机器学习

零、翻译

- 第 1 章
  - 机器学习 ML (machine learning)
  - 属性空间 (attribute space)
  - 分类 (classification)
  - 回归 (regression)
  - 二分类 (binary classification)
  - 多分类 (multi-class classification)
  - 聚类 (clustering)
  - 簇 (cluster)
  - 监督学习 SL (supervised learning)
  - 无监督学习 (unsupervised learning)
  - 分布 (distribution)
  - 独立同分布 i.i.d (independent and identically distributed)
  - 归纳 (induction) 和演绎 (deduction)
  - 泛化 (generalization) 和特化 (specialization)
  - 归纳学习 (induction learning)
  - 版本空间 (version space)
  - 归纳偏好 (inductive bias)
  - 特征选择 (feature)
  - 奥卡姆剃刀 (Occam's razor)
  - 没有免费午餐定理 NFL (No Free Lunch Theorem)
  - 人工智能 AI (artificial intelligence)
  - 逻辑理论家 (Logic Theorist)
  - 通用问题求解 (General Problem Solving)
  - 连接主义 (connectionism)
  - 感知机 (Perceptron)
  - 符号主义 (symbolism)
  - 归纳逻辑程序设计 ILP (Inductive Logic Programming)
  - 统计学习 (statistical learning)
  - 支持向量机 SVM (Support Vector Machine)
  - 核方法 (kernel methods)
  - 核技巧 (kernel trick)
  - 数据挖掘 (data mining)

    - SDM 模型 (Sparse Distributed Memory)
- 第 2 章
    - 错误率 (error rate)
    - 精度 (accuracy)
    - 训练误差 (training error)
    - 经验误差 (empirical error)
    - 泛化误差 (generalization error)
    - 过拟合 (overfitting)
    - 欠拟合 (underfitting)
    - 模型选择 (model selection)
    - 留出法 (hold-out)
    - 交叉验证法 (cross validation)
    - k 折交叉验证法 (k-fold cross validation)
    - 留一法 LOO (Leave-One-Out)
    - 包外估计 (out-of-bag estimate)
    - 调参 (parameter tuning)
    - 验证集 (validation set)
    - 性能度量 (performance measure)
    - 均方误差 MSE (mean squared error)
    - 查准率/准确率 (precision)
    - 查全率/召回率 (recall)
    - 混淆矩阵 (confusion matrix)
    - 平衡点 BEP (Break-Even Point)
    - 受试者工作特征 ROC (Receiver Operating Characteristic)
    - 代价矩阵 (cost matrix)
    - 代价敏感 (cost-sensitive)
    - 统计假设检验 (hypothesis test)
    - 偏差-方差分解 (bias-variance decomposition)
    - 偏差-方差窘境 (bias-variance dilemma)
- 第 3 章
    - 线性回归 LR (linear regression)
    - 多元线性回归 MLR (multivariate linear regression)
    - 正则化 (regularization)
    - 对数线性回归 (log-linear regression)
    - 替代函数 (surrogate function)
    - 对数几率函数 (logistic function)
    - 极大似然法 MLM (maximum likehood method)
    - 梯度下降法 GD (gradient descent method)
    - 牛顿法 (Newton method)
    - 线性判别分析 LDA (Linear Discriminant Analysis)

