



周志华《机器学习》西瓜书 手推笔记 (v2.8)

这个公式看起来很熟，却怎么也搞不懂怎么办？

作者：王博（Kings）、Sophia
博士微信：**Kingsplus** （添加时请备注 学校/单位+专业）
Github: <https://github.com/Sophia-11/Machine-Learning-Notes>
（荣登趋势榜）
公众号【计算机视觉联盟】持续更新
后台回复【**西瓜书手推笔记**】可下载 pdf 打印版本



已完结待更笔记：《深度学习-花书手推笔记》、《无人驾驶手推笔记》、《SLAM 十四讲》
请继续关注公众号【计算机视觉联盟】最新消息或 Github

Update log

- 2019/09/20 * - 更新第一章, 第二章, 第三章
- 2019/09/24 * - 更新第四章
- 2019/09/27 * - 更新第五章
- 2019/10/08 * - 更新第六章
- 2019/11/04 * - 更新第七章
- 2019/11/30 * - 更新第八章
- 2019/12/12 * - 更新第九章
- 2019/12/28 * - 更新第十章
- 2019/01/14 * - 更新第十一章

Table of Contents

- [第一章绪论](#)
- [第二章模型评估与选择](#)
- [第三章线性模型](#)
- [第四章决策树](#)
- [第五章神经网络](#)
- [第六章支持向量机](#)
- [第七章贝叶斯分类器](#)
- [第八章集成信息](#)
- [第九章聚类](#)
- [第十章降维与度量学习](#)
- [第十一章特征选择与稀疏学习](#)
- [第十二章计算学习理论](#)
- [第十三章半监督学习](#)
- [第十四章概率图模型](#)
- [第十五章规则学习](#)
- [第十六章强化学习](#)

今天，在技术科学的主动分支学科领域中，无论是多媒体、图形学，还是网络通信、软件工程，乃至体系结构、芯片设计都能找到机器学习（Machine-Learning）技术的身影，尤其是在计算机视觉、自然语言处理等“计算机应用技术”领域，机器学习（Machine-Learning）已成为最重要的技术进步源泉之一。

“计算”的目的往往是数据分析，而数据科学的核心也恰是通过分析数据来获得价值。

若要列出目前计算机科学技术中最活跃、最受瞩目的研究分支，那么机器学习（Machine-Learning）必居其中。

美国国家科学基金会在加州大学伯克利分校启动加强计划，强调要深入研究和整合大数据时代的三个关键技术：机器学习（Machine-Learning）、云计算、众包。

机器学习（Machine-Learning）是人工智能（AI）研究发展到一定阶段的必然产物。

决策树学习技术由于简单易用，到今天仍是最常用的机器学习（Machine-Learning）技术之一。事实上，BP算法一直被应用得最广泛的机器学习（Machine-Learning）算法之一。

连接主义学习的最大局限性是其“试错性”，简单地说，其学习过程涉及大量参数，而参数的设置缺乏理论指导，主要靠手工“调参”，夸张一点说，参数调节上失之毫厘，学习结果可能谬以千里。

以往机器学习（Machine-Learning）技术在应用中取得好性能，对使用者的要求较高；而深度学习技术涉及的模型复杂度非常高，以至于只要下工夫“调参”，把参数调节好，性能往往就好。因此，深度学习虽然缺乏严格的理论基础，但它显著降低了机器学习（Machine-Learning）应用者的门槛，为机器学习（Machine-Learning）技术走向工程实践带来了便利。

深度学习如今火起来的基本原因有两个：数据大了、计算能力强了。深度学习模型拥有大量参数，若数据样本少，则很容易“过拟合”；如此复杂的模型、如此大的数据样本，若缺乏强力计算设备，根本无法求解。

机器学习（Machine-Learning）算法在学习过程中对某种类型假设的偏好，称为“归纳偏好”，或简称为“偏好”。任何一个有效的机器学习（Machine-Learning）算法必有其归纳偏好。

