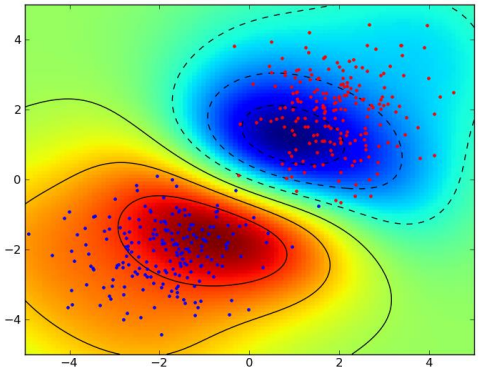
自然语言处理的机器学习基础

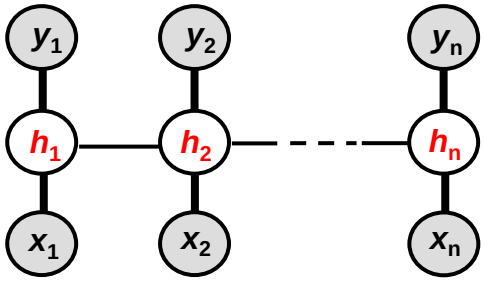
孙栩
信息科学技术学院
xusun@pku.edu.cn
http://klcl.pku.edu.cn/member/sunxu/index.htm

自然语言处理的机器学习基础

- **简单分类问题**
  - 典型问题：文本分类
  - 模型：感知器(perceptron)，SVM



- **结构化分类问题**
  - 链状结构
    - 典型问题：分词、词性标注
    - 模型：结构化感知器(structured perceptron)，HMM
  - 树状结构、图模型
    - 典型问题：句法分析
    - 模型：PCFG模型、依存句法分析模型



大纲

- **简单分类问题**
  - 感知器模型(perceptron)
  - 支持向量机模型(support vector machine, SVM)

二元分类器

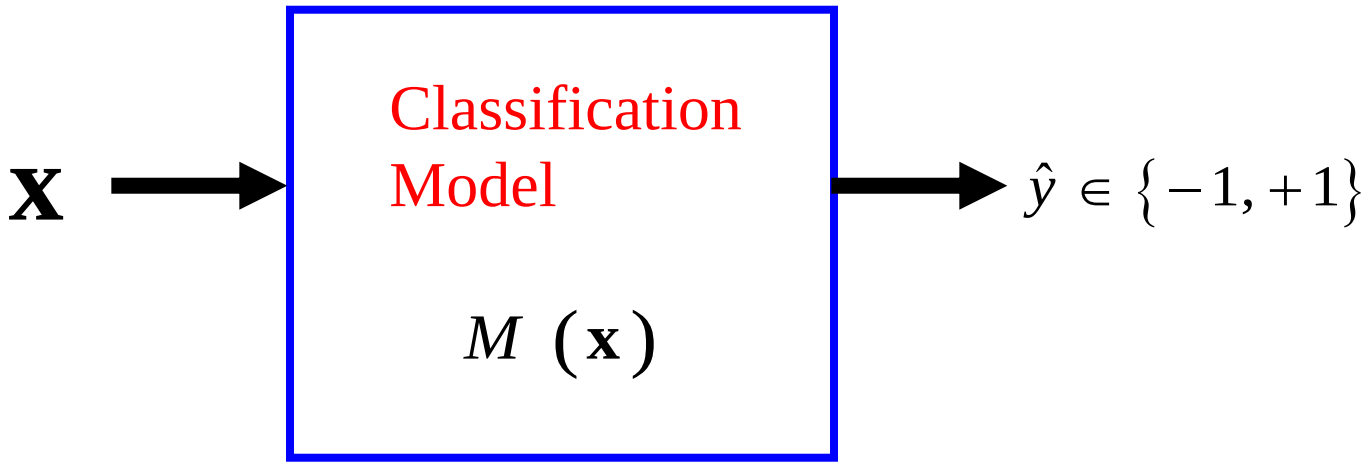
- **什么是二元分类器？**
  - 是指从一个输入数据d映射到一个输出值，且该输出值是一个二元值的映射函数/算法

- **一般的二元分类器把这两个类（二元值）设为-1和+1**
  - 其实没有严格的要求一定要把二元值设为-1和+1，但是很多二元分类器这样设置，主要是为了数学计算的方便

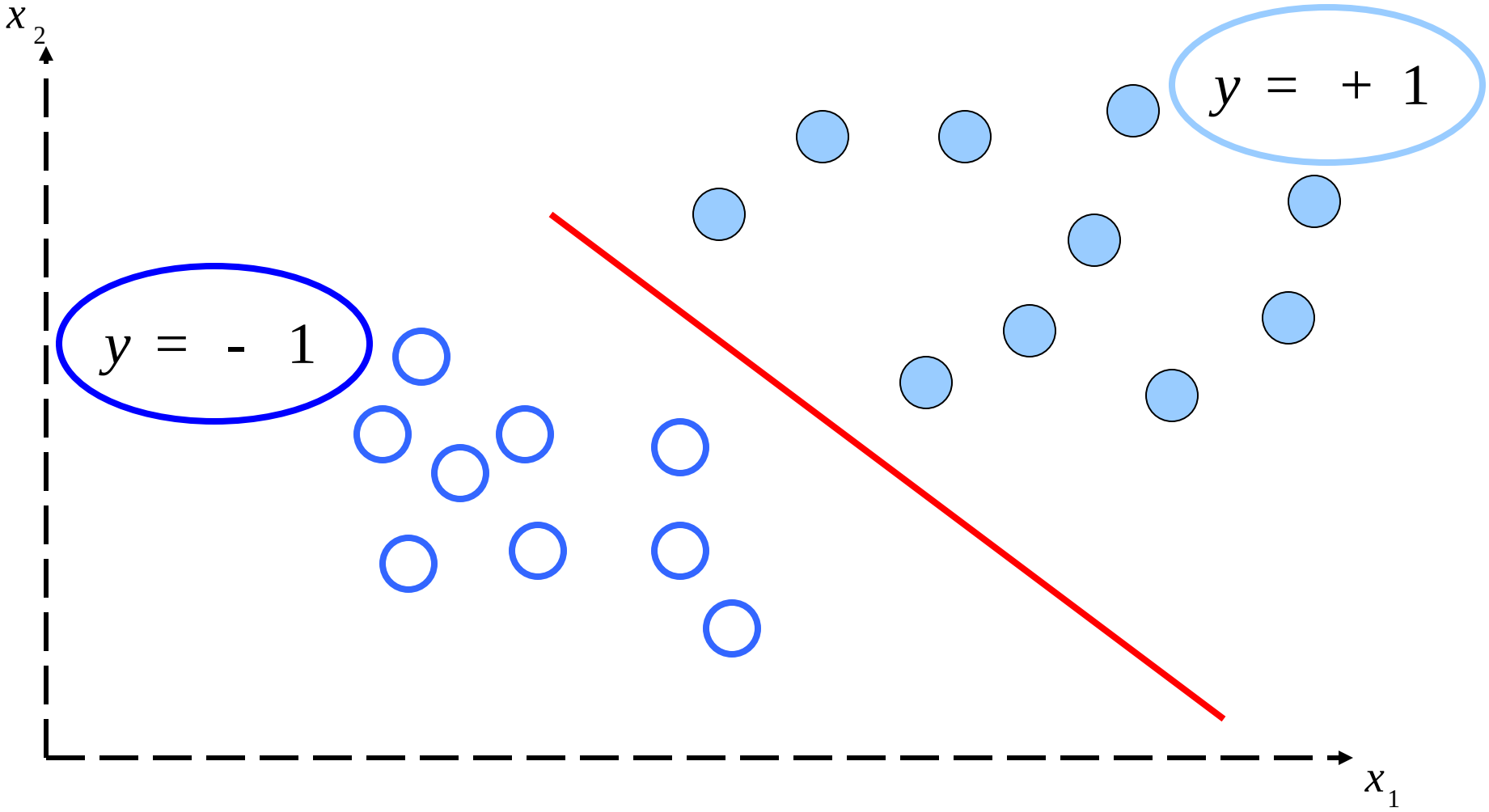
二元分类器

给定训练数据: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

二元分类器的建立:



二元分类器



感知器模型

- **感知器模型(perceptron)**

- **假设问题是线性可分的**
  - 我们需要一种学习方法，能够较快速地收敛到稳定状态，实现自动分类(classification)

- **主要思路**
  - 如果遇到一个新实例（比如句子、文本），跟原有实例(已知分类结果)相似的实例更有可能被分类为相似的类

早期思想由Rosenblatt在1950年代提出，但是现有的perceptron模型和原来早期的模型已经有了较大的不同。经过了大幅度算法改进，如今使用很广泛。

感知器模型

- **主要步骤：**
  - 随机初始化一个超平面
  - 一个接一个扫描训练数据（已经标注了正确的分类结果），基于现有的模型参数(weight vector)，计算分类结果
  - 如果分类结果正确，则继续
  - 如果分类错误，则修改模型参数，加上正确分类结果对应的特征向量，减去错误分类结果对应的特征向量
  - 如果达到收敛状态（稳定状态），则结束

主要受到神经元网络的启发

- 生物学的解释：
- 有点像大脑神经元的正向反馈和负向反馈

感知器模型

具体算法

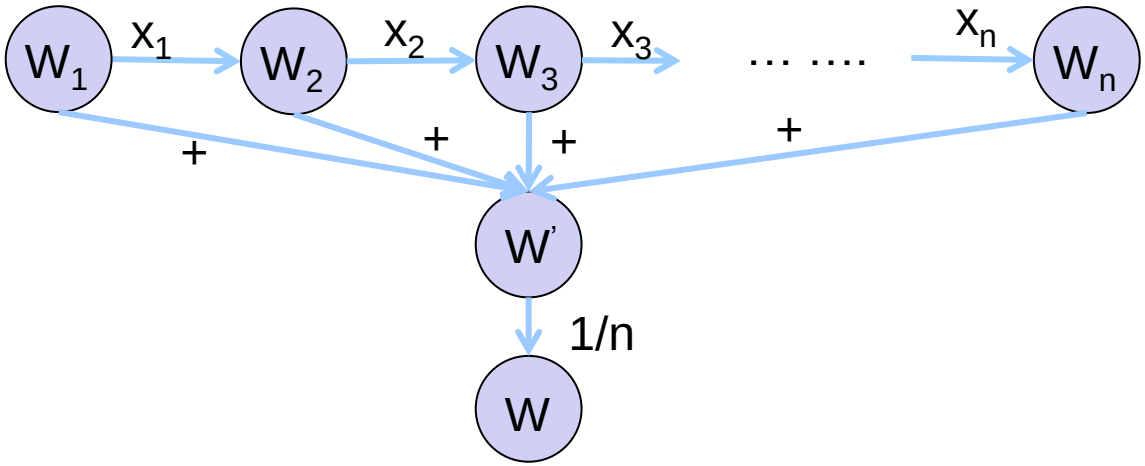
Input: *example* $\mathbf{x}_i$ *with gold standard label sequence* $\mathbf{y}_i^*$, *weight vector* $\mathbf{\Theta}$, *and feature vector* $\mathbf{f}(\mathbf{y}, \mathbf{x})$
Initialization: *set parameters* $\mathbf{\Theta}^1 = 0$
for $i = 1 \ldots d$ **do**
  $\mathbf{y}_i = \text{argmax}_{\mathbf{y}} F(\mathbf{y}|\mathbf{x}_i, \mathbf{\Theta}^i)$
  if $\mathbf{y}_i \neq \mathbf{y}_i^*$ **then**
    $\mathbf{\Theta}^{i+1} = \mathbf{\Theta}^i + \mathbf{f}(\mathbf{y}_i^*, \mathbf{x}_i) - \mathbf{f}(\mathbf{y}_i, \mathbf{x}_i)$
  else
    $\mathbf{\Theta}^{i+1} = \mathbf{\Theta}^i$
Output: *parameter vectors* $\mathbf{\Theta}^{i+1}$ *for* $i = 1 \ldots d$

