机器学习和自然语言处理

常宝宝
北京大学计算语言学研究所
chbb@pku.edu.cn

机器学习导引

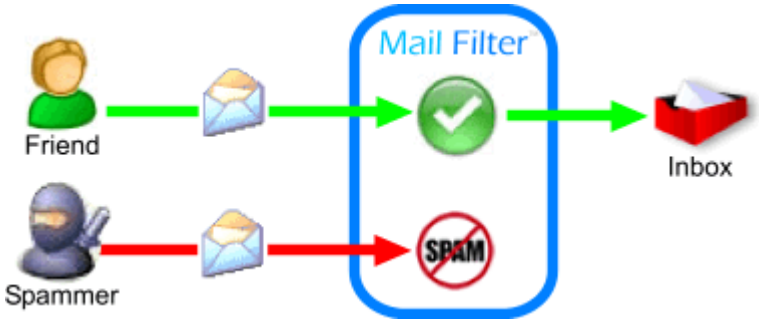
- 问题定义
  - 输入：x
  - 输出：y
- 阶乘的计算 $y = factorial(x)$
- 设计算法、编写程序

- 给出输入输出的例子集合

  {(0,1), (2,2), (5,120)…}
- 采用机器学习方法解决

任务及思路：判别垃圾邮件

- 标记好例子（数据）
- 让机器去学习
- 输出一个函数（模型）



- 把新邮件输入函数，判别是不是垃圾邮件(应用)

- 有指导的机器学习(Supervised Machine Learning)

垃圾邮件判别：标记例子

- 收集邮件，标记例子

  email-1, spam

  email-2, spam

  email-3, ham

  email-4, spam

  …

  email-100, ham
- 该例子集合称作训练集

判别垃圾邮件---特征工程

- 垃圾邮件和正常邮件在用词方面不同
  - free, loan, mortgage, Abacha, credit, Viagra, sexy
  - 不同的词在判别邮件类别方面有不同作用
  - 把词作为垃圾邮件的判别特征
- 选择$|V|$个词作为特征，邮件表示为$|V|$维向量

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_{|V|})$$

x_i代表特征i在邮件文本中出现的次数

- 设计特征、提取特征、表示邮件
- 特征工程

判别垃圾邮件---模型设计

- 学习一个函数$f(\boldsymbol{x})$
  - 若$f(\boldsymbol{x}) \geq 0$，垃圾邮件。值越大，越可信
  - 若$f(\boldsymbol{x}) < 0$，正常邮件。值越小，越可信

- 假定函数是一个线性函数
  - $f(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_{|V|} x_{|V|} + b = \boldsymbol{w}^\top \boldsymbol{x} + b$

- 给每个特征设定一个权重w_i
  - $w_i > 0$，表明该特征在垃圾邮件中出现的更多
  - $w_i < 0$，表明该特征在正常邮件中出现的更多

判别垃圾邮件---模型学习

- 模型参数 $\boldsymbol{\theta} = (\boldsymbol{w}, b)$

- 在无数个线性函数中做选择，如何选择？
  - 在训练集上，选出的函数要有良好的表现
  - 选择的原则：判别错误会造成损失

- 选择在训练集上导致损失最小的函数

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, Loss(\boldsymbol{\theta})$$

判别垃圾邮件---模型学习

- 训练集损失 $L(\hat{\boldsymbol{y}}, \boldsymbol{y})$
  - $\hat{\boldsymbol{y}}$是模型结果，$\boldsymbol{y}$是人工标注结果
  - 模型完全判别正确，损失为0，否则大于0

- 0-1损失

$$L(\hat{y}, y) = \mathbb{I}(\hat{y}, y) = \begin{cases} 0, & \hat{y} = y \\ 1, & \hat{y} \neq y \end{cases}$$

不是$\boldsymbol{\theta}$的连续函数

注：Spam: $y = +1$ $\hat{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$

ham: $y = -1$

损失函数

- 正确分类　$y \cdot f(\boldsymbol{x}) \geq 0$
- 错误分类　$y \cdot f(\boldsymbol{x}) < 0$

- 定义如下损失

$$L(\hat{y}, y; \theta) = \max\{0, -y \cdot f(\boldsymbol{x})\}$$

- 训练集上的损失，定义为平均损失

$$L(\hat{\boldsymbol{y}}, \boldsymbol{y}; \theta) = \frac{1}{n} \sum_{i=1}^{n} L(\hat{y}_i, y_i; \theta)$$