“奥卡姆剃刀”是一种常用的、自然科学研究中最基本的原则，即“若多个假设与观察一致，则选最简单的那个”。

在具体问题现实问题中，算法的归纳偏好是否与问题本身匹配，大多数时候直接决定了算法能否取得好的性能。

1.6 应用现状

机器学习（Machine-Learning）是人工智能（AI）研究发展到一定阶段的必然产物。

决策树学习技术由于简单易用，到今天仍是最常用的机器学习（Machine-Learning）技术之一。事实上，BP算法一直被应用得最广泛的机器学习（Machine-Learning）算法之一。

连接主义学习的最大局限性是其“试错性”，简单地说，其学习过程涉及大量参数，而参数的设置缺乏理论指导，主要靠手工“调参”，夸张一点说，参数调节上失之毫厘，学习结果可能谬以千里。

以往机器学习（Machine-Learning）技术在应用中取得好性能，对使用者的要求较高；而深度学习技术涉及的模型复杂度非常高，以至于只要下工夫“调参”，把参数调节好，性能往往就好。因此，深度学习虽然缺乏严格的理论基础，但它显著降低了机器学习（Machine-Learning）应用者的门槛，为机器学习（Machine-Learning）技术走向工程实践带来了便利。

深度学习如今火起来的基本原因有两个：数据大了、计算能力强了。深度学习模型拥有大量参数，若数据样本少，则很容易“过拟合”；如此复杂的模型、如此大的数据样本，若缺乏强力计算设备，根本无法求解。

机器学习（Machine-Learning）算法在学习过程中对某种类型假设的偏好，称为“归纳偏好”，或简称为“偏好”。任何一个有效的机器学习（Machine-Learning）算法必有其归纳偏好。

“奥卡姆剃刀”是一种常用的、自然科学研究中最基本的原则，即“若多个假设与观察一致，则选最简单的那个”。

在具体问题现实问题中，算法的归纳偏好是否与问题本身匹配，大多数时候直接决定了算法能否取得好的性能。

1.5 发展历程

机器学习（Machine-Learning）是人工智能（AI）研究发展到一定阶段的必然产物。

决策树学习技术由于简单易用，到今天仍是最常用的机器学习（Machine-Learning）技术之一。事实上，BP算法一直被应用得最广泛的机器学习（Machine-Learning）算法之一。

连接主义学习的最大局限性是其“试错性”，简单地说，其学习过程涉及大量参数，而参数的设置缺乏理论指导，主要靠手工“调参”，夸张一点说，参数调节上失之毫厘，学习结果可能谬以千里。

以往机器学习（Machine-Learning）技术在应用中取得好性能，对使用者的要求较高；而深度学习技术涉及的模型复杂度非常高，以至于只要下工夫“调参”，把参数调节好，性能往往就好。因此，深度学习虽然缺乏严格的理论基础，但它显著降低了机器学习（Machine-Learning）应用者的门槛，为机器学习（Machine-Learning）技术走向工程实践带来了便利。

深度学习如今火起来的基本原因有两个：数据大了、计算能力强了。深度学习模型拥有大量参数，若数据样本少，则很容易“过拟合”；如此复杂的模型、如此大的数据样本，若缺乏强力计算设备，根本无法求解。

机器学习（Machine-Learning）算法在学习过程中对某种类型假设的偏好，称为“归纳偏好”，或简称为“偏好”。任何一个有效的机器学习（Machine-Learning）算法必有其归纳偏好。

“奥卡姆剃刀”是一种常用的、自然科学研究中最基本的原则，即“若多个假设与观察一致，则选最简单的那个”。

在具体问题现实问题中，算法的归纳偏好是否与问题本身匹配，大多数时候直接决定了算法能否取得好的性能。

1.4 归纳偏好

机器学习（Machine-Learning）是人工智能（AI）研究发展到一定阶段的必然产物。

决策树学习技术由于简单易用，到今天仍是最常用的机器学习（Machine-Learning）技术之一。事实上，BP算法一直被应用得最广泛的机器学习（Machine-Learning）算法之一。