- 感知器模型是基于简单的加减法！
  - 优点一：非常容易实现
  - 优点二：而且实际效果好

平均感知器(Averaged Perceptron)

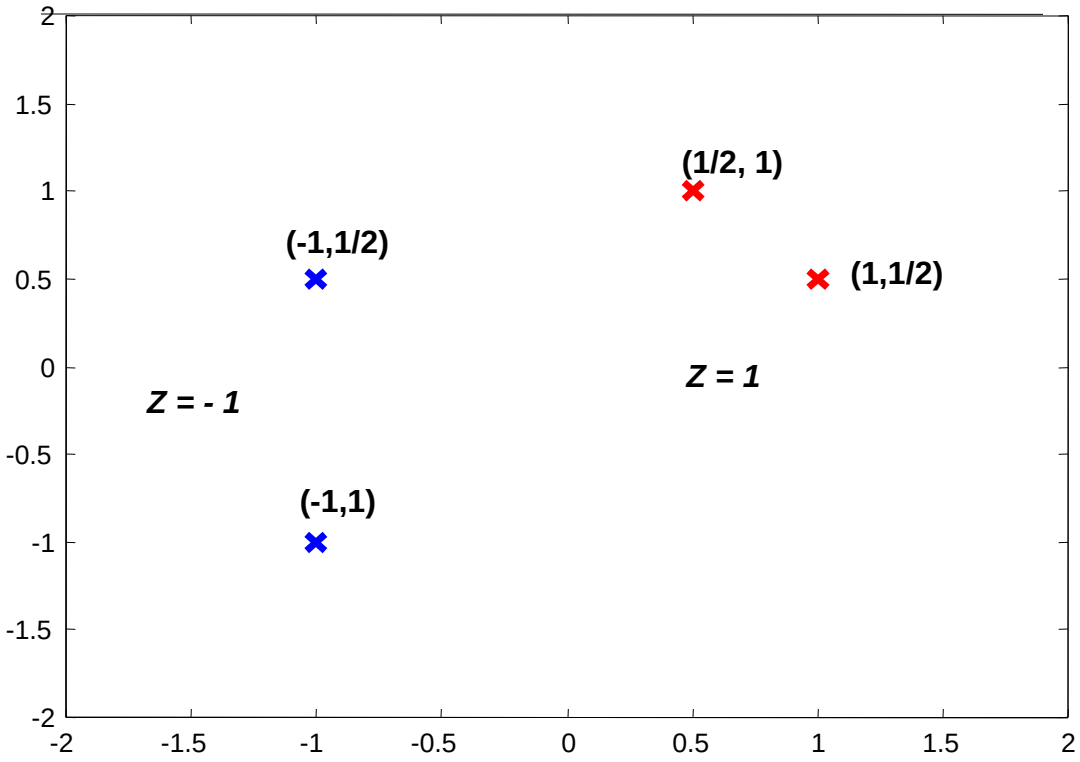
- **平均感知器:** 每次训练样本X_i后，保留先前训练权值W_i, 训练结束后平均所有权值即$W = \sum_{i=1}^{n} W_i/n$，最终用平均权值作为最终判别准则的权值。

- **参数平均化可以由于学习速率过大所引起的训练过程中出现的震荡现象。**

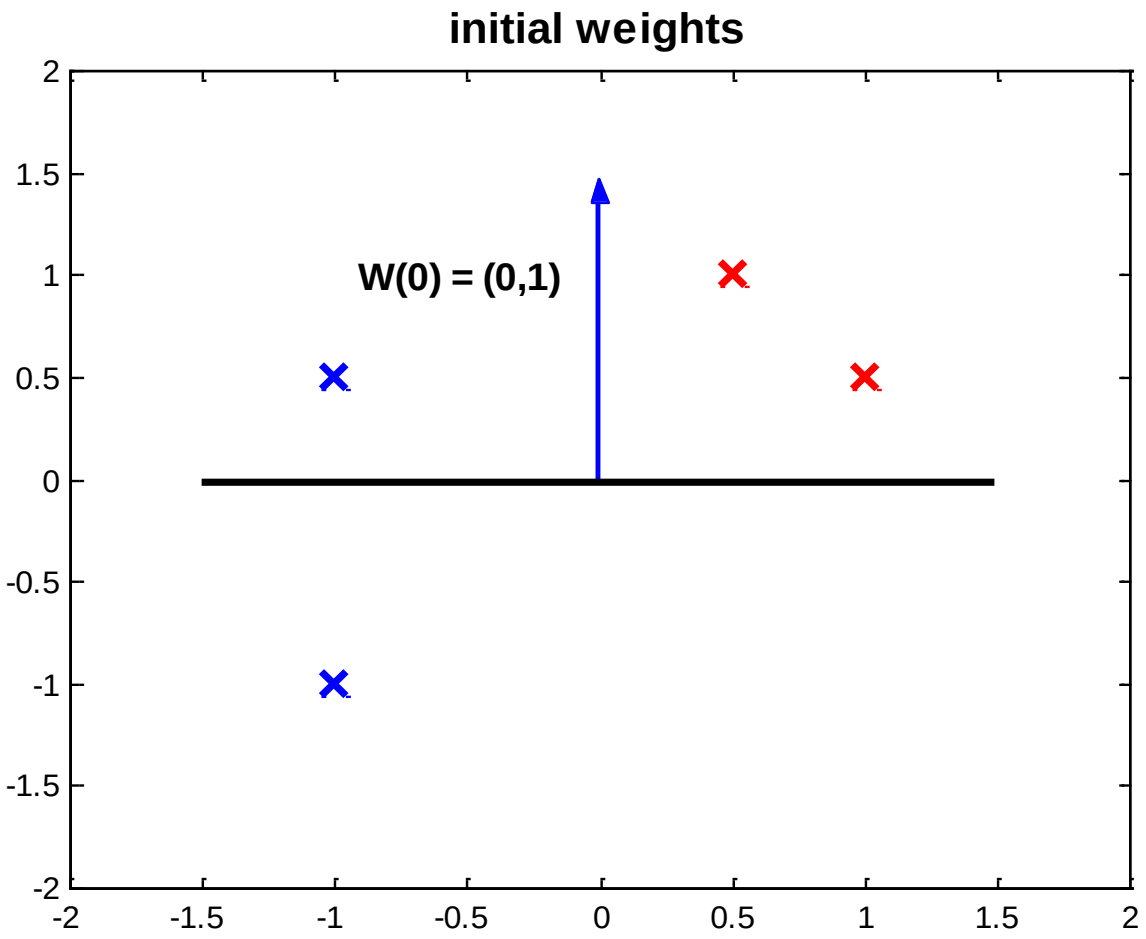


举例说明：为什么简单的加减法可以实现线性分类

4个线性可分(linearly separable)的数据点



举例说明：为什么简单的加减法可以实现线性分类

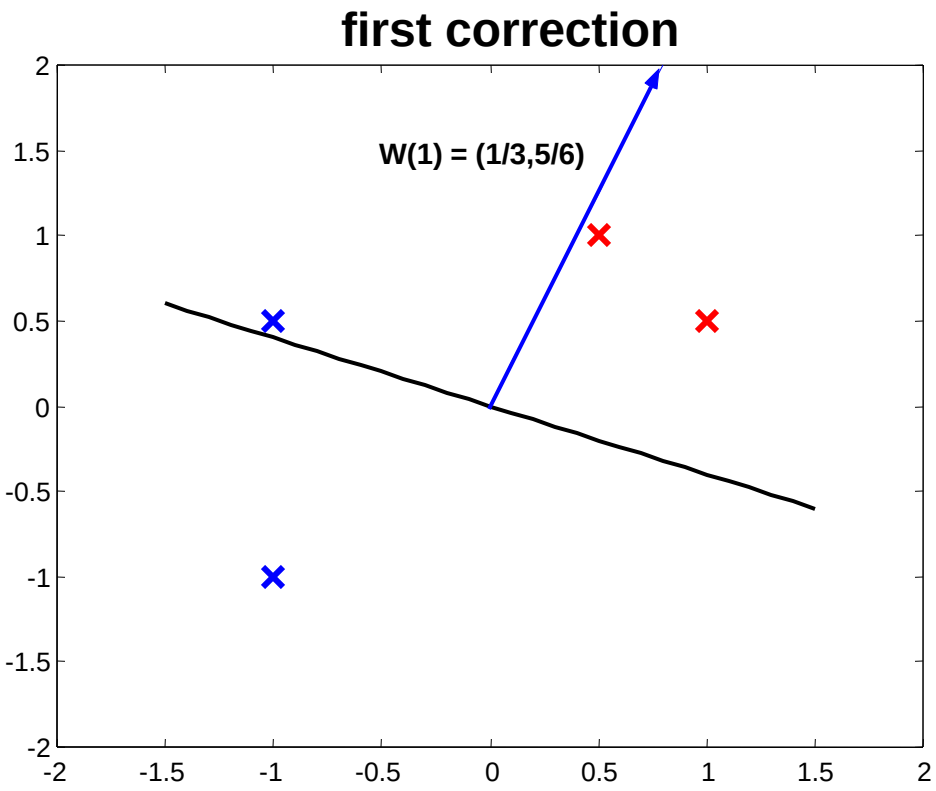


举例说明：为什么简单的加减法可以实现线性分类

参数的更新

- 左上角的数据点被错误分类了
- eta = 1/3 , W(0) = (0,1)
- W ==>W + eta * Z * X
- W_x = 0 + 1/3 *(-1) * (-1) = 1/3
- W_y = 1 + 1/3 * (-1) * (1/2) = 5/6
- W(1) = (1/3,5/6)