    - Fisher 判别分析 FDA (Fisher Dscriminant Analysis)
    - 类内散度矩阵 (within-class scatter matrix)
    - 类间散度矩阵 (between-class scatter matrix)
    - 广义瑞利商 (generalized Rayleigh quotient)
    - 一对一 OvO (One vs. One)
    - 一对余 OvR (One vs. Rest)
    - 多对多 MvM (Many vs Many)
    - 纠错输出码 ECOC (Error Correcting Output Codes)
    - 欠采样 (undersampling)
    - 过采样 (oversampling)
    - 阈值移动 (threshold-moving)
    - 稀疏表示 (sparse representation)
- 第 4 章
    - 决策树 DT (decision tree)
    - 分而治之 (divide-and-conquer)
    - 信息熵 (information entropy)
    - 信息增益 (information gain)
    - 迭代二分类器3 ID3 (Iterative Dichotomiser)
    - CART (Classification and Regression Tree)
    - 二分法 (bi-partition)
- 第 5 章
    - 神经网络 NN (neural network)
    - 激活函数 (activation function)
    - 挤压函数 (squashing function)
    - 哑结点 (dummy node)
    - 多层前馈神经网络 (multi-layer feedforward neural networks)
    - 误差逆传播法/反向传播算法 BP (error BackPropagation)
    - 梯度下降 (gradient descent)
    - 随机梯度下降 SGD (stochastic gradient descent)
    - 试错法 (trial-by-error)
    - 累积误差逆传播算法 (accumulated error BackPropagation)
    - 模拟退火 (simulated annealing)
    - 遗传算法 (genetic algorithms)
    - 随机梯度下降 SGD (stochastic gradient descent)
    - 径向基函数网络 RBF (Radial Basis Function)
    - 自适应谐振网络 ART (Adaptive Resonance Theory)
    - 自组织映射网络 SOM (Self-Organizing Map)
    - 级联相关网络 CC (Cascade-Correlation)
    - 递归神经网络 RNN (recurrent neural networks)
    - 深度学习 DL (deep learning)

    - 深度信念网络 DBN (deep belief network)
    - 卷积神经网络 CNN (Convolutional Neural Network)
    - 修正线性单元 ReLU (Rectified Linear Unit)
    - 特征工程 (feature engineering)
- 第 6 章
    - 支撑向量机 SVM (Support Vector Machine)
    - SMO (Sequential Minimal Optimization)
    - 支持向量回归 SVR (Support vector Regression)
    - 主成分分析 PCA (Principal Component Analysis)
- 第 7 章
    - 贝叶斯最优分类器 (Bayes optimal classifier)
    - 判别式模型 (discriminative models)
    - 生成式模型 (generative models)
    - 极大似然估计 MLE (Maximum Likehood Estimation)
    - 朴素贝叶斯分类器 (naive Bayes classifier)
    - 拉普拉斯修正 (Laplacian correction)
    - 半朴素贝叶斯分类器 (semi-naive Bayes classifier)
    - 独依赖估计 ODE (One-Dependent Estimator)
    - 超父 ODE SPODE (Super-Parent ODE)
    - TAN (Tree Augmented naive Bayes)
    - AODE (Averaged One-Dependent Estimator)
    - 最小描述长度 MDL (Minimal Description Length)
    - EM 算法 (Expectation-Maximization)
- 第 8 章
    - 集成学习 (ensemble learning)
    - 随机森林 RF (Random Forest)
    - Bagging (Bootstrap aggregating)
    - 多响应线性回归 MLR (Multi-response Linear Regression)
    - 贝叶斯模型平均 BMA (Bayes Model Averaging)
- 第 9 章
    - VDM (Value Difference Metric)
    - 学习向量量化 LVQ (Learning Vector Quantization)
    - 高斯混合聚类 GMM (Gausian Mixture Clustering)
    - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

- 翻译
    - 机器学习 ML (machine learning)
    - 监督学习 SL (supervised learning)
    - 没有免费午餐定理 NFL (No Free Lunch Theorem)

- 留一法 LOO (Leave-One-Out)
- 均方误差 MSE (mean squared error)
- 平衡点 BEP (Break-Even Point)
- 受试者工作特征 ROC (Receiver Operating Characteristic)
- 线性回归 LR (linear regression)
- 极大似然法 MLM (maximum likehood method)
- 梯度下降法 GD (gradient descent method)
- 线性判别分析 LDA (Linear Discriminant Analysis)
- Fisher 判别分析 FDA (Fisher Dscriminant Analysis)
- 一对一 OvO (One vs. One)
- 一对余 OvR (One vs. Rest)
- 多对多 MvM (Many vs Many)
- 纠错输出码 ECOC (Error Correcting Output Codes)
- CART (Classification and Regression Tree)
- 随机梯度下降 SGD (stochastic gradient descent)
- 径向基函数网络 RBF (Radial Basis Function)
- 自适应谐振网络 ART (Adaptive Resonance Theory)
- 自组织映射网络 SOM (Self-Organizing Map)
- 级联相关网络 CC (Cascade-Correlation)
- 递归神经网络 RNN (recurrent neural networks)
- 深度学习 DL (deep learning)
- 深度信念网络 DBN (deep belief network)
- 卷积神经网络 CNN (Convolutional Neural Network)
- 修正线性单元 ReLU (Rectified Linear Unit)
- 支撑向量机 SVM (Support Vector Machine)
- SMO (Sequential Minimal Optimization)
- 支持向量回归 SVR (Support vector Regression)
- 主成分分析 PCA (Principal Component Analysis)
- 超父 ODE SPODE (Super-Parent ODE)
- TAN (Tree Augmented naive Bayes)
- AODE (Averaged One-Dependent Estimator)
- 最小描述长度 MDL (Minimal Description Length)
- EM 算法 (Expectation-Maximization)
- 随机森林 RF (Random Forest)
- Bagging (Bootstrap aggregating)
- 多响应线性回归 MLR (Multi-response Linear Regression)
- 贝叶斯模型平均 BMA (Bayes Model Averaging)
- VDM (Value Difference Metric)
- 学习向量量化 LVQ (Learning Vector Quantization)
- 高斯混合聚类 GMM (Gausian Mixture Clustering)

    - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

一、绪论

- 存在一个与训练集一致的 "假设集合", 我们称之为 "版本空间".
- 机器学习算法在学习过程中对某种类型假设的偏好, 称为 "归纳偏好". 亦称 "特征选择".
- 奥卡姆剃刀: 若有多个假设与观察一致, 则选择最简单的那个.
- NFL 定理: 脱离具体问题, 空谈 "什么学习算法更好" 毫无意义, 因为若考虑所有潜在问题, 则所有学习算法一样好.

二、模型评估与选择

2.2 评估方法

- 留出法 (hold-out): 直接将数据集 D 划分为两个互斥的集合, 其中一个集合作为训练集 S, 另一个作为测试集 T.
    - 要尽可能保持数据分布一致性, 例如使用分层采样的方式.
    - 因为样本划分不同可能引入差别, 单次留出法估计结果往往不够稳定可靠, 一般要采用若干次随机划分, 重复进行实验评估后取平均值作为评估结果.
    - 训练样本和测试样本的比例也很重要, 测试集过小时, 评估结果的方差较大; 训练集过小时, 评估结果的偏差较大.
- 交叉验证法 (cross validation): 先将数据集 D 划分为 k 个大小相似的互斥子集, 每个子集 D_i 都尽可能保持数据分布一致性, 即从 D 中通过分层采样得到. 因此经常也叫 k 折交叉验证.
    - 为了减少因样本划分不同而引入的差别, k 折交叉验证通常要随机使用不同的划分重复 p 次.
    - 留一法: 令 $k = m$ 则得到了 k 折交叉验证法的一个特例, 留一法. 留一法中被实际评估的模型与期望评估的用 D 训练出的模型很相似.
- 自助法 (bootstrapping): 有放回地重复采样出 m 个样本, 得到新的数据集 D', m 取极限可得不被采样到的概率为 0.368.
    - 可以减少因为样本训练规模不同而导致的估计偏差.
    - 自助法没被采样到的样本用于测试, 这样的测试结果称为包外估计.
- 在模型选择完成后, 学习算法和参数配置都已选定, 此时应该用数据集 D 重新训练模型, 充分利用所有样本.

2.3 性能度量

回归任务常用均方误差:

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}(f(\boldsymbol{x}_i)-y_i)^2$$

或

$$E(f;\mathcal{D})=\int_{\boldsymbol{x}\sim\mathcal{D}}(f(\boldsymbol{x})-y)^2p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

分类任务常用错误率和精度:

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\boldsymbol{x}_i\neq y_i))$$

此外常用的还有查准率和查全率.

$$P=\frac{TP}{TP+FP}$$

$$R=\frac{TP}{TP+FN}$$

查准率和查全率是一对相互矛盾的度量.

如果我们能够根据学习器的预测结果对样例进行排序, 以此顺序逐个把样本作为正例进行预测, 则每次可以计算出当前的查全率和查准率, 进行作图, 然后就得到了 P-R 曲线.