连接主义学习的最大局限性是其“试错性”，简单地说，其学习过程涉及大量参数，而参数的设置缺乏理论指导，主要靠手工“调参”，夸张一点说，参数调节上失之毫厘，学习结果可能谬以千里。

以往机器学习（Machine-Learning）技术在应用中取得好性能，对使用者的要求较高；而深度学习技术涉及的模型复杂度非常高，以至于只要下工夫“调参”，把参数调节好，性能往往就好。因此，深度学习虽然缺乏严格的理论基础，但它显著降低了机器学习（Machine-Learning）应用者的门槛，为机器学习（Machine-Learning）技术走向工程实践带来了便利。

深度学习如今火起来的基本原因有两个：数据大了、计算能力强了。深度学习模型拥有大量参数，若数据样本少，则很容易“过拟合”；如此复杂的模型、如此大的数据样本，若缺乏强力计算设备，根本无法求解。

机器学习（Machine-Learning）算法在学习过程中对某种类型假设的偏好，称为“归纳偏好”，或简称为“偏好”。任何一个有效的机器学习（Machine-Learning）算法必有其归纳偏好。

“奥卡姆剃刀”是一种常用的、自然科学研究中最基本的原则，即“若多个假设与观察一致，则选最简单的那个”。

在具体问题现实问题中，算法的归纳偏好是否与问题本身匹配，大多数时候直接决定了算法能否取得好的性能。

1.3 假设空间

机器学习（Machine-Learning）是人工智能（AI）研究发展到一定阶段的必然产物。

决策树学习技术由于简单易用，到今天仍是最常用的机器学习（Machine-Learning）技术之一。事实上，BP算法一直被应用得最广泛的机器学习（Machine-Learning）算法之一。

连接主义学习的最大局限性是其“试错性”，简单地说，其学习过程涉及大量参数，而参数的设置缺乏理论指导，主要靠手工“调参”，夸张一点说，参数调节上失之毫厘，学习结果可能谬以千里。

以往机器学习（Machine-Learning）技术在应用中取得好性能，对使用者的要求较高；而深度学习技术涉及的模型复杂度非常高，以至于只要下工夫“调参”，把参数调节好，性能往往就好。因此，深度学习虽然缺乏严格的理论基础，但它显著降低了机器学习（Machine-Learning）应用者的门槛，为机器学习（Machine-Learning）技术走向工程实践带来了便利。

深度学习如今火起来的基本原因有两个：数据大了、计算能力强了。深度学习模型拥有大量参数，若数据样本少，则很容易“过拟合”；如此复杂的模型、如此大的数据样本，若缺乏强力计算设备，根本无法求解。

机器学习（Machine-Learning）算法在学习过程中对某种类型假设的偏好，称为“归纳偏好”，或简称为“偏好”。任何一个有效的机器学习（Machine-Learning）算法必有其归纳偏好。

“奥卡姆剃刀”是一种常用的、自然科学研究中最基本的原则，即“若多个假设与观察一致，则选最简单的那个”。

在具体问题现实问题中，算法的归纳偏好是否与问题本身匹配，大多数时候直接决定了算法能否取得好的性能。

1.2 基本术语

机器学习（Machine-Learning）是人工智能（AI）研究发展到一定阶段的必然产物。

决策树学习技术由于简单易用，到今天仍是最常用的机器学习（Machine-Learning）技术之一。事实上，BP算法一直被应用得最广泛的机器学习（Machine-Learning）算法之一。

连接主义学习的最大局限性是其“试错性”，简单地说，其学习过程涉及大量参数，而参数的设置缺乏理论指导，主要靠手工“调参”，夸张一点说，参数调节上失之毫厘，学习结果可能谬以千里。

以往机器学习（Machine-Learning）技术在应用中取得好性能，对使用者的要求较高；而深度学习技术涉及的模型复杂度非常高，以至于只要下工夫“调参”，把参数调节好，性能往往就好。因此，深度学习虽然缺乏严格的理论基础，但它显著降低了机器学习（Machine-Learning）应用者的门槛，为机器学习（Machine-Learning）技术走向工程实践带来了便利。