举例说明：为什么简单的加减法可以实现线性分类

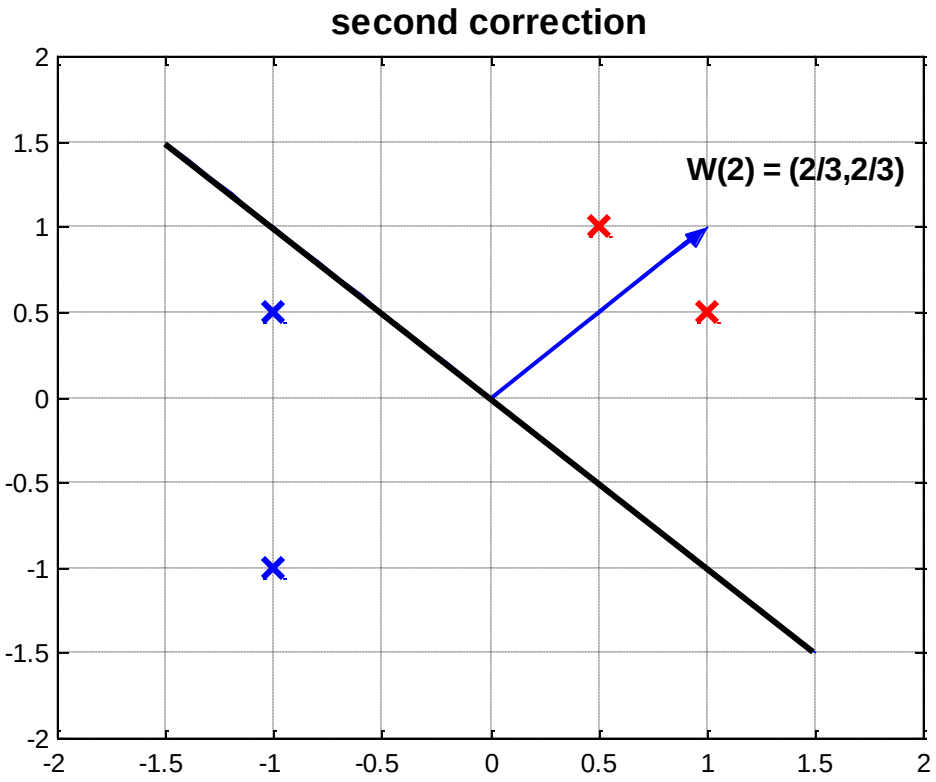


举例说明：为什么简单的加减法可以实现线性分类

- **参数的更新(继续)**

- 左上角的数据点还是被错误分类了
- W ==>W + eta * Z * X
- W_x = 1/3 + 1/3 *(-1) * (-1) = 2/3
- W_y = 5/6 + 1/3 * (-1) * (1/2) = 4/6 = 2/3
- W(2) = (2/3,2/3)

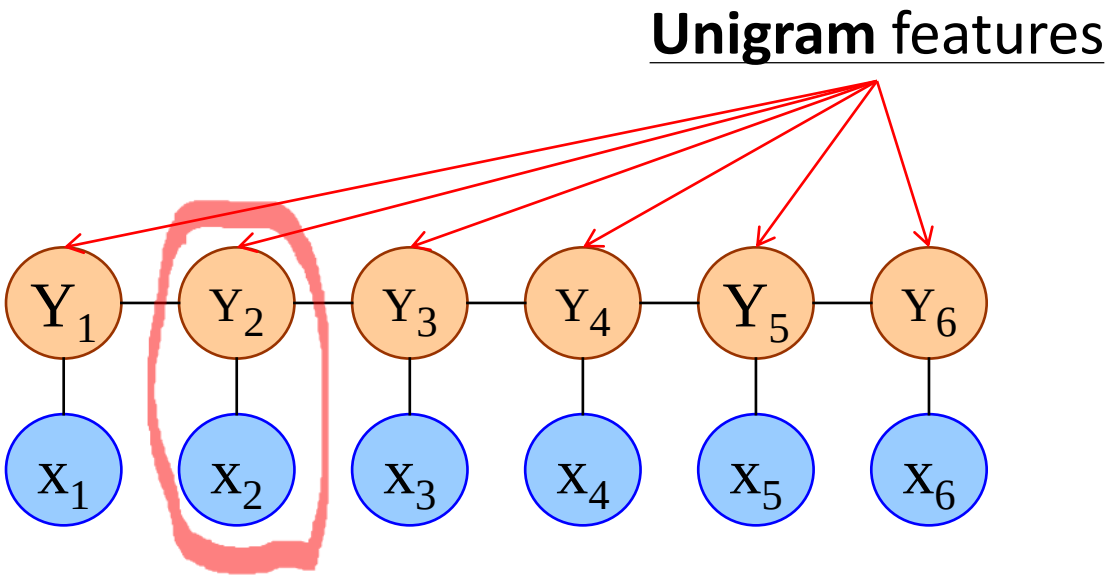
举例说明：为什么简单的加减法可以实现线性分类



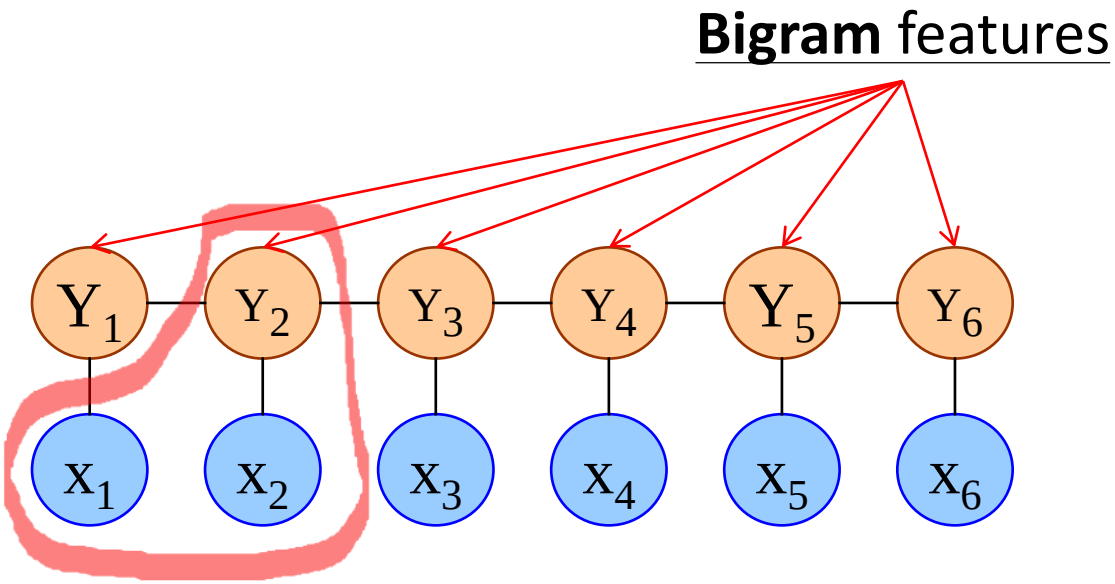
举例说明：为什么简单的加减法可以实现线性分类

- 通过（基于数据点的）简单的加减法可以有效学习到“正确的”模型参数
- 最后，4个数据点都可以正确分类
- 在这个实例里，只需要更新2次参数
- 在几何拓扑的层面上，参数的更新其实就是“分类超平面”的转向

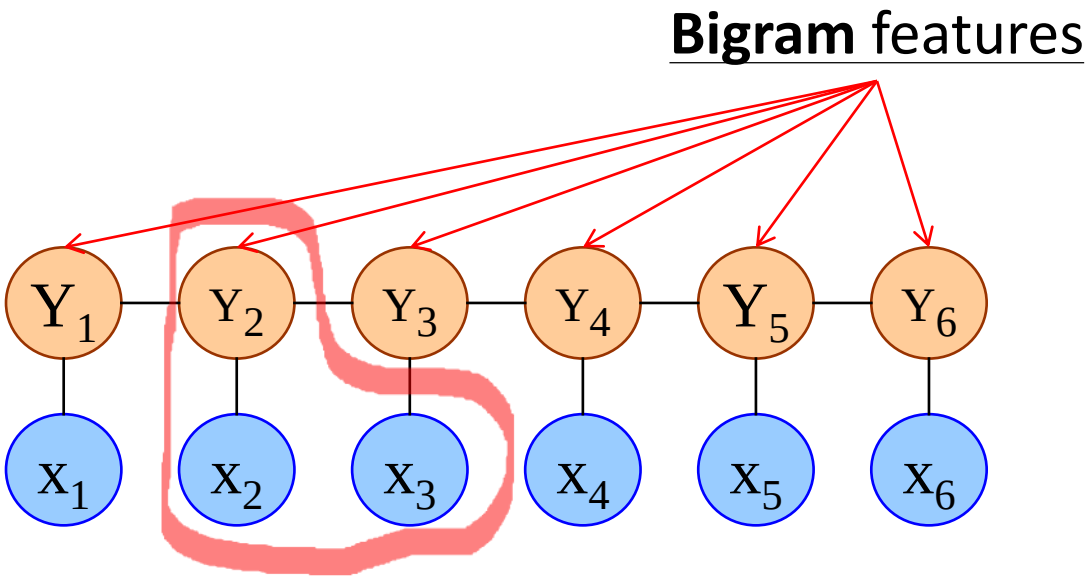
怎么提取相关特征(Feature)



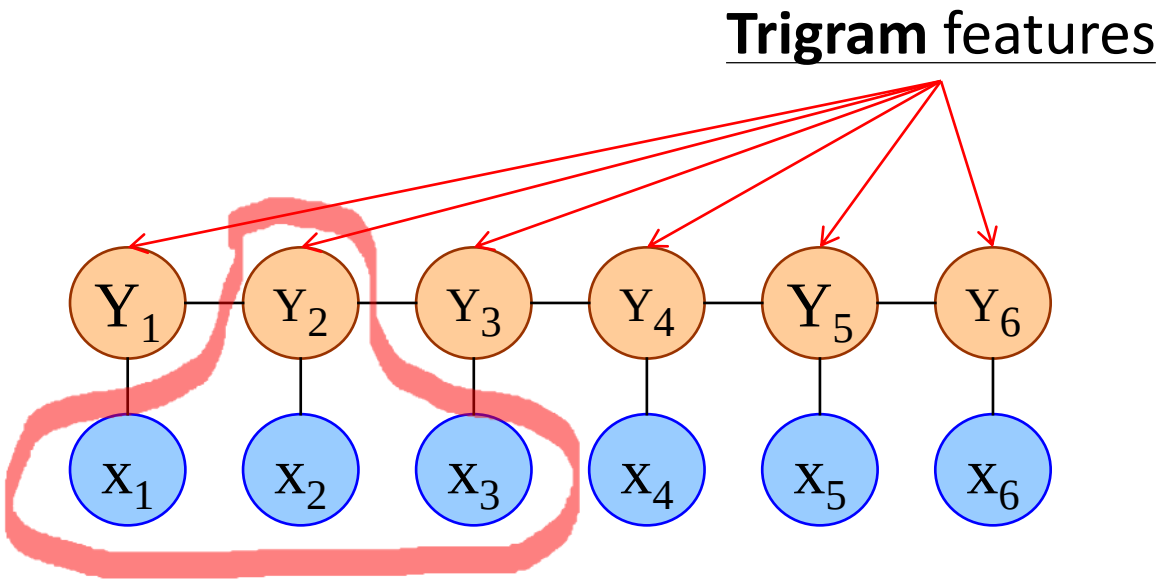
怎么提取相关特征(Feature)



