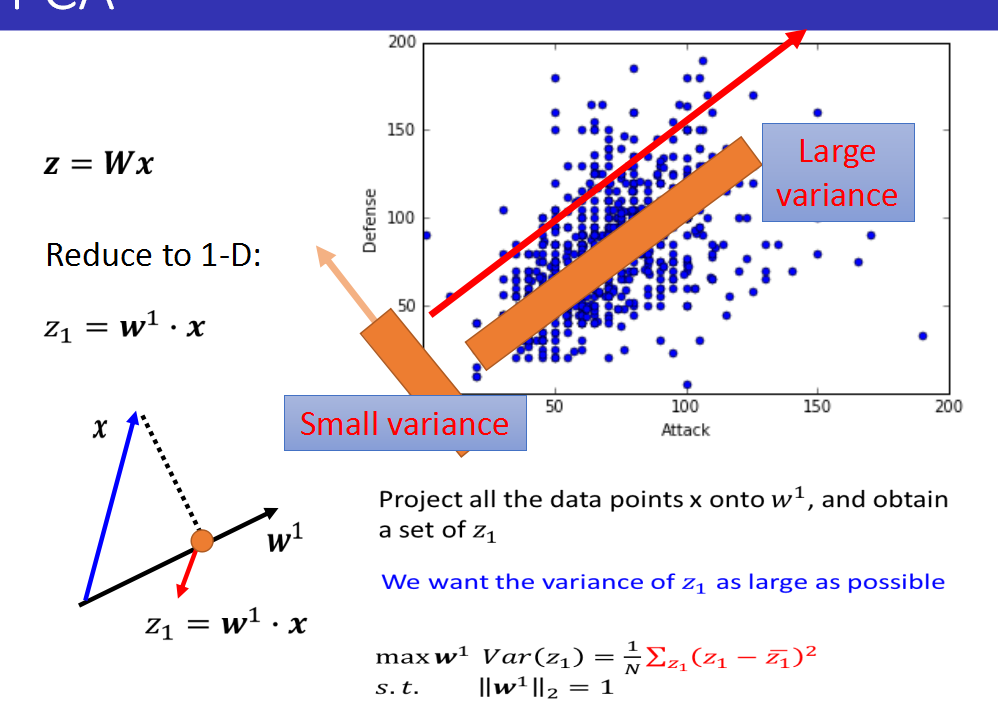


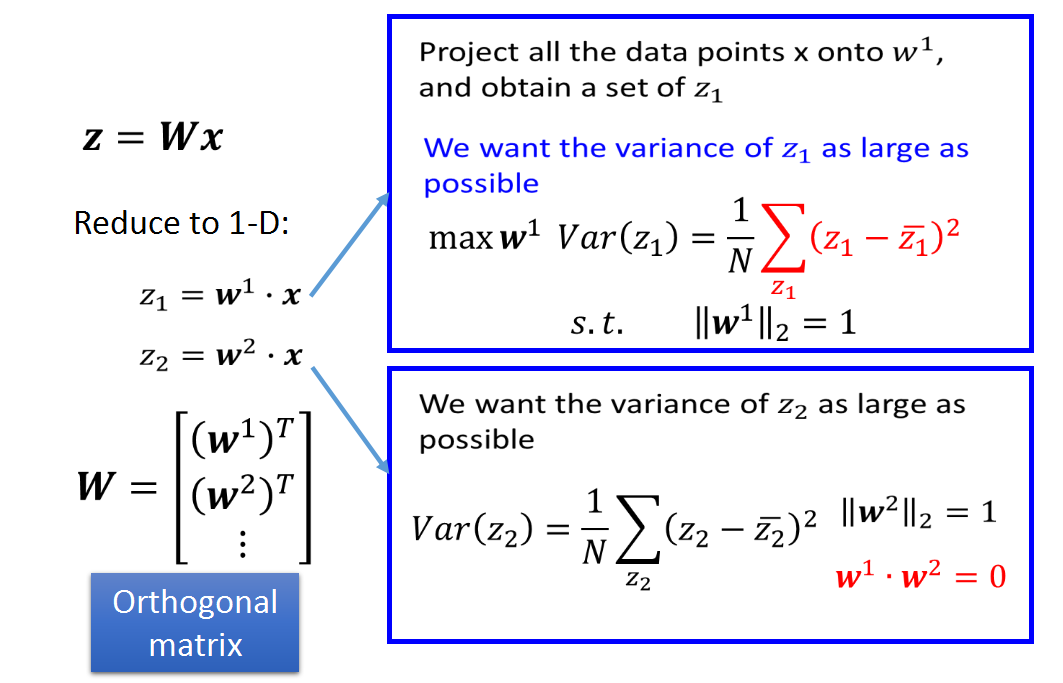
### Conclusion

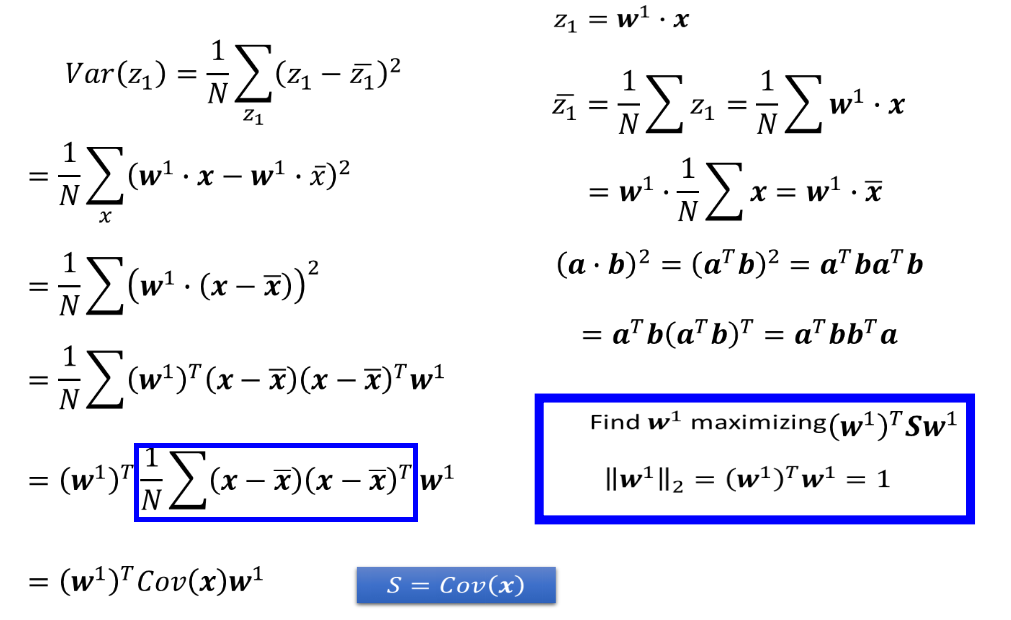


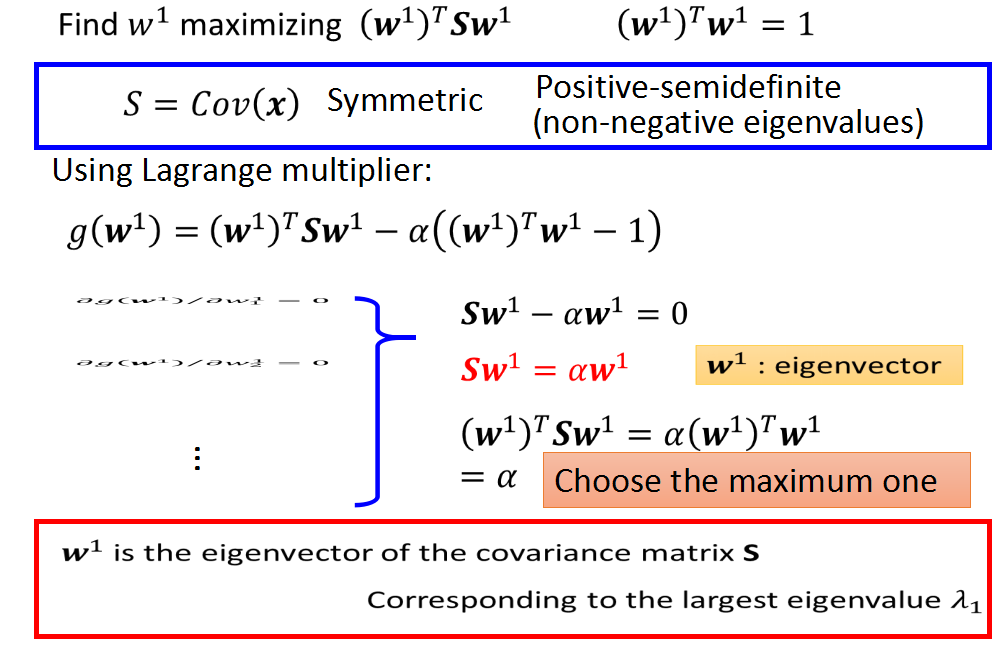