深度学习如今火起来的基本原因有两个：数据大了、计算能力强了。深度学习模型拥有大量参数，若数据样本少，则很容易“过拟合”；如此复杂的模型、如此大的数据样本，若缺乏强力计算设备，根本无法求解。

机器学习（Machine-Learning）算法在学习过程中对某种类型假设的偏好，称为“归纳偏好”，或简称为“偏好”。任何一个有效的机器学习（Machine-Learning）算法必有其归纳偏好。

“奥卡姆剃刀”是一种常用的、自然科学研究中最基本的原则，即“若多个假设与观察一致，则选最简单的那个”。

在具体问题现实问题中，算法的归纳偏好是否与问题本身匹配，大多数时候直接决定了算法能否取得好的性能。

1.1 引言

机器学习（Machine-Learning）所研究的主要内容是关于在计算机上从数据中产生“模型”（model）的算法，即“学习算法”（learning algorithm）。可以说机器学习（Machine-Learning）是研究关于“学习算法”的学问。

一组记录的集合称为一个“数据集”（data set），其中每条记录是关于一个事件或对象的描述，称为一个“示例”（instance）或“样本”（sample）。反映事件或对象在某方面的表现或性质的事项，称为“属性”（attribute）或“特征”（feature），属性上的取值称为“属性值”（attribute value），属性张成的空间称为“属性空间”（attribute space）、“样本空间”（sample space）或“输入空间”。

由于空间中的每个点对应一个坐标向量，因此也把一个示例称为一个“特征向量”（feature vector）。每个示例由d个属性描述，则d称为样本的“维数”（dimensionality）。

从数据中学得模型的过程称为“学习”（learning）或“训练”（training）。训练过程中使用的数据称为“训练数据”（training data），其中每个样本称为一个“训练样本”（training sample），训练样本组成的集合称为“训练集”（training set）。

关于示例结果的信息称为“标记”（label），拥有了标记信息的示例称为“样例”（example），所有标记的集合称为“标记空间”（label space）或“输出空间”。

若预测的是离散值，此类学习任务称为“分类”（classification），如“好瓜”，“坏瓜”；若预测的是连续值，此类学习任务称为“回归”

对只涉及两个类别的“二分类”（binary classification）任务，通常称其中一个类为“正类”（positive class），另一个为“反类”（negative class）；涉及多个类别时，则称为“多分类”（multi-class classification）任务。

学得模型后，使用其进行预测的过程称为“测试”（testing），被预测的样本称为“测试样本”（testing sample）。

“聚类”（clustering）有助于我们了解数据的内在规律，能为更深入地分析数据建立基础。

根据训练数据是否拥有标记信息，学习任务可大致分为两大类：“监督学习”（supervised learning）和“无监督学习”（unsupervised learning），分类和回归是前者的代表，而聚类则是后者的代表。

学得模型适用于新样本的能力，称为“泛化”（generalization）能力。

归纳与演绎是科学推理的两大基本手段。前者是从特殊到一般的“泛化”（generalization）过程，即从具体的事例归结出一般性规律；后者则是从一般到特殊的“特化”（specialization）过程，即从基础原理推演出具体状况。



周志华《机器学习》西瓜书 手推笔记 (v2)

第二章 《模型评估与选择》

作者: 王博 (Kings)、Sophia
博士微信: **Kingsplus** (添加时请备注 学校/单位+专业)
Github: <https://github.com/Sophia-11/Machine-Learning-Notes>
(**荣登趋势榜**)
公众号【计算机视觉联盟】持续更新
后台回复**【西瓜书手推笔记】**可下载 pdf 打印版本