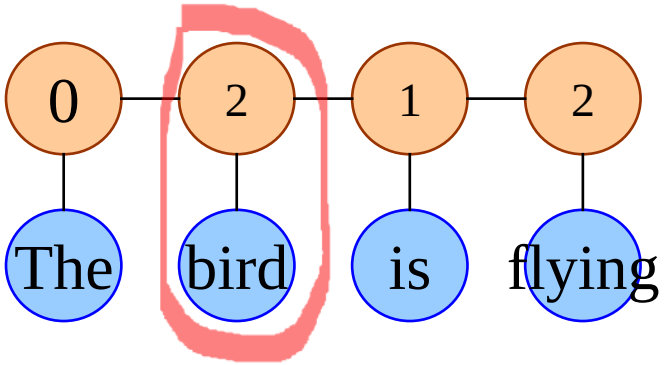
怎么提取相关特征(Feature)



怎么提取相关特征(Feature)



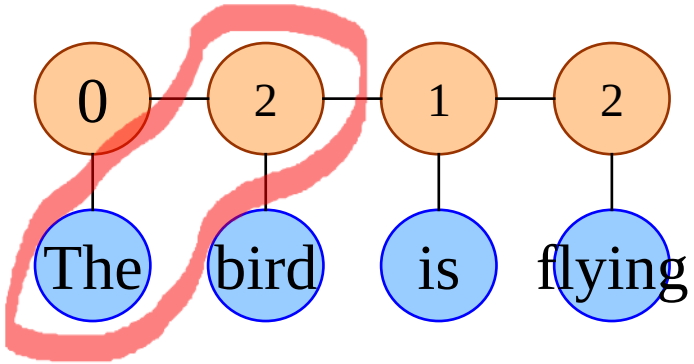
特征提取举例



当y=2以及x=bird，特征**f(y, x)**示例如下：

Unigram: bird_2

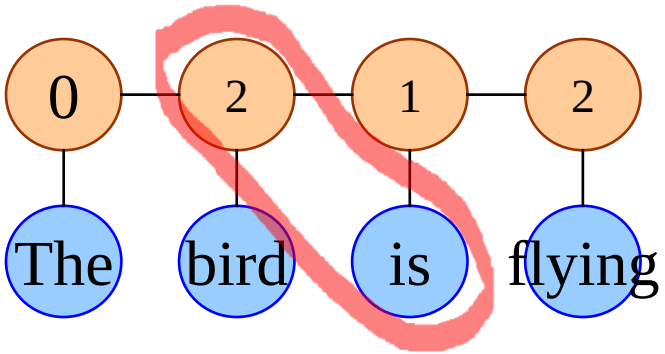
特征提取举例



当y=2以及x=bird，特征**f(y, x)**示例如下：

Unigram: bird_2，The_2

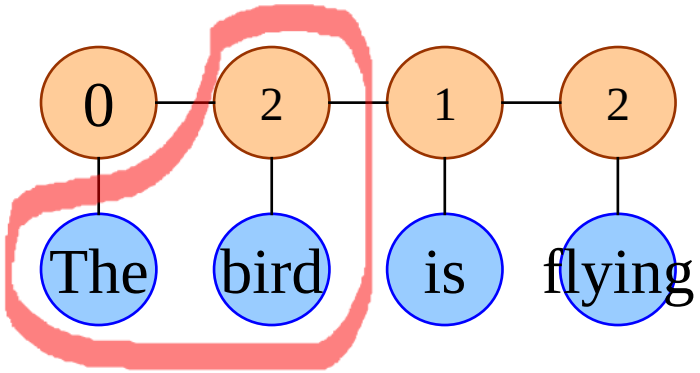
特征提取举例



当y=2以及x=bird，特征**f(y, x)**示例如下：

Unigram: bird_2，The_2，is_2

特征提取举例

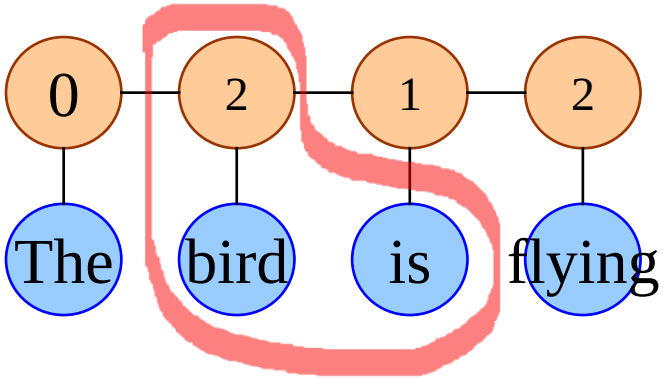


当y=2以及x=bird，特征**f(y, x)**示例如下：

Unigram: bird_2，The_2，is_2

Bigram: The_bird_2

特征提取举例

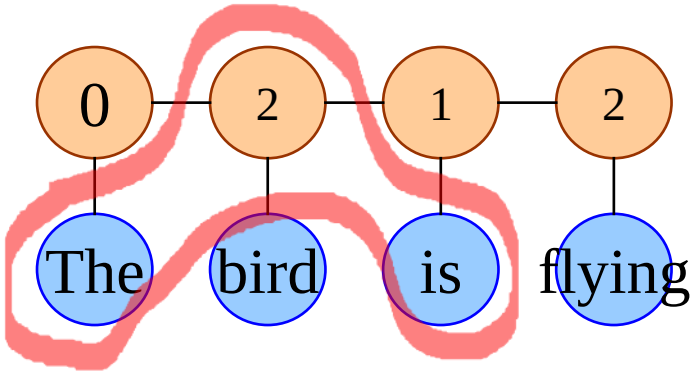


当y=2以及x=bird，特征**f(y, x)**示例如下：

Unigram: bird_2，The_2，is_2

Bigram: The_bird_2，bird_is_2

特征提取举例

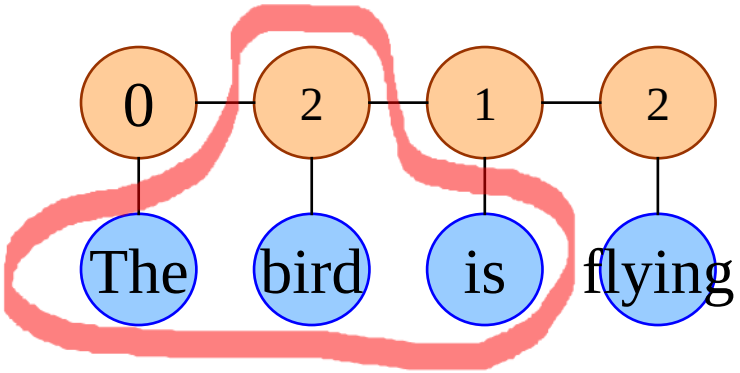


当y=2以及x=bird，特征**f(y, x)**示例如下：

Unigram: bird_2，The_2，is_2

Bigram: The_bird_2，bird_is_2，the_is_2

特征提取举例



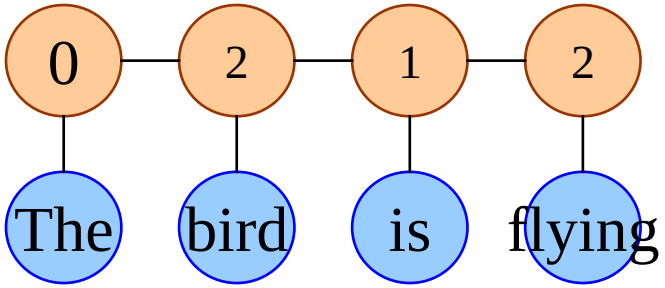
当y=2以及x=bird，特征**f(y, x)**示例如下：

Unigram: bird_2，The_2，is_2

Bigram: The_bird_2，bird_is_2，the_is_2

Trigram: The_bird_is_2

特征提取举例



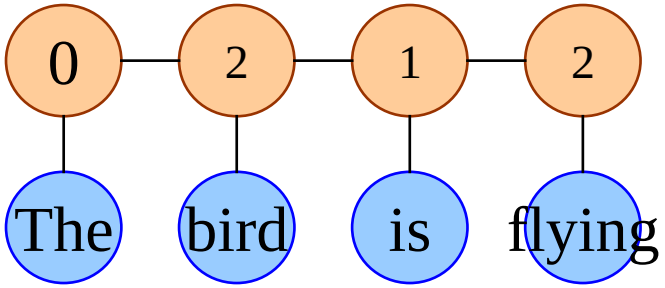
当y=2以及x=bird，特征**f(y, x)**示例如下（基于数字）：

Unigram: 2, 4, 5

Bigram: 10, 11, 15

Trigram: 19

特征提取举例



当y=2以及x=bird，特征**f(y, x)**如下（基于向量）：

<0,1,0,1,1,0,0,0,0,1,1,0,0,0,1,0,0,0,1,0,0,0,0>

特征出现记为1，否则记为0

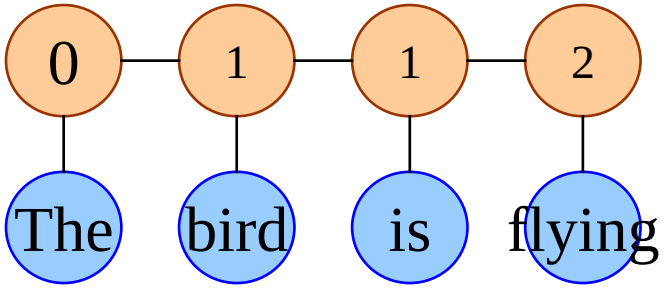
假设目前的感知器的权重向量**theta**如下：