模型训练——梯度下降

- 最佳参数

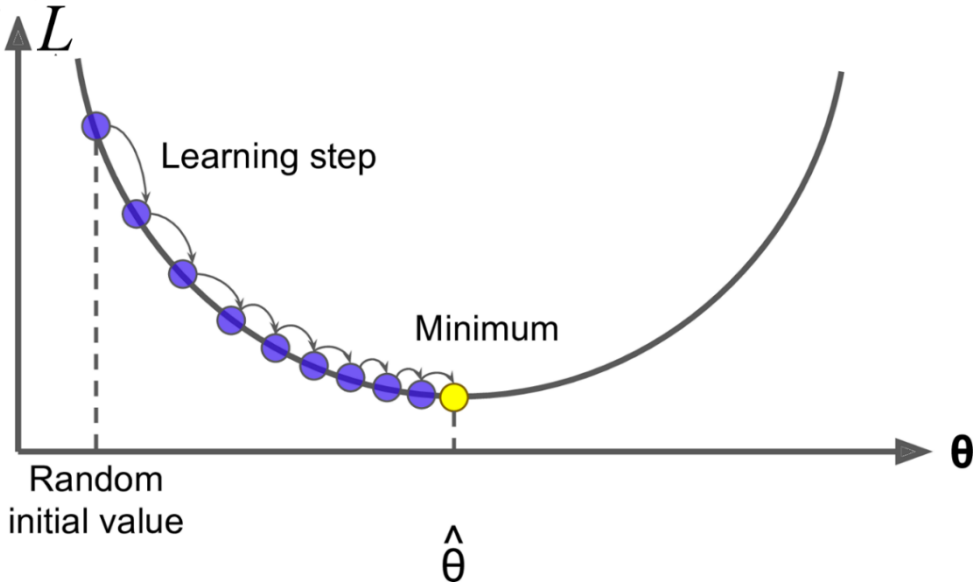
$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, L(\hat{\boldsymbol{y}}, \boldsymbol{y}; \boldsymbol{\theta})$$

寻找训练集上损失最小的模型参数

- 最优化问题

梯度下降法

(Gradient Descent)



模型训练——梯度下降

- 给定一组初始参数值θ_0
- 沿着损失下降最快的方向寻找极值点
- 损失下降最快的方向——负梯度方向

$$\boldsymbol{g} = \nabla_\theta L(\hat{\boldsymbol{y}}, \boldsymbol{y}; \boldsymbol{\theta})$$

$$\boldsymbol{g} = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta L(\hat{y}_i, y_i; \boldsymbol{\theta})$$

- 按照一定步幅(学习率)沿负梯度方向逼近极小值点

$$\theta_i \leftarrow \theta_{i-1} - \eta \cdot \boldsymbol{g}$$

机器学习总结

- 给定一个任务：把$\boldsymbol{x}$映射为y
- 给定一个训练集，进行特征设计和特征提取

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$$

- 构造一个参数化的模型

$$y = f(\boldsymbol{x}; \boldsymbol{\theta})$$

- 定义损失函数

$$L(\hat{\boldsymbol{y}}, \boldsymbol{y}; \boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} L(\hat{y}_i, y_i; \theta)$$

- 寻求可使训练集上损失最小的参数$\hat{\boldsymbol{\theta}}$，得到模型
- 利用所得到模型处理未来的输入对象

思考

基于训练集得到的模型处理未来数据

有条件吗？

独立同分布假设(i.i.d. assumption)

- 根据已标注邮件学习模型
- 模型能否用于过滤未来的邮件？
- 未来的邮件应和已标注的邮件具有相同的统计规律

- 基本假设：独立同分布假设
  - 所有数据实例源自同一个概率分布

$$(\boldsymbol{x}_i, y_i) \sim p(\boldsymbol{x}, y), i = 1,2, \ldots$$

    数据生成分布
  - 实例和实例之间是独立的

思考

模型处理未来数据

效果会如何？

模型推广能力(model generalization)