## PCA

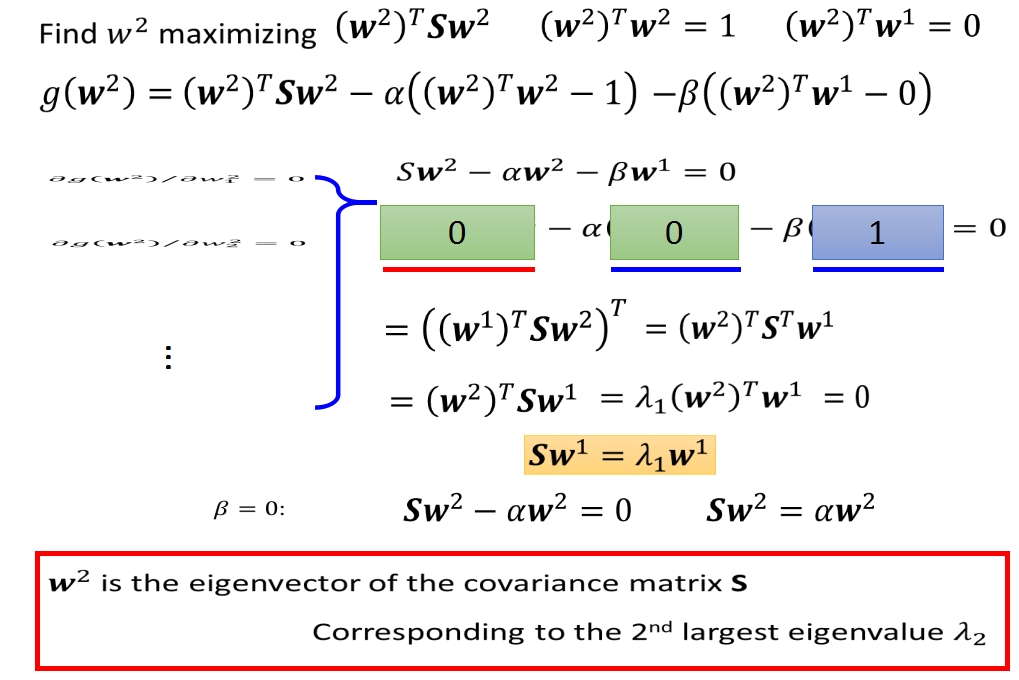


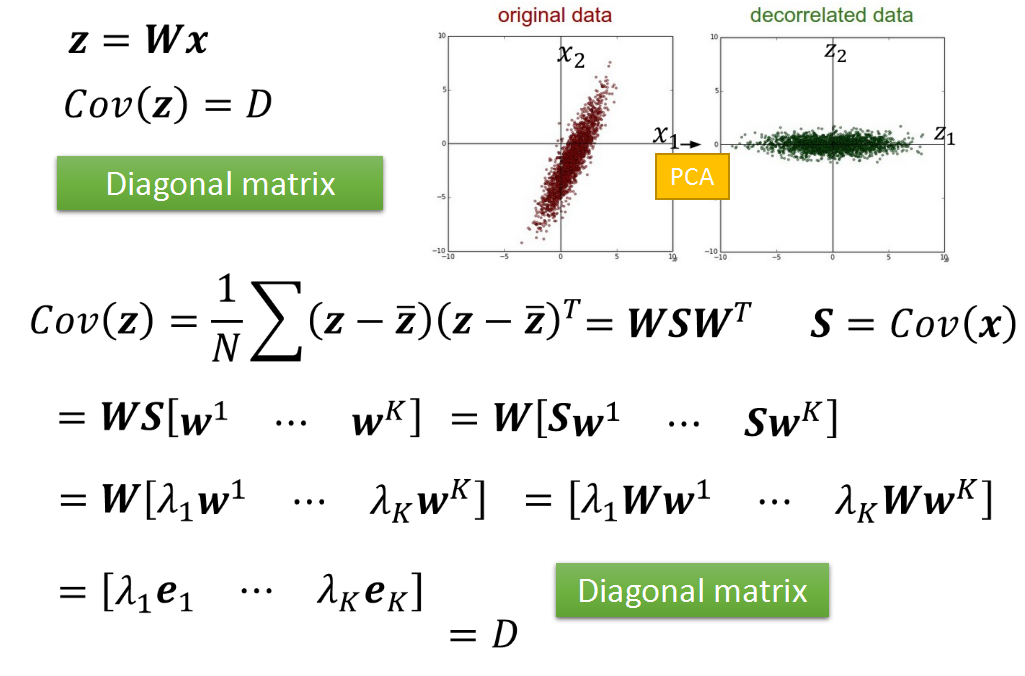


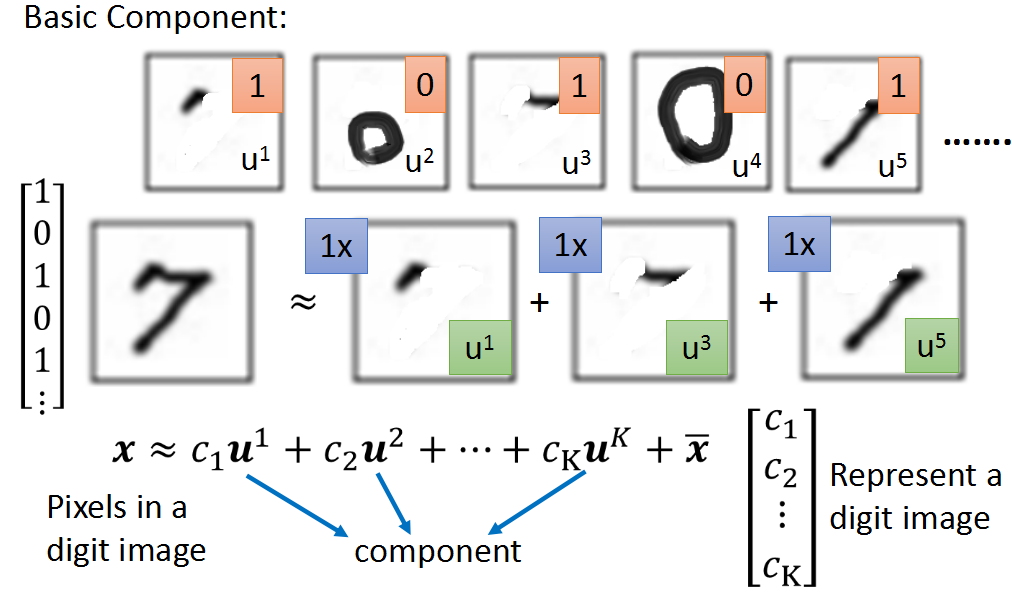


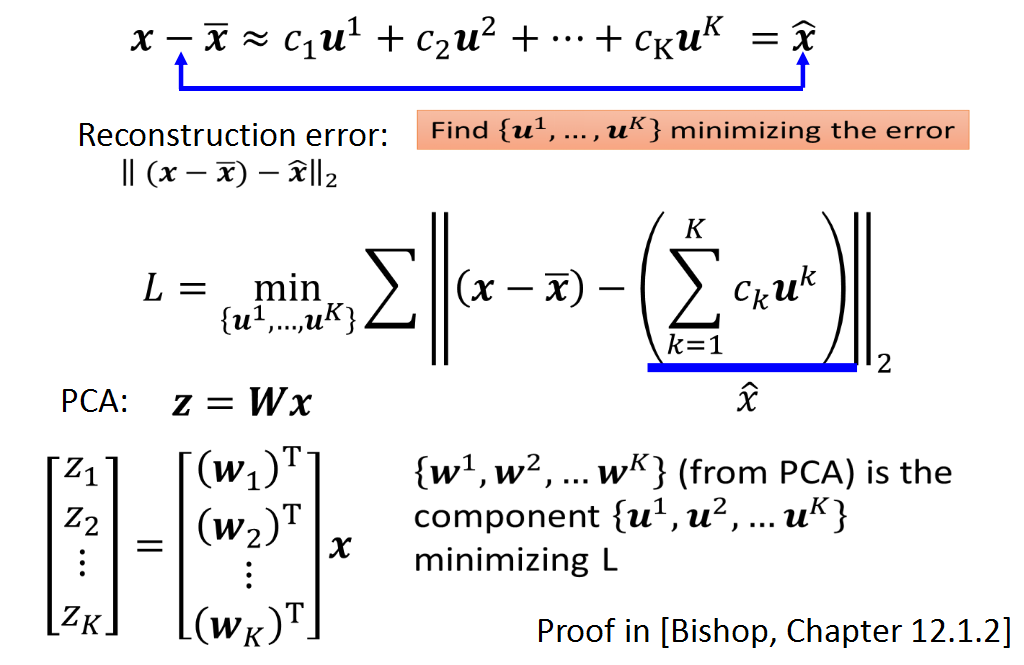