<0,0,0,2,0,1,3,0,2,5,1,0,0,1,0,0,0,0,0,2,0,2,4>

则对应的分数**F(y|x,theta)**如下：

2+5+1=8

特征提取举例



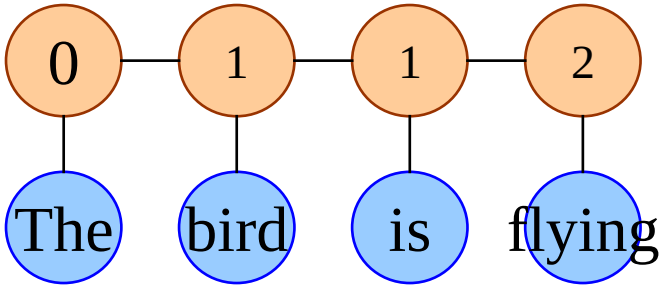
当y=1以及x=bird，特征**f(y, x)**示例如下：

Unigram: bird_1，The_1，is_1

Bigram: The_bird_1，bird_is_1，the_is_1

Trigram: The_bird_is_1

特征提取举例



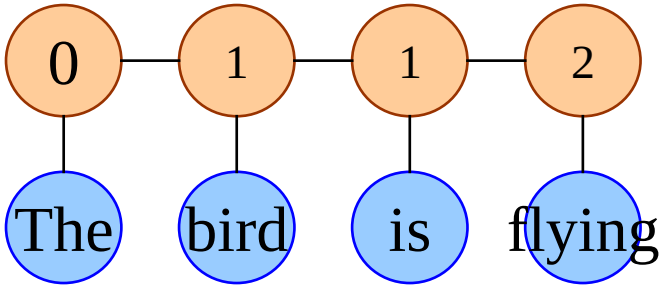
当y=1以及x=bird，特征**f(y, x)**示例如下（基于数字）：

Unigram: 1, 3, 6

Bigram: 9, 12, 16

Trigram: 23

特征提取举例



当y=1以及x=bird，特征**f(y, x)**如下（基于向量）：

<1,0,1,0,0,1,0,0,1,0,0,1,0,0,0,1,0,0,0,0,0,0,1>

特征出现记为1，否则记为0

假设目前的感知器的权重向量**theta**如下：

<0,0,0,2,0,1,3,0,2,5,1,0,0,1,0,0,0,0,0,2,0,2,4>

则对应的分数**F(y|x,theta)**如下：

1+2+4=7

感知器模型

分词任务的标签选择

- **0-1标签**：0代表不切分，1代表切分。
- **B-M-E-S标签**：B代表位于词的起始位置，M代表位于词的中间位置，E代表位于词的结束位置，S代表单个字成词。

感知器模型

分词任务过程

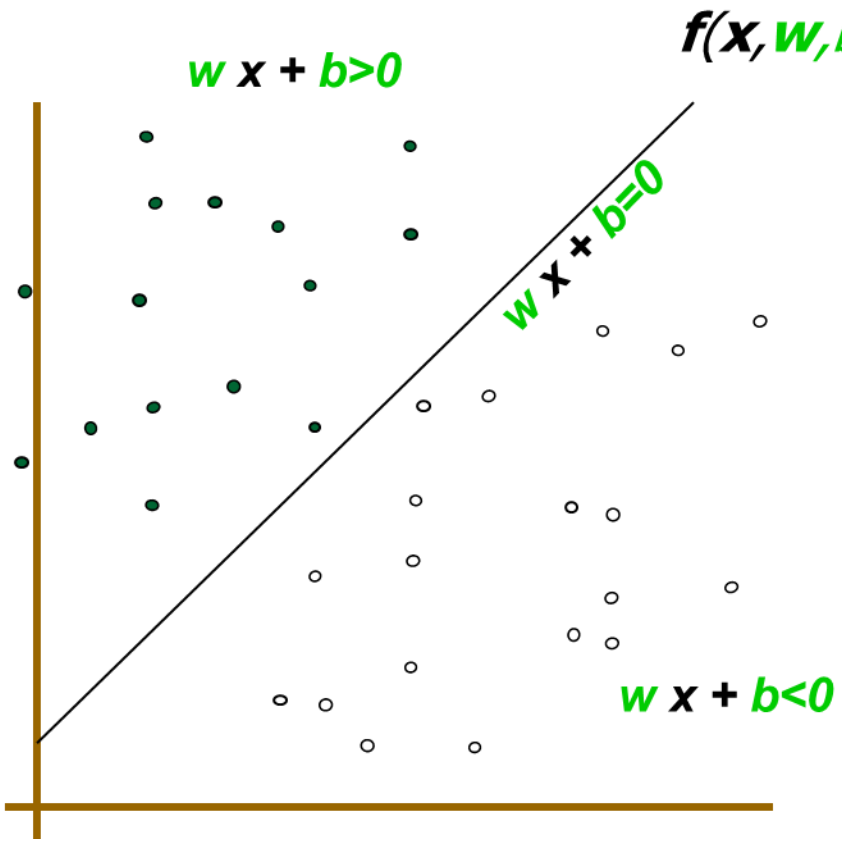
- **抽取特征**: 自定义特征模板，在训练集上抽取并保存特征。
- **生成特征向量**: 如果该特征出现记为1，否则记为0。每一个特征对应一个参数。
- **训练**: 按照标准感知器或者平均感知器的训练流程训练模型。
- **预测**: 将预测序列转换为分词结果。比如采用0-1标签的话，0对应不切分，1对应切分。得出准确率，召回率和F-score。

大纲

- **简单分类问题**
  - 感知器模型(perceptron)
  - 支持向量机模型(support vector machine, SVM)

二元分类器的新问题

- denotes +1
- denotes -1

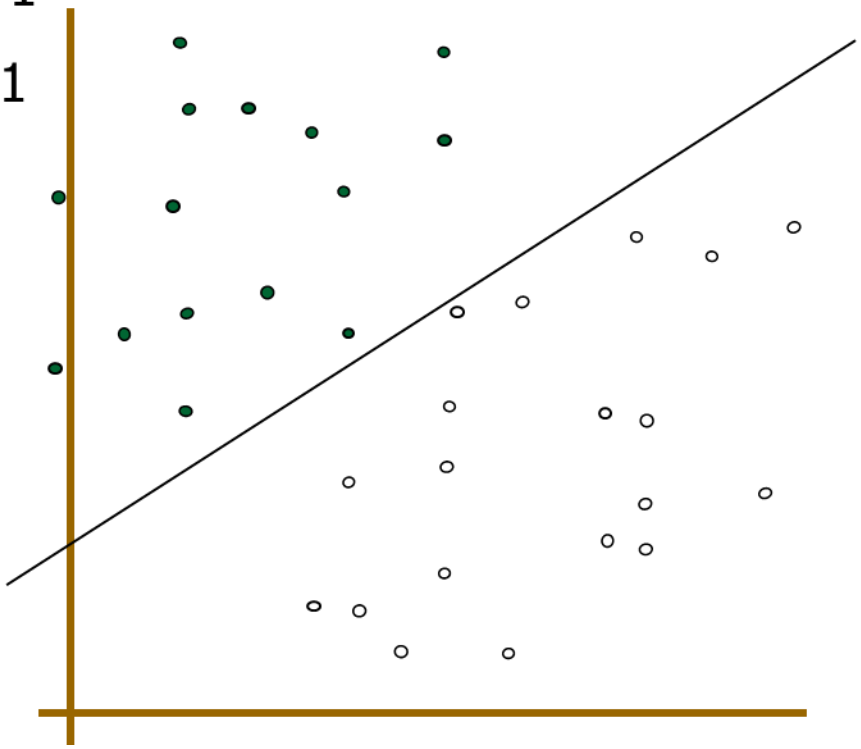


How would you classify this data?

二元分类器的新问题

- denotes +1
- denotes -1

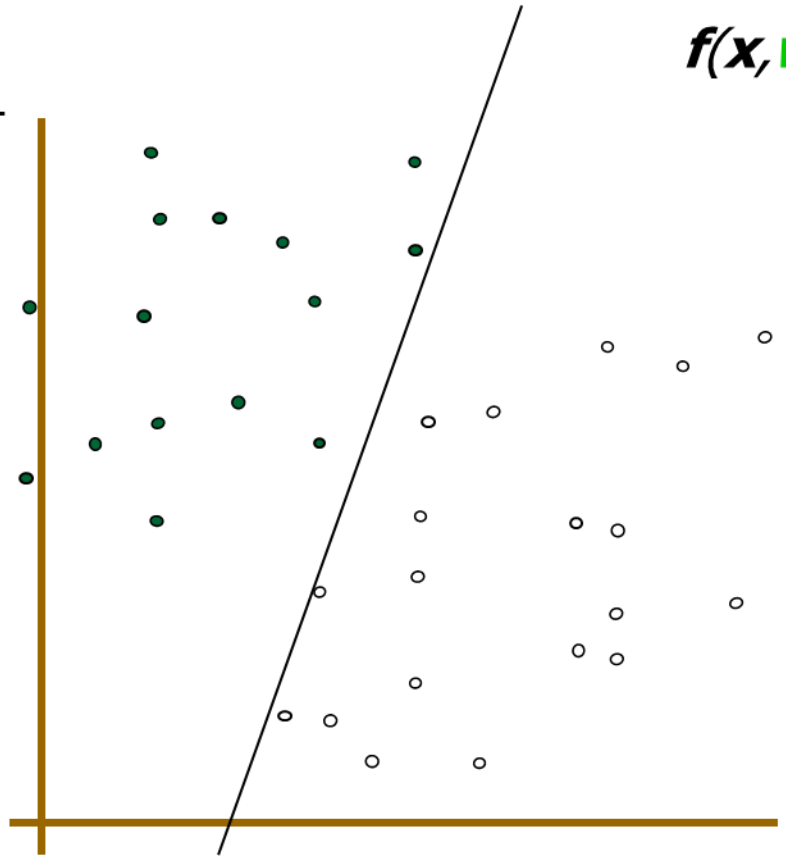
$$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}\,\mathbf{x} + b)$$



How would you classify this data?