- 模型学到了训练集中没有推广价值的特有模式
- 如何衡量模型处理未来数据的性能？
- 把标注数据分成两个部分
  - 训练集(training set)：训练模型
  - 测试集(test set)：衡量模型的效果
- 若基于训练集学习模型，则

$$\mathbb{E}_{train}[ErrorRate] < \mathbb{E}_{test}[ErrorRate]$$

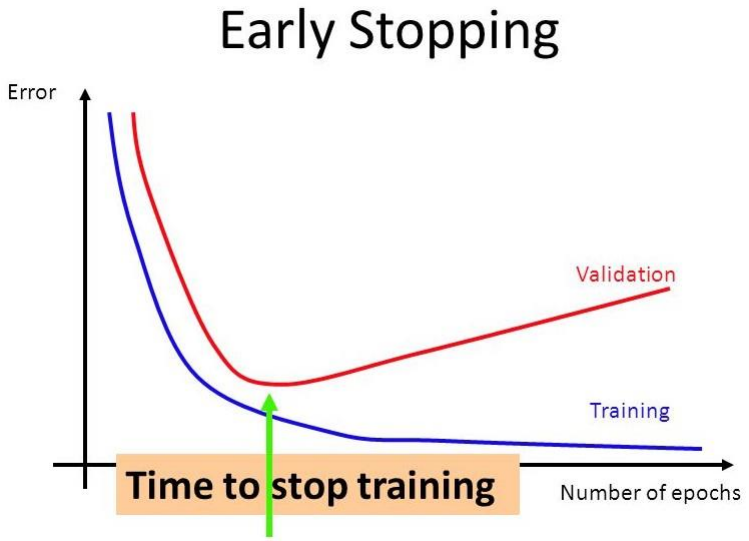
- 测试集上错误率表征模型的推广能力(generalization ability)

模型推广能力(model generalization)

- 过拟合(overfitting):低训练集错误率、高测试集错误率
- 理想状态
  - 低训练集错误率
  - 低测试集错误率 (或者 较小的训练集与测试集错误率差异)
- 不能单纯基于训练集损失最小求解模型
- 训练时，监控模型在测试集上的错误率？
- 选择测试集上错误率较小的模型？
  - 测试集代表未来数据
  - 基于测试集选择模型参数，扭曲模型推广能力评价

开发集(development set)和早停止(early stop)

- 把标注数据分成三部分
  - 训练集(training set)
  - 测试集(test set)
  - 开发集(development set)
- 开发集也称验证集(validation set)
- 训练时，监控开发集错误率



- 在开发集错误率上升时，提前终止训练，即早停止策略
  - 在模型过拟合之前终止训练过程
- 用测试集上的错误率衡量模型的推广能力
- 训练集-测试集-开发集 分配比例？(60:20:20？ 没有标准)

思考

线性模型总是好模型吗？

模型容量

模型容量(capacity)

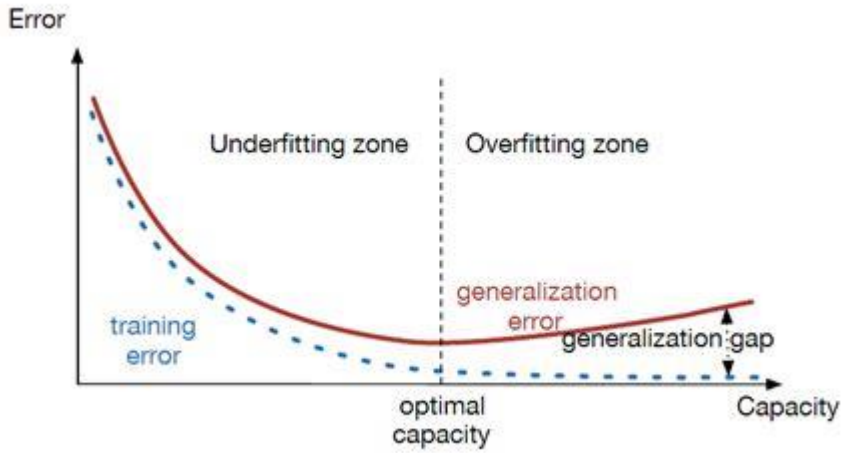
- 垃圾邮件过滤中假定模型为线性模型
  - 排除了非线性关系的可能
  - 输入和输出之间可能不是线性关系
- 模型表达能力(expressive power)不够
  - 高训练集错误率，欠拟合(underfitting)现象
- 模型容量(capacity)：模型拟合复杂关系的能力
- 非线性模型的容量高于线性模型
  - 非线性模型涵盖了线性模型
  - 可以学到非线性关系
- 我们应该追求高容量吗？

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

--John von Neumann

过拟合、欠拟合和容量

- 模型容量过小，欠拟合现象
  - 高训练集错误率、高测试集错误率
- 模型容量过大，过拟合现象
  - 更容易学到训练集的特质
  - 低训练集错误率、高测试集错误率