已完结待更笔记: 《深度学习-花书手推笔记》、《无人驾驶手推笔记》、《SLAM 十四讲》
请继续关注公众号【计算机视觉联盟】最新消息或 Github

第二章 模型评估与选择

公众号

【计算机视觉联盟】

2.1 经验误差与过拟合

{ m 个样本， a 个分类错误， 错误率 $E = \frac{a}{m}$

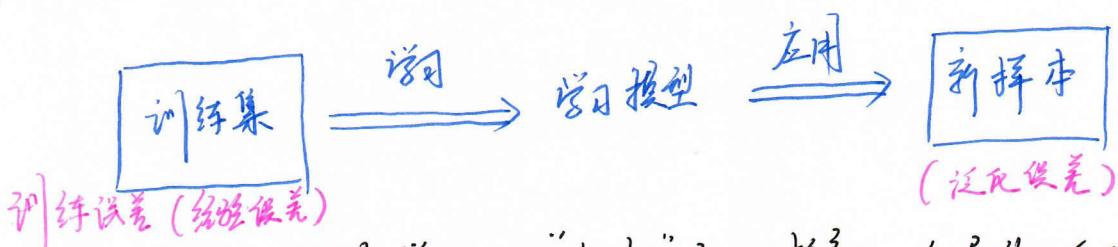
{精度 = $1 - E$ 错误率 : 精度 $1 - \frac{a}{m}$

{预测输出与样本的真实输出之间的差异称为“误差”

{学习器在训练集上的误差称为：“训练误差” or “经验误差”

在新样本上的误差称为“泛化误差”

希望误差最小化，错误率低精度高的学习器



当学习器把训练样本学的“太好”的时候，会导致泛化性下降，也就是面对新样本，效果不佳。这种现象称为“过拟合”
(与之相对，训练不够，“欠拟合”)

过拟合 学习能力过于强大，不可避免，只能缓解

欠拟合 学习能力不足，加大学习

现实中，往往有多种学习算法可供选择。甚至同一算法不同参数配置时，也会产生不同模型。如何选择，即“模型选择”。

理想解决方案是对候选模型泛化误差进行评估，然后选择泛化误差最小的模型。

第二章 模型评估与选择

公众号

【计算机视觉联盟】

2.2 评估方法

通过“测试集”来测试学习器对新样本的判别能力，之后以测试集上的“测试误差”作为“泛化误差”的近似。

测试样本：
①从样本真实分布中独立同分布采样
②与训练集尽可能互斥（未出现，未使用过）

举例：你是一个老师，教了学生 10 道题，你对他进行考核时，肯定考不是这 10 道题，才能体现他“举一反三”的能力。

但是，我们只有一个包含 m 个样本的数据集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ 如何做到既要做训练，又要测试？

对 D 适当处理，产生训练集 S 和测试集 T.

2.2.1 留出法

“留出法”直接将数据集 D 划分为两个互斥集合，一个训练集 S，一个测试集 T.

即 $D = S \cup T$, $S \cap T = \emptyset$, S 上训练, T 上评估.

举例： $D = 1000$ 个，训练 $S = 700$ 个，测试 $T = 300$ 个。

测试中有 90 个出错，错误率为 $\frac{90}{300} \times 100\% = 30\%$ ，精度 $1 - 30\% = 70\%$

注意，训练/测试划分子集尽可能与数据分布一致。

保留类别比例的采样方式称为“分层采样”