二元分类器的新问题

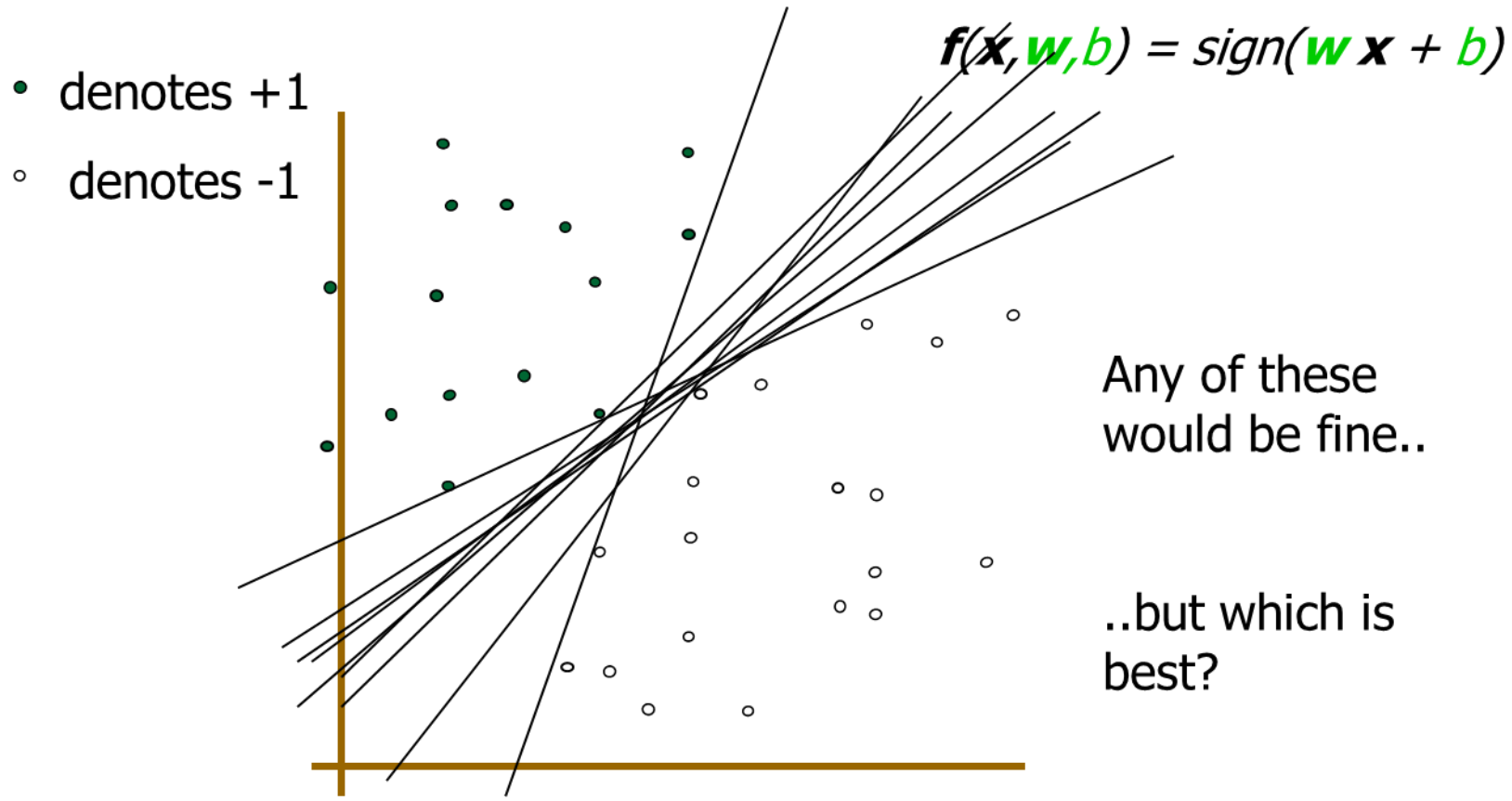
- denotes +1
- denotes -1



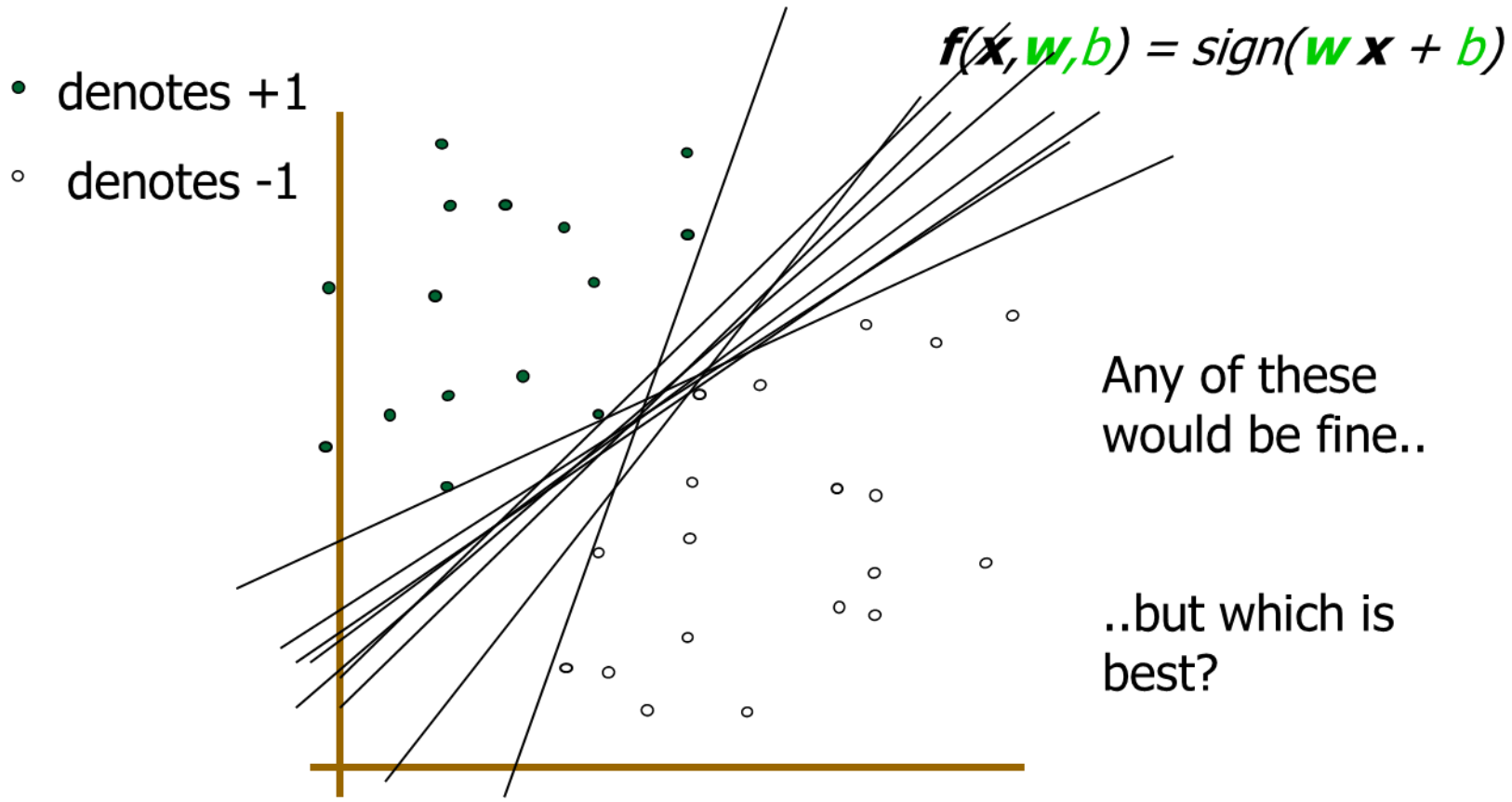
$$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}\,\mathbf{x} + b)$$

How would you classify this data?

二元分类器的新问题

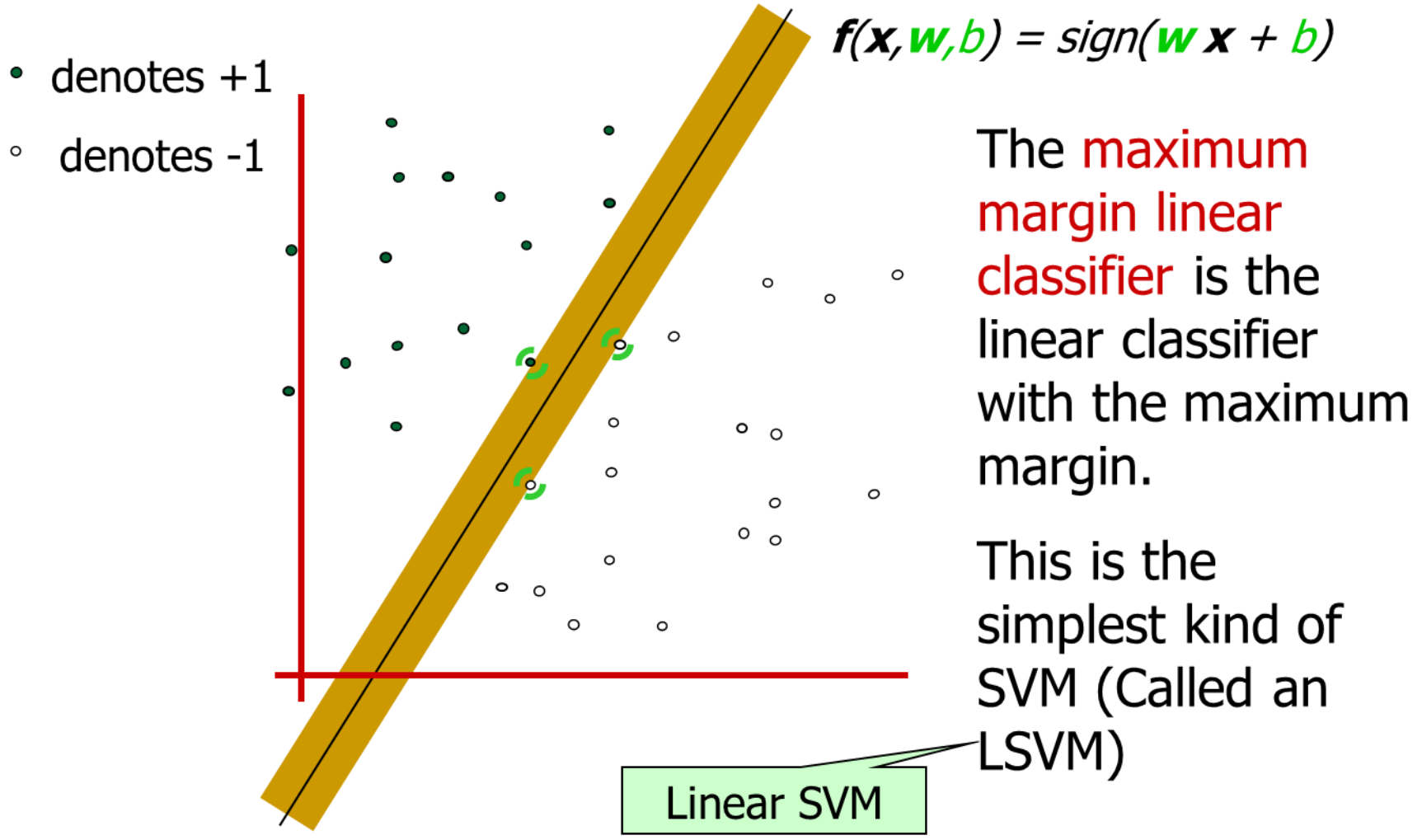

denotes +1
denotes -1
f(x,w,b) = sign(w x + b)
Any of these would be fine..
..but which is best?