奥卡姆剃刀(Occam's razor)原则

- 吝啬原则(law of parsimony)
- 如无必要，勿增实体
- 如果你有两个原理，它们都能解释观测到的事实，那么你应该使用简单的那个(让事情保持简单)

- 如果有两个模型，都可以同样好地解释观察数据(训练集)，我们应该优先选择简单的模型。

思考

控制模型的复杂度

正则化

正则化(regularization)

- 模型越复杂，越容易出现过拟合现象
- 正则化——控制模型复杂度
- 求解参数，尽量让损失最小
- 求解参数，尽量让模型简单
- 优化目标

$$\widehat{\boldsymbol{\theta}} = \min_{\boldsymbol{\theta}} L(\hat{\boldsymbol{y}}, \boldsymbol{y}; \boldsymbol{\theta}) + \lambda R(\boldsymbol{\theta})$$

$L(\hat{\boldsymbol{y}}, \boldsymbol{y}; \boldsymbol{\theta})$ 训练集上的损失

$R(\boldsymbol{\theta})$ 正则项，模型复杂度(罚项)

λ正则项在优化目标中的权重

正则化(regularization)

令$\boldsymbol{\theta} = (w_1, w_2, \ldots, w_n)$代表模型参数

- L_1-正则

$$R(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1 = |w_1| + |w_2| + \cdots + |w_n|$$

- L_2-正则

$$R(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2 = w_1^2 + w_2^2 + \cdots + w_n^2$$

- $L_1 + L_2$-正则

$$R(\boldsymbol{\theta}) = \alpha\|\boldsymbol{\theta}\|_1 + (1-\alpha)\|\boldsymbol{\theta}\|_2^2$$

- 机器学习中有很多正则化技术

思考

常见的损失函数有哪些？

损失函数(Loss)

- 在训练集上学习(正则化)损失最小的模型。
- 损失函数的特点
  - 单个例子的损失$L(\hat{y}, y)$
  - 训练集上的损失$L(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n} L(\hat{y}_i, y_i)$
  - $L(\hat{y}, y) \geq 0$
  - 是参数的连续函数
- 二分类问题(回顾)

$$L(\hat{y}, y) = \max\{0, -\hat{y} \cdot y\}$$

$\hat{y}$是模型预测结果，$\hat{y} \in \mathbb{R}$　$\hat{y} \geq 0$ 正类　$\hat{y} < 0$ 负类

y是正确结果(答案)，$y \in \{+1, -1\}$

铰链损失(hinge loss)

- 二分类(binary classification) $y \in \{+1, -1\}$

$$L_{hinge}(\hat{y}, y) = \max(0, 1 - y \cdot \hat{y})$$

当$y \cdot \hat{y} > 1$时，$L_{hinge}(\hat{y}, y) = 0$

- 多分类(multi-class classfication) $y \in \{1,2, \dots, m\}$

$\hat{y}_i$模型预测$y = i$的得分

t正确类别，$k \neq t$是其他类别

$$L_{hinge}(\hat{y}, y) = \max(0, 1 - (\hat{y}_t - \hat{y}_k))$$

当$\hat{y}_t - \hat{y}_k \geq 1$时，$L_{hinge}(\hat{y}, y) = 0$

交叉熵损失(cross entropy loss)

- 交叉熵(cross entropy)

$$H(p,q) = -\sum_{x} p(x) \log q(x)$$

  - 可以衡量两个概率分布的相似程度
  - 当$p = q$时，交叉熵取最小值
- 概率型机器学习模型 $\boldsymbol{x}$所属类别的分布$p(c|\boldsymbol{x})$
- 类别为i的实例$\boldsymbol{x}$
  - 真实分布：one-hot分布 $y_i = 1$ $y_k = 0$ $(k \neq i)$，即 $(0,0,\dots,1,\dots,0)$
  - 模型预测分布：$(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_i, \dots, \hat{y}_m)$
- 预测分布与真实分布的交叉熵 $-\sum_{c} y_c \log \hat{y}_c = -\log \hat{y}_i$

交叉熵损失(cross entropy loss)

- 二分类(binary classification)

$$L_{CrossEntropy}(\hat{y}, y) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

y是$\boldsymbol{x}$为正类的概率，若$\boldsymbol{x}$是正类$y = 1$，否则$y = 0$

$\hat{y}$是模型预测$\boldsymbol{x}$为正类的概率

- 多分类(multi-class classification)

$$L_{CrossEntropy}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -\log \hat{y}_t$$