如 $D = 1000$ 个 = 500 个正 + 500 个反

则 $S = 700$ 个 = 350 个正 + 350 个反

$T = 300$ 个 = 150 个正 + 150 个反

然而，即使如此分层比例，在实际中先正，先反也会产生不同结果。

所以单次“留出法”并不可靠，一般采用若干次随机划分，重复实验取平均。

常见的方法： $\frac{2}{3} \sim \frac{4}{5}$ 样本用于训练，剩余用于测试。

第二章 模型评估与选择

公众号

【计算机视觉联盟】

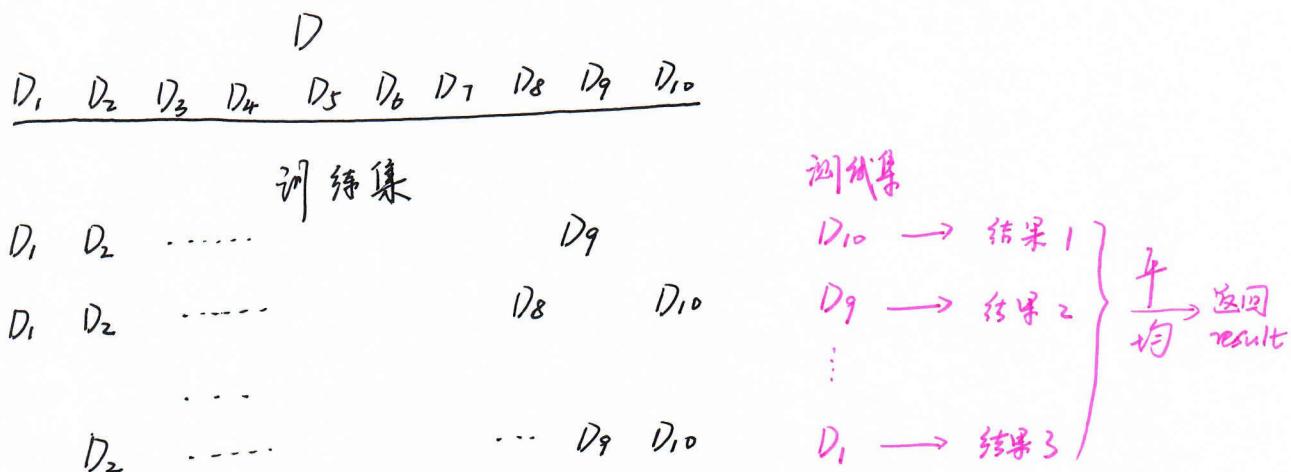
2.2.2 交叉验证法

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 都含有
(即 $D = D_1 \cup D_2 \dots \cup D_k$, $D_i \cap D_j = \emptyset (i \neq j)$)

能与数据分布保持一致，即“分层采样”。

每次用 $k-1$ 个子集的并集作为训练集，余下子集为测试集。
最终返回 k 个测试结果的均值。

又称“ k 折交叉验证”， k 通常取 10，称为 10 折交叉验证



将数据集 D 划分为 k 个子集同样存在多种划分方式，为减少
因样本划分为不同而引入的差别，通常要随机使用不同的划分
重复 P 次。最终结果是第 P 次 k 折交叉验证的均值。

常见“10 次 10 折交叉验证”

特例：留一法。

数据集 D 中包含 m 个样本，令 $k=m$ ，则每次只留 1 个测试。

留一法不受随机样本划分方式影响。

结果准确。(也不全是)

但数据量较大时，计算量太大。

2.2.3 自助法

以自助采样法为基础，给定包含 m 个样本的数据集 D ，采样 D' ：

每次从 D 中随机选一个样本，放入 D' 中，然后该样本在 D 中仍保留，使得该样本下次采样也可能被采到；重复 m 次，得到包含 m 个样本的数据集 D' (D 中有一部分在 D' 中重复出现，有一部分从未出现)

样本在 m 次采样中始终不被采到的概率：

$$\lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m \rightarrow \frac{1}{e} \approx 0.368$$

数据集 D 中大约有 36.8% 的样本未出现在训练集 D' 中，
 $D \setminus D'$ 用作测试集。

实际评估的模型与期望评估的模型都使用 m 个训练本，而
 我们仍有数据总量约 $\frac{1}{3}$ 的、没有训练集中出现，用于测试。

又称“包外估计”

使用场合： 数据量小，难以有效训练 / 测试集

此外，能产生多个不同的训练集，对集成学习有益

然而，改变了原始分布，引入估计偏差。

因此，在数据量充足时，留出法、交叉验证法更常用。

2.2.4 调参与最终模型

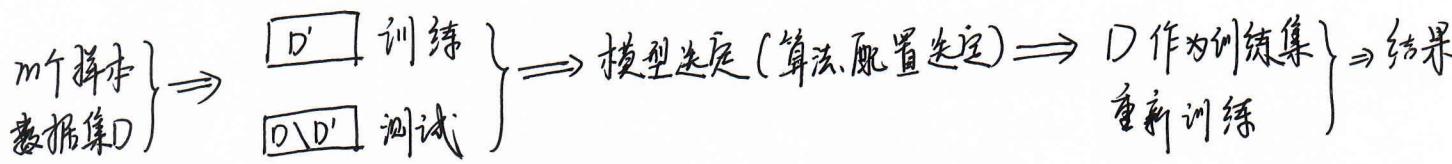
算法都有些参数需要设定，参数配置不同，模型性能不同。

“参数调节”、“调参”