二元分类器的新问题

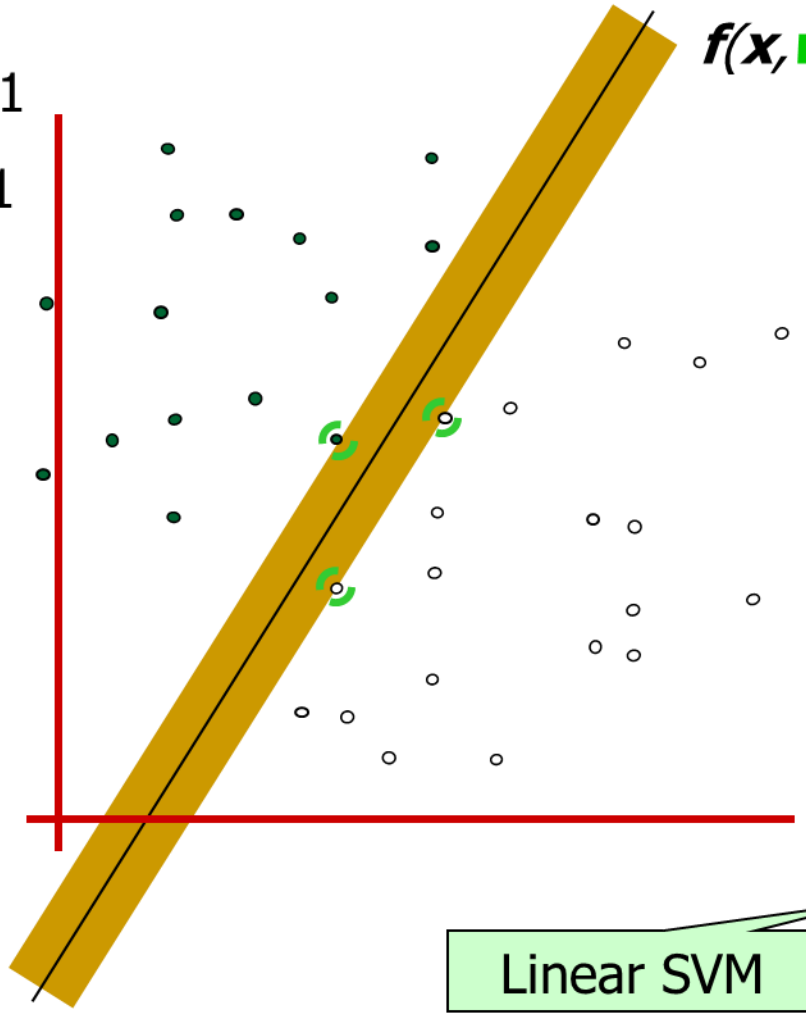


感知器属于**大边界分类器(large margin classifier)**，产生的结果可能是其中**任意一个分界线**

最大边界理论（Maximum Margin）



最大边界理论（Maximum Margin）



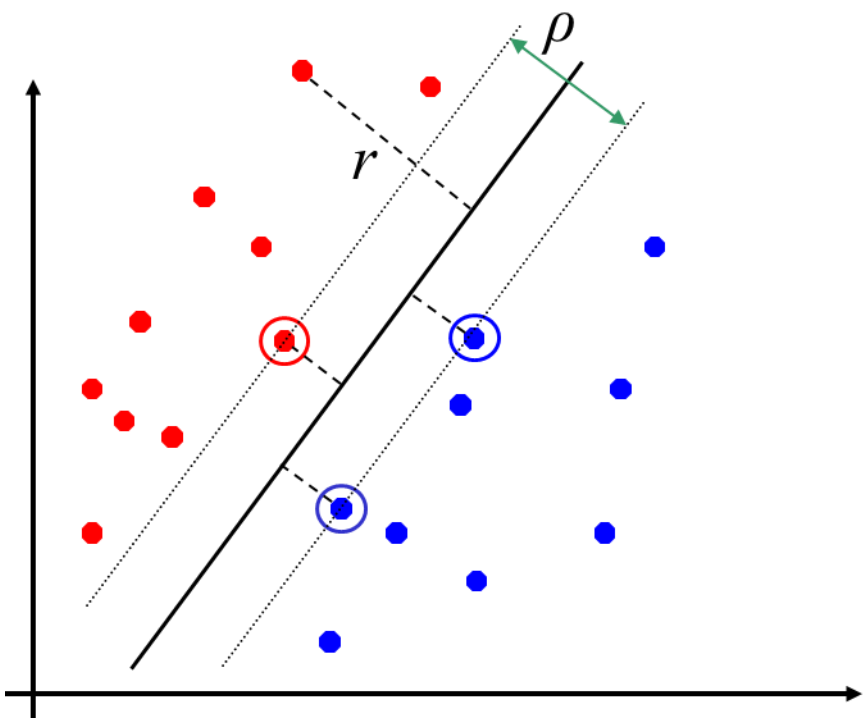
边界上的这些数据点被称为支持向量(support vectors)

这是支持向量机名字的由来

最大边界理论（Maximum Margin）

- 从数据点 $\mathbf{x}_i$ 到分界线的距离为 $r = \dfrac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$

- ***分界线的“边界(margin)”ρ*** 定义为支持向量之间的距离



最大边界理论（Maximum Margin）

- 最大边界理论：只有支持向量（support vectors)是重要的，其它的非支持向量的数据点可以忽略
- 最大化边界思想在理论和直觉上都是一个比较好的选择
- 最大边界理论在众多的实际应用上，包括很多自然语言处理任务（文本分类等）都效果很不错

大边界分类器和最大边界分类器

简单分类问题

- 感知器模型 (perceptron)

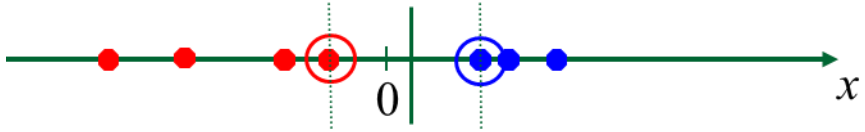
感知器属于**大边界分类器 (large margin classifier)**，学习的结果可能是其中**任意一个分界线**

- 支持向量机模型 (support vector machine, SVM)

支持向量机SVM属于**最大边界分类器(maximum margin classifier)**，学习的结果是能产生最大边界的**一个分界线**

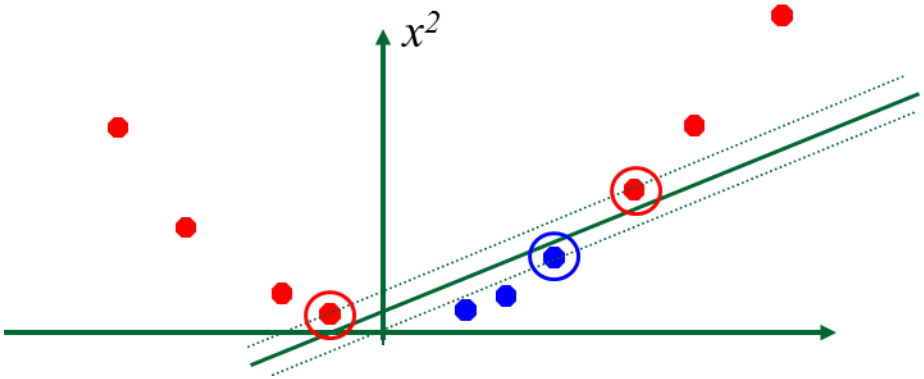
非线性支持向量机(Non-linear SVMs)

- 较简单的数据，线性分类可以很好地工作：



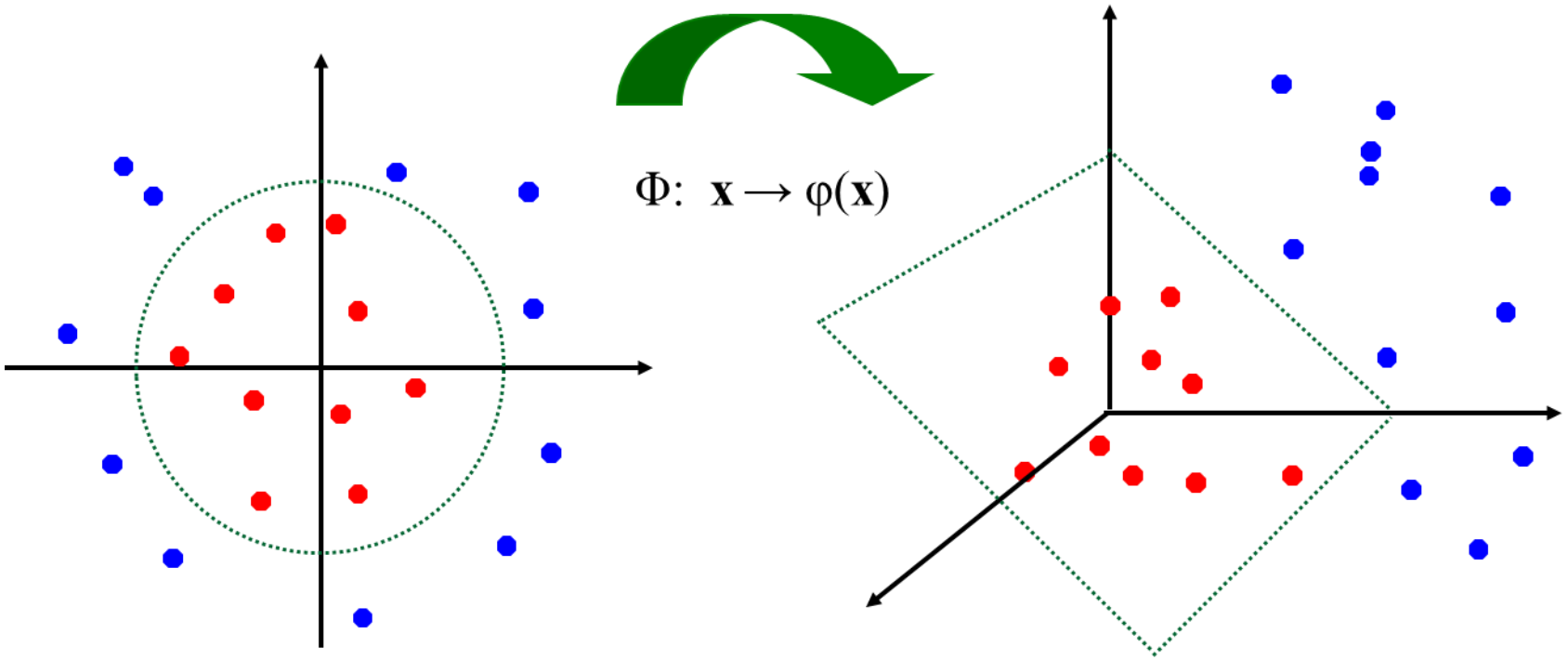
- 但是遇到这种比较困难的数据的话，应该怎么处理？

- 如果能够把低维数据投射到高维空间，则能够很好地解决这些问题！而且可以(在高维空间)继续使用线性分类！



非线性支持向量机(Non-linear SVMs)