$\boldsymbol{x}$的正确类别为t,$\hat{y}_t$是模型预测$\boldsymbol{x}$类别为t的概率

- 调整参数使预测分布逼近真实分布
- 交叉熵损失只适用于概率型模型(模型输出是概率分布)
- 也称作负对数似然损失(negative log likelihood)

思考

梯度下降的效率

从梯度下降到随机梯度下降

梯度下降(gradient descent)

- 最速下降方向: 负梯度方向 $-\boldsymbol{g}$

$$\boldsymbol{g} = \nabla_\theta L(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

$$\boldsymbol{g} = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta L(\hat{y}_i, y_i)$$

- 更新步幅：学习率 η
- 梯度更新规则

$$\boldsymbol{\theta}_{i-1} \leftarrow \boldsymbol{\theta}_{i-1} - \eta \cdot \boldsymbol{g}$$

- 梯度计算针对训练集中所有例子(计算代价太大大)
- 确定性梯度下降(deterministic gradient descent)
- 整批梯度下降(full batch or batch gradient descent)

随机梯度下降(stochastic gradient descent)

$$\boldsymbol{g} = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta L(\hat{y}_i, y_i) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\hat{p}_{traing}}[\nabla_\theta L(\hat{y}, y)]$$

- 梯度是训练集中例子损失梯度的期望
- 从训练集中随机选择m个例子组成样本
- 用样本的梯度平均值近似估算训练集梯度期望
- 样本中例子的梯度均值

$$\hat{\boldsymbol{g}} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta L(\hat{y}_i, y_i)$$

- 每次更新参数无需逐个例子计算梯度，提高了效率
- 所选样本通常也称作minibatch(小批)

随机梯度下降(stochastic gradient descent)

Input: Learning rate η

Initial Parameters $\boldsymbol{\theta}$

while stopping criterion not met **do**

· Sample a minibatch of m examples from the training set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ with corresponding targets y_i

· Compute gradient estimate

$$\hat{\boldsymbol{g}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} \nabla_{\boldsymbol{\theta}} L(\hat{y}_i, y_i)$$

· Applying update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \hat{\boldsymbol{g}}$

end while

- 若样本规模$m = 1$, 称作online随机梯度下降

思考

自然语言处理中有哪些类型的机器学习问题？

关于任务

- 对自然语言处理而言，大多数任务都没有明确的算法解，需要使用机器学习方法

- 常见任务形态
  - 分类任务(classification)
  - 序列标注任务(sequence labeling)
  - 结构预测任务(structured prediction)
  - 序列转写任务(sequence transduction)

分类任务(Classification)

- 确定输入对象$\boldsymbol{x}$的类别y,$y \in \{1,2,\cdots,k\}$，y是标量
- 二分类($k = 2$)
  - 垃圾邮件过滤(Spam Filter)
    - 输入是电子邮件，输出是{spam, non-spam}
  - 情感极性分析(Sentiment Analysis)
    - 输入是文本或者句子，输出是{positive, negative}
- 多分类($k > 2$)
  - 文本主题分类(Text Classification)
    - 输入是文本，输出是{Science, Health, Education, Sports, Culture, History, Entertainment, Business, Politics}等主题分类

序列标注任务(Sequence Labeling)

- 输入对象$\boldsymbol{x}$是序列，输出也是序列$\boldsymbol{y}$，长度相等

$$\boldsymbol{x} = x_1 x_2 \cdots x_n \rightarrow \ \boldsymbol{y} = y_1 y_2 \cdots y_n$$

- 汉语分词(Chinese Word Segmentation)
  - 汉语以“字”为书写单位，识别出其中的“词”
    小李毕业于北京大学 → 小李/毕业/于/北京大学
  - 序列标注建模

小	李	毕	业	于	北	京	大	学
B	E	B	E	S	B	M	M	E

注：B-词首字 M-词中字 E-词尾字 S-单字词

序列标注任务(Sequence Labeling)

- 词类标注(Part of speech tagging; POS tagging)
  - 判别句子中词的词性(Part of speech)
  - 输入是(分词后的)句子，输出是词性序列

    小李 毕业 于 北京大学 →小李/n 毕业/v 于/p 北京大学/n
  - 序列标注建模

    小李　　毕业　　于　　北京大学

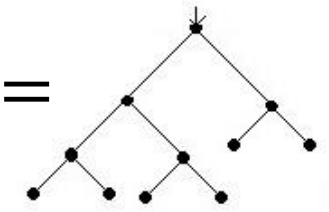
    n　　v　　p　　n

    注：n-名词　　v-动词 p-介词 a-形容词 ……
- 其他序列标注任务
  - 命名实体识别、语义角色标记……

结构预测任务(Structure Prediction)