调参与算法选择本质上是一致的：不同配置得到不同模型，把对应最好的模型参数作为结果。

实际操作中，参数选定是一个范围加一个步长。
如 $[0, 0.2]$ 以 0.05 为步长，有 0, 0.05, 0.1, 0.15, 0.2 这 5 种参数选择。
这已经是计算开销和性能估计的折中。

举个例子，假定 3 个参数，每个参数有 3 种选择，模型将有 $3^3 = 27$ 种需要对比。



用测试集上的判别效果来估计模型在实际使用中的泛化能力。
而把训练集分为：训练集 和 验证集。

↓
验证集上的性能进能模型选择和调参

第二章 模型评估与选择

公众号

【计算机视觉联盟】

2.3 性能度量

用来衡量模型泛化能力的评价标准。

性能度量反映了任务需求，在对比相同模型能力时，采用不同的性能度量往往会导致不同的评判结果；模型的“好坏”是相对的。

预测任务中，样例集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ 其中 y_i 是示例 x_i 的真实标记。要评估学习器 f 的性能，就要把预测结果 $f(x)$ 与真实标记 y 比较。

回归任务最常用的性能度量是“均方误差”

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

更一般的，数据分布 D 和概率密度 $p(\cdot)$ ，均方误差为

$$E(f; D) = \int_{x \sim D} (f(x) - y)^2 p(x) dx$$

2.3.1 错误率与精度

错误率是分类错误占的比例
精度是分类正确的比例 } 最常用的两种性能度量

错误率定义： $E(f; D) = \frac{1}{m} \sum_{i=1}^m I(f(x_i) \neq y_i)$

精度定义： $acc(f; D) = \frac{1}{m} \sum_{i=1}^m I(f(x_i) = y_i)$
 $= 1 - E(f; D)$

更一般表示方法：数据分布 D ，概率密度 $p(\cdot)$

错误率： $E(f; D) = \int_{x \sim D} I(f(x) \neq y) p(x) dx$

精度： $acc(f, D) = \int_{x \sim D} I(f(x) = y) p(x) dx$
 $= 1 - E(f; D)$

第二章 模型评估与选择

2.3.2 查准率、查全率与 F1

\downarrow
准确度 召回率

公众号

【计算机视觉联盟】

错误率和精度虽常用，但部分场合需求不同。例如用户关心“挑出的瓜中有多少是好瓜？这个比例是多少？”（不理解没弄，继续看）

二分类问题，根据真实类别与学习器预测类别的组合分为：

真正例 <i>true positive</i>	假正例 <i>false positive</i>	真反例 <i>true negative</i>	假反例 <i>false negative</i>
<i>TP</i>	<i>FP</i>	<i>TN</i>	<i>FN</i>

这四种情形。 $TP + FP + TN + FN = \text{样例总数}$.

分类结果混淆矩阵

真实情况	预测结果		被测值
	正例	反例	
正例	<i>TP</i> (真正例)	<i>FN</i> (假反例)	总例数
反例	<i>FP</i> (假正例)	<i>TN</i> (真反例)	

查准率 $P = \frac{TP}{TP + FP}$

查全率 $R = \frac{TP}{TP + FN}$

举例子:

总例数	120个	80个
100个正	80个	20个
100个反	40个	60个

查僻率、查全率是一对矛盾的量。

一般而言，查准率高，查全率低

查准率低，查全率高。

只有在一些简单任务中，

才能使查准率、查全率都很好。

上表含义：

-组 D = {100个正, 100个反}

预测结果 D' = {120个正, 80个反}

而120个正里实际有80个正
40个反

反星实际有20个正
60个反

$$\text{查僻率 } P = \frac{80}{80+40}$$

(准不准)

$$\text{查全率 } R = \frac{80}{80+20}$$

(全不全)