2.4 假设检验

统计假设检验 (hypothesis test) 为我们进行学习器性能比较提供了重要依据. 基于假设检验结果可以推断出, 若在测试集上学习器 A 比 B 好, 那么 A 的泛化性能在统计意义上优于 B 的把握有多大, 也就是概率有多大.

对于回归任务, 泛化误差可以分解为

$$\begin{aligned}E(f;D)&=\mathbb{E}_D[(f(\boldsymbol{x};D)-y_D)^2]\\&=\mathbb{E}_D[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2]+(\bar{f}(\boldsymbol{x})-y)^2+\mathbb{E}_D[(y_D-y)^2]\\&=\mathrm{bias}^2(\boldsymbol{x})+\mathrm{var}(\boldsymbol{x})+\epsilon^2\end{aligned}$$

也就是偏差, 方差和噪声之和. 偏差刻画了学习算法本身的拟合能力; 方差度量了同样大小的训练集的变动所导致的学习性能的变化, 即数据扰动所造成的影响; 噪声表达了当前任务任何学习算法期望泛化误差的下界, 是问题本身的难度.

偏差和方差是有冲突的, 称为偏差-方差窘境.

三、线性模型

对数线性回归: $\ln y-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$.

更一般地, 对于单调可微函数 $g(\cdot)$: $y=g^{-1}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)$

这样的模型为广义线性模型, 其中 $g(\cdot)$ 为联系函数 (link function).

对数几率回归是分类模型, 本质是: $\ln \dfrac{y}{1-y} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$ 即 $y = \frac{1}{1+e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}}$.

我们将其改写为

$$\ln \frac{p(y=1)\boldsymbol{x}|}{p(y=0|\boldsymbol{x})} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

然后使用极大似然估计来优化.

线性判别分析 LDA:

类内散度矩阵 $S_w = \Sigma_0 + \Sigma_1$ 以及 $w^{\mathrm{T}} S_w w$ 尽可能小.

类间散度矩阵 $S_b = (\mu_0 - \mu_1)^{\mathrm{T}}(\mu_0 - \mu_1)$ 以及 $w^{\mathrm{T}} S_b w$ 尽可能大.

则有最大化广义瑞利商 $J = \dfrac{w^{\mathrm{T}} S_b w}{w^{\mathrm{T}} S_w w}$.

多分类学习:

OvO 每次将 N 个类别两两配对, 产生 $N(N-1)/2$ 个二分类任务, 最后结果通过投票产生.

OvR 每次将一个类作为正例, 其余作为反例, 最后选择置信度最大的类别作为分类结果.

类别不平衡:

欠采样去除一些反例, 例如集成学习的 EasyEnsemble.

过采样增加一些正例, 例如 SMOTE, 通过插值产生额外的正例.

四、决策树

信息熵: $\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$

信息熵越小, 纯度越高.

信息增益: $\mathrm{Gain}(D, a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \dfrac{|D^v|}{|D|} \mathrm{Ent}(D^v)$

信息增益越大, 纯度提升越大.

增益率: $\mathrm{Gain\_ratio}(D, a) = \dfrac{\mathrm{Gain}(D, a)}{\mathrm{IV}(a)}$

其中 $\mathrm{IV}(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}$ 称为属性 a 的固有值.

C4.5 先从候选划分属性中找出信息增益高于平均水平的属性, 再从中选择增益率高的.

CART 决策树使用基尼指数.

基尼值: $\mathrm{Gini}(D)=\sum_{k=1}^{|\mathcal{Y}|}\sum_{k'\neq k}p_kp_{k'}=1-\sum_{k=1}^{|\mathcal{Y}|}p_k^2$

基尼指数: $\mathrm{Gini\_index}(D,a)=\sum_{v=1}^{V}\frac{|D^v|}{|D|}\mathrm{Gini}(D^v)$

我们选择基尼指数小的属性作为最优划分属性.

连续值处理:

包含 $n-1$ 个元素的候选划分点集合 $T_a=\{\frac{a^i+a^{i+1}}{2}|1\leq i\leq n-1\}$

五、神经网络

- 标准 BP 算法
  - 每次针对单个训练样例更新权值与阈值
  - 参数更新频繁, 不同样例可能抵消, 需要多次迭代
- 累积 BP 算法
  - 其优化目标是最小化整个训练集上的累计误差
  - 读取整个训练集一遍才对参数进行更新, 参数更新频率较低
- 累计误差下降到一定程度之后, 进一步下降会非常缓慢, 这时使用标准 BP 算法往往会获得较好的解
- 读取训练集一遍称为进行了一轮 (one round / one epoch) 学习.