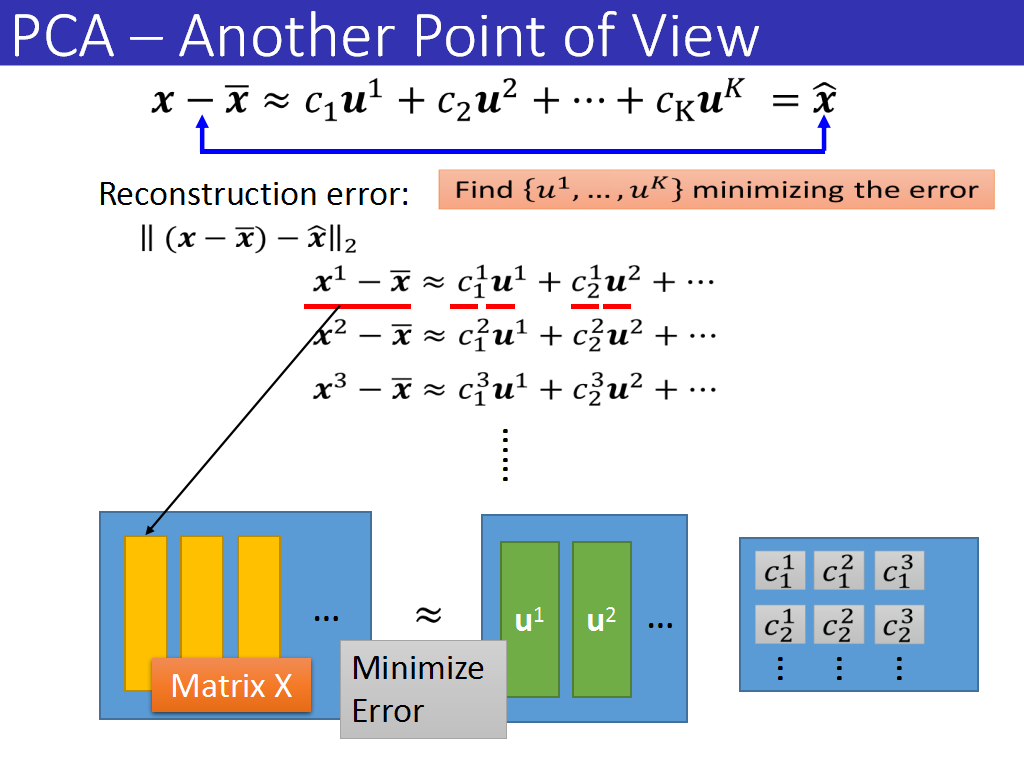


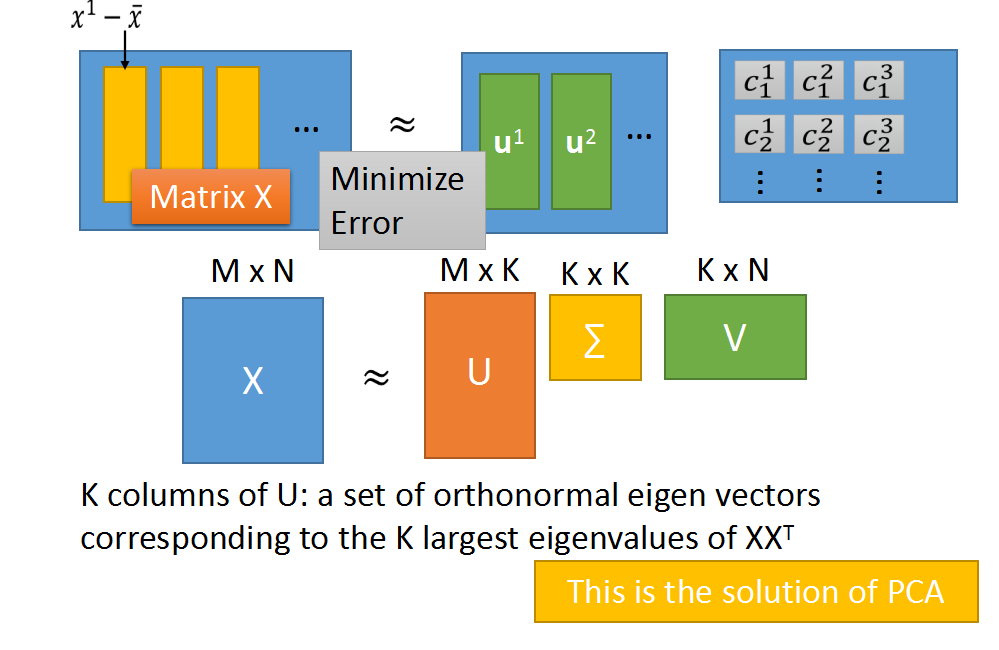










.

## Recommendation System

推荐系统

Recommender System applies statistical and knowledge