- **主要思想：** 寻找一种通用而高效率的方法，从而能够把低维空间的数据投射到高维空间，得以在高维空间进行线性分类，解决低维空间无法线性分类的问题。



非线性支持向量机(Non-linear SVMs)

非线性SVM通过 核函数(Kernel Function)来实现

- 具体来说，线性SVM可以通过“点积”(dot product)来实现 $K(x_i,x_j)=x_i^T x_j$

- 核函数是一个特殊的函数，从而可以把点积扩展到高维的特征空间

$$K(x_i,x_j)= \varphi(x_i)^T \varphi(x_j)$$

- 核函数举例:

假设2维向量$x=[x_1 \ x_2]$，我们可以使用核函数$K(x_i,x_j)=(1 + x_i^T x_j)^2$，
从而有如下等式 $K(x_i,x_j)= \varphi(x_i)^T \varphi(x_j)$:

$$K(x_i,x_j)=(1 + x_i^T x_j)^2,$$

$$= 1+ x_{i1}^2 x_{j1}^2 + 2 x_{i1}x_{j1} x_{i2}x_{j2}+ x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1 \ x_{i1}^2 \ \sqrt{2} x_{i1}x_{i2} \ x_{i2}^2 \ \sqrt{2}x_{i1} \ \sqrt{2}x_{i2}]^T [1 \ x_{j1}^2 \ \sqrt{2} x_{j1}x_{j2} \ x_{j2}^2 \ \sqrt{2}x_{j1} \ \sqrt{2}x_{j2}]$$

$$= \varphi(x_i)^T \varphi(x_j), \quad \text{where } \varphi(x) = [1 \ x_1^2 \ \sqrt{2} x_1x_2 \ x_2^2 \ \sqrt{2}x_1 \ \sqrt{2}x_2]$$

可见，该核函数把2维空间的点积扩展到了6维空间的点积，但是几乎没有增加计算量(计算量只是很小的增加)

非线性支持向量机(Non-linear SVMs)

常用的核函数举例

- 线性核函数 Linear: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$
- 多项式核函数 Polynomial: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j)^p$
- 高斯RBF核函数 Gaussian RBF (radial-basis function network):

$$K(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}\right)$$

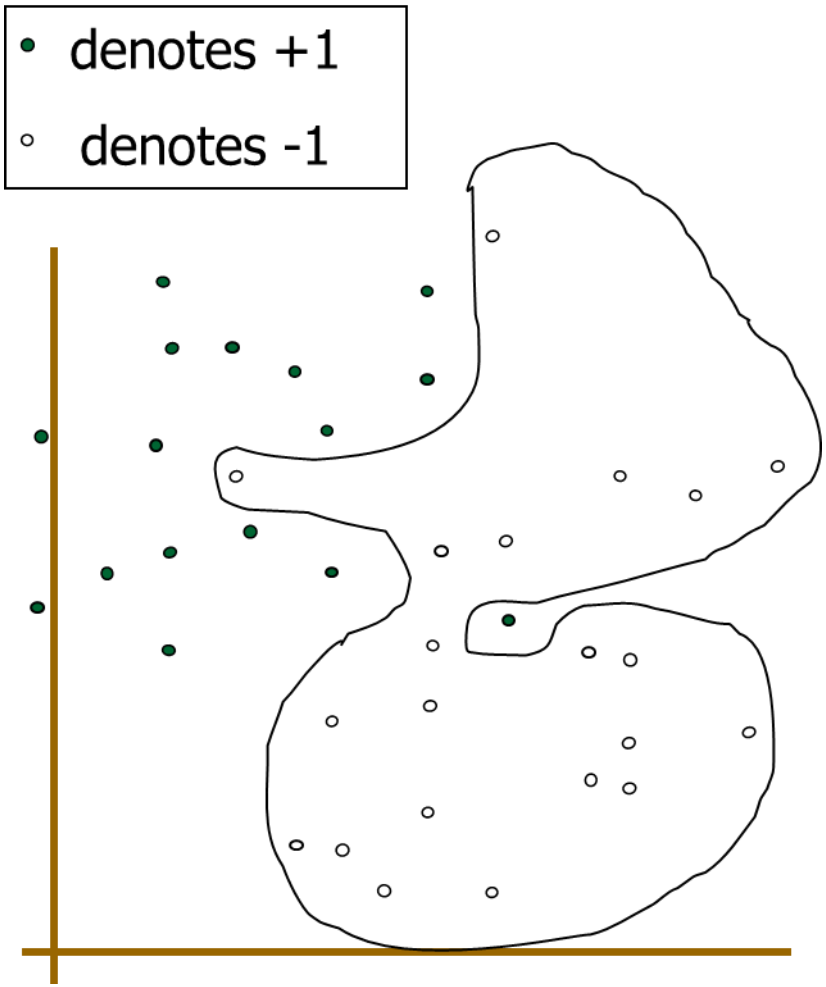
- Sigmoid核函数 Sigmoid: $K(\mathbf{x}_i,\mathbf{x}_j)= \tanh(\beta_0\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j + \beta_1)$

非线性支持向量机(Non-linear SVMs)

也就是说：

- 对于数据分布复杂的数据集，难以进行线性分类的，把低维空间数据投射到高维空间，从而在高维空间实现线性分类
- 无需显式地定义高维空间，只需要定义一个核函数就可以了，实现隐性的空间升维
- 核函数扮演了高维空间中的点积函数的功能

软边界分类(Soft Margin Classification）

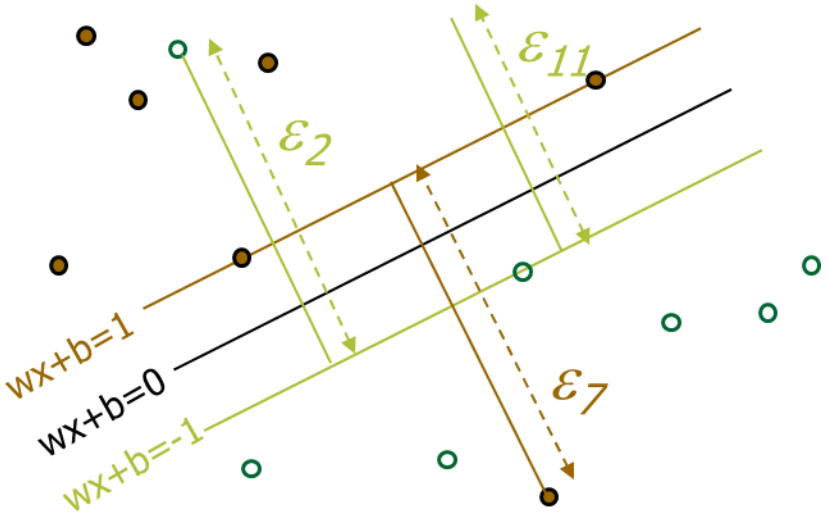


- **硬边界（Hard Margin）：严格要求所有的数据点都被正确分类**
  - 也就是说，不允许训练错误的存在
- **问题：如果训练数据噪音很大的话怎么办?**

如果直接使用核函数投射到高维空间，反而会导致严重的过拟合问题

软边界分类(Soft Margin Classification)

可以添加懈怠变量（Slack variables） ξ_i，用于允许在复杂和多噪音数据上的错误分类



软边界分类的新目标函数：

Minimize

$$\frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{k=1}^{R}\varepsilon_k$$

SVM的优点

- **在处理大数据的时候有优势**
  - 这是因为只有支持向量被用于学习分界线
- **能够处理高维的特征空间**
  - 这是因为SVM学习过程的复杂度跟特征空间的维度无关
- **能够较好地处理过拟合的问题**
  - 比如可以使用软边界方法（soft margin）
- **较好的数学特性**
  - 比如目标函数是个凸函数，可以找到全局最优点

SVM的缺点

- **核函数的选择很多，有时候不知道哪个核函数好…选择太多了也是问题☹**
  - 一般来说高斯核函数、多项式核函数的默认的选项
  - 如果默认选项效果不好，则要根据数据的特点考虑更复杂的核函数，甚至可能需要自己定义核函数

- **一般只能用于二元分类**
  - 如果多于两个类，比如文本分类可能有几十个类，怎么使用SVM进行多元分类？

一个可能的选项：

1) 对于m个输出类, 学习m个独立的SVM模型
   - SVM 1 learns “Output==1” vs “Output != 1”
   - SVM 2 learns “Output==2” vs “Output != 2”
   - :
   - SVM m learns “Output==m” vs “Output != m”

2)通过对这m个独立的SVM的输出结果和分数进行排序，可以确定最佳的输出类别

感知器和SVM的应用

- **感知器和SVM成功地应用到很多现实数据处理任务**
  - **自然语言处理**
    - 文本分类 text categorization
    - 手写识别 hand-written character recognition
  - 其它领域
    - 图像分类 image classification
    - 生物信息学 bioinformatics

感知器和SVM的应用

- **比如，在文本分类任务里面，需要对文本（text）、网页文本（hypertext）的内容进行自然语言分析，目标是把文本、网页文本自动分类到指定的一组类别集合中（比如新闻、体育、政治）**
  - 首先，需要对每一个文档提取自然语言特征（比如之前介绍的n-gram语言模型的概率信息）
  - 在此基础上可以使用感知器或者支持向量机进行分类
  - 文本分类有广泛的应用
    - 包括垃圾邮件识别（email filtering）
    - 网页搜索（web searching）
    - 文本的自动归类（sorting documents by topic），等等

参考文献、进一步阅读资料

- **感知器**
  - 经典论文：【Freund and Schapire, 1999】 Yoav Freund and Robert Schapire. Large margin classification using the perceptron algorithm. Machine Learning, 37(3):277–296, 1999.
  - 自然语言处理相关经典论文：【Collins, 2002】 Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. Proc. of EMNLP’ 02, 2002.

- **支持向量机**
  - 经典书籍：【Vapnik 1998】Statistical Learning Theory，Vladimir Vapnik, Wiley-Interscience; 1998
  - 一个相对浅显易懂的介绍: 【Burges 1998】C.J.C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):955-974, 1998.
  - 自然语言处理相关经典论文：【Joachims 1998】Text categorization with Support Vector Machines: learning with many relevant features， T. Joachims, ECML 1998

课程规划

- **课程内容安排**
  - NLP的概率统计基础(4-5周)
    - 一个小作业
    - 孙栩
  - NLP的语言学基础(6-7周)
    - 一个小作业
    - 中文系詹卫东教授
  - NLP的具体应用(4-5周)
    - 一个大作业
    - 孙栩

课程规划

- **课程内容安排**
  - NLP的概率统计基础(4-5周)
    - 一个小作业
    - 孙栩
  - NLP的语言学基础(6-7周)
    - 一个小作业
    - 中文系詹卫东教授
  - NLP的具体应用(4-5周)
    - 一个大作业
    - 孙栩

谢谢！

QUESTION？