六、支持向量机

样本空间任意点 $\boldsymbol{x}$ 到超平面 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=0$ 的距离为 $r=\frac{|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b|}{\|\boldsymbol{w}\|}$

距离超平面最近的几个样本使得 $y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_b)\geq 1$ 等号成立, 称为支持向量.

两个异类支持向量到超平面的距离 $\gamma=\frac{2}{\|\boldsymbol{w}\|}$ 被称为间距 (margin).

最大化间距等价于

$$\begin{aligned}\min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\| \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b) \geq 1, i=1,2,\cdots,m\end{aligned}$$

可以构造出拉格朗日函数

$$L(\boldsymbol{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{m}\alpha_i(1-y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b))$$

其中 $\alpha_i \geq 0$.

求偏导等于零之后最后可以将问题化为

$$\begin{aligned}\max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j \\ \text{s.t.} \quad & \sum_{i=1}^{m}\alpha_i y_i = 0 \\ & \alpha_i \geq 0, i=1,2,\cdots,m\end{aligned}$$

满足 KKT 条件中的 $\alpha_i(y_i f(\boldsymbol{x}_i)-1)=0$.

我们可以使用 SMO 算法来高效优化出 α.

1. 选取一对需要更新的变量 α_i 和 α_j;
2. 固定 α_i 和 α_j 以外的参数, 求解该优化问题即可.

重复这两个步骤直至收敛即可.

最终模型: $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b = \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}+b$

核方法: $f(\boldsymbol{x}) = \sum_{i=1}^{m}\alpha_i y_i \kappa(\boldsymbol{x}_i,\boldsymbol{x})+b$

SVR: $f(\boldsymbol{x}) = \sum_{i=1}^{m}(\hat{\alpha}_i-\alpha_i)\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}+b$

重点在于支持向量带来的稀疏性.

PCA:

主成分分析需要有下列性质

- 最近重构性: 样本点到这个超平面的距离都足够近;
- 最大可分性: 样本点在这个超平面上的投影能尽可能分开.

从最大可分性出发, 样本点在超平面上的投影 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i$ 的方差应该最大化, 因此

$$\bar{\quad} \qquad \bar{\quad}$$

$$\begin{aligned}\max \quad & \operatorname{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}) \\ \text{s.t.} \quad & \boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}=\boldsymbol{I}\end{aligned}$$

因此有 $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}=\lambda\boldsymbol{W}$.

七、贝叶斯分类器

条件风险: $R(c_i|\boldsymbol{x})=\sum_{j=1}^{N}\lambda_{ij}P(c_j|\boldsymbol{x})$

总体风险: $R(h)=\mathbb{E}_{\boldsymbol{x}}[R(h(\boldsymbol{x})|\boldsymbol{x})]$

贝叶斯最优分类器: $h^*(\boldsymbol{x})=\arg\min_{c\in\mathcal{Y}}R(c|\boldsymbol{x})$

贝叶斯风险: $R(h^*)$

生成式模型: $P(c|\boldsymbol{x})=\dfrac{P(c)P(\boldsymbol{x}|c)}{P(\boldsymbol{x})}$

朴素贝叶斯分类器: $P(c|\boldsymbol{x})=\dfrac{P(c)P(\boldsymbol{x}|c)}{P(\boldsymbol{x})}=\dfrac{P(c)}{P(\boldsymbol{x})}\prod_{i=1}^{d}P(x_i|c)$

EM 算法:

- E 步 (Expectation): 基于参数 Θ 推断隐变量 Z.
- M 步 (Maximization): 基于隐变量 Z 推断参数 Θ.

八、集成学习

AdaBoost:

用加性模型

$$H(\boldsymbol{x})=\sum_{t=1}^{T}\alpha_t h_t(\boldsymbol{x})$$

来最小化指数损失函数

$$\ell_{\exp}(H|\mathcal{D})=\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H(\boldsymbol{x})}]$$