- 输入对象$\boldsymbol{x}$常是序列，输出$\boldsymbol{y}$是某种结构

$$\boldsymbol{x} = x_1 x_2 \cdots x_n \quad \rightarrow \quad \boldsymbol{y} =$$



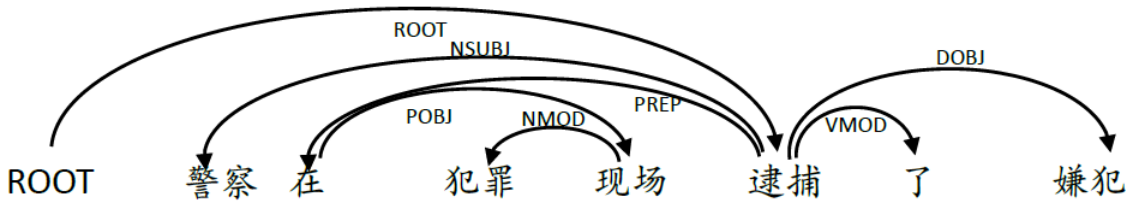
- 句法分析(Syntactic parsing)
  - 输入是句子，输出是句子的句法结构(Syntactic structure)
  - 存在不同的句法结构描写理论
    - 依存结构(Dependency structure)
    - 成分结构(Constituency structure)

结构预测任务(Structure Prediction)

- 依存句法分析(syntactic dependency parsing)
  - 输入是句子，输出是依存树(dependency tree)
  - 中心词(head)、修饰词(modifier)
  - 依存关系(dependency relation)

警察 在 犯罪 现场 逮捕 了 嫌犯

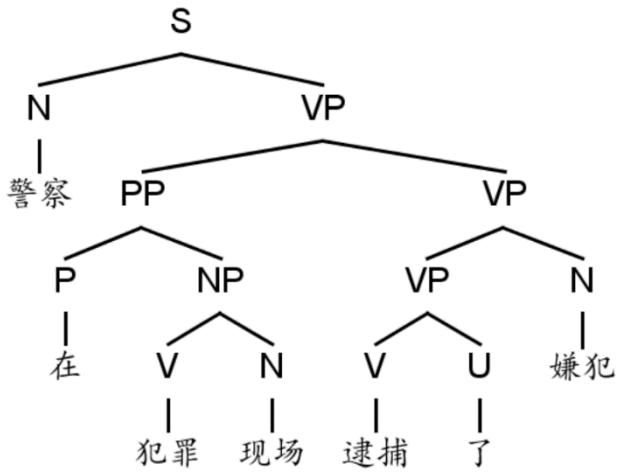
↓



结构预测任务(Structure Prediction)

- 成分句法分析(constituency parsing)
  - 输入是句子，输出是成分树(constituency tree)
  - 词、短语(phrase)、句子，成分组成关系
  - 短语类型及标签(label)

警察 在 犯罪 现场 逮捕 了 嫌犯 →



- 其他结构预测任务
  - 语义依存分析、篇章结构分析……

序列转写任务(sequence transduction)

- 输入对象$\boldsymbol{x}$是序列，输出$\boldsymbol{y}$也是序列，长度不等

$$\boldsymbol{x} = x_1 x_2 \cdots x_n \rightarrow \ \boldsymbol{y} = y_1 y_2 \cdots y_m$$

- 机器翻译(Machine Translation)
  - 把源语言翻译成目标语言
  - 输入是源语言句子，输出是目标语言句子

the spirit is willing but
the flesh is weak → 心有余而力不足

- 其他序列转写任务
  - 文本摘要(document summarization)

没有免费的午餐(No free lunch theorem)

- 独立同分布假设：所有数据源自同一分布

$$(\boldsymbol{x}_i, y_i) \sim p(\boldsymbol{x}, y), i = 1,2, \ldots$$

- 不同的数据分布$p(\boldsymbol{x}, y)$会生成不同的数据集合
- 不同机器学习方法在所有数据集上的平均错误率相同
- 通俗地说，在某些任务上，方法A效果优于方法B，那么一定存在一些另外的任务，方法B效果优于方法A
- 不追求在所有任务上都表现最好的绝对最好方法
- 针对具体任务，寻求表现优异的方法