discovery techniques to the problem of making product recommendations.

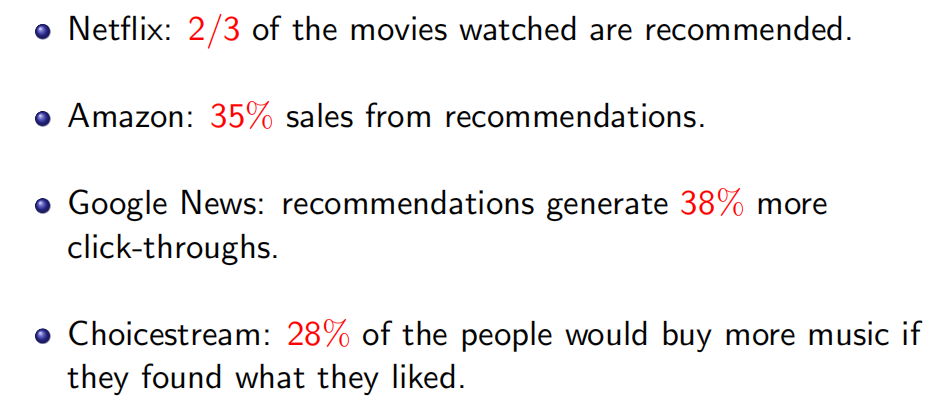
推荐系统的优缺点：

Improve conversion rate: Help customers find a product she/he wants to buy.

Cross-selling: Suggest additional products.

Improve customer loyalty: Create a value-added relationship.

推荐的值：



Collaborative Filtering：

Make automatic predictions (filtering) about the interests of a

user by collecting preferences or taste information from many

other users (collaboration)



基于内存的协作滤波器

基于用户的推荐系统

基于Item的推荐系统

混合种的推荐系统