↑ 坚相加

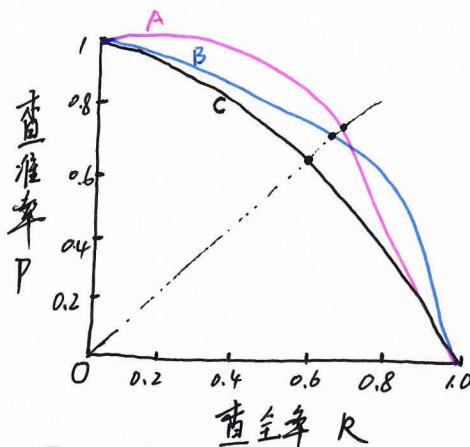
↔ 橫相加

第二章 模型评估与选择

2.3.2 查准率 P, 查全率 R

在很多情形下，根据预测结果对样例排序，排前面的“最可能”是正例的样本，排后面的“最不可能”是正例的样本。

按此顺序逐个把样本做为正例进行预测，计算前的 P, R 值。得到“P-R曲线”，称为“P-R图”。



公众号
【计算机视觉联盟】

若一个学习器的PR曲线被另一个包围，完全“包围”，则可断言前者优于后者，如A>B而A,B不能随意下定论，只能具体情况具体分析。

比较合理的方法是对比曲线下面积大小，但不好估算，于是有3个综合考虑查准率、查全率的性能度量。

① “平衡点” Break-Even Point, BEP，“查准率=查全率”的取值。

$$② F_1 \text{ 度量} : F_1 = \frac{2 \times P \times R}{P+R} = \frac{2 \times TP}{\text{样例总数} + TP - TN}$$

调和平均：

$$\frac{1}{F_1} = \frac{1}{2} \left(\frac{1}{P} + \frac{1}{R} \right)$$

由于在一些情况下，对查准率、查全率重视程度不同。

$$③ F_\beta \text{ 度量更一般形式} \quad F_\beta = \frac{(1+\beta^2) \times P \times R}{(\beta^2 \times P) + R} \quad \text{加权调和平均}.$$

$$\frac{1}{F_\beta} = \frac{1}{1+\beta^2} \left(\frac{1}{P} + \frac{\beta^2}{R} \right)$$

$\beta > 0$ 度量查全率对查准率的相对重要性。

$\beta = 1$ ，退化为 F_1

$\beta > 1$ ，查全率更重要

$\beta < 1$ ，查准率更重要。

第二章 模型评估与选择

2.3.2 查准率 P, 查全率 R, F₁

很多时候有多个二分类混淆矩阵(见前两页, 或最下方)

希望在几个二分类混淆矩阵上综合考虑查准率、查全率。

① 最直接的办法, 分别计算各混淆矩阵上的 R 和 P.

记 $(P_1, R_1), (P_2, R_2), \dots, (P_n, R_n)$, 求平均值

公众号

【计算机视觉联盟】

$$\text{宏查准率 } \text{macro-}P = \frac{1}{n} \sum_{i=1}^n P_i$$

$$\text{宏查全率 } \text{macro-}R = \frac{1}{n} \sum_{i=1}^n R_i$$

$$\text{宏 } F_1 \quad \text{macro-}F_1 = \frac{2 \times \text{macro-}P \times \text{macro-}R}{\text{macro-}P + \text{macro-}R}$$

② 还可以将各混淆矩阵对应元素平均, 得到 TP, FP, TN, FN 的平均值, 分别记 $\bar{TP}, \bar{FP}, \bar{TN}, \bar{FN}$.

再根据这些计算

$$\text{微查准率 } \text{micro-}P = \frac{\bar{TP}}{\bar{TP} + \bar{FP}}$$

$$\text{微查全率 } \text{micro-}R = \frac{\bar{TP}}{\bar{TP} + \bar{FN}}$$

$$\text{微 } F_1 \quad \text{micro-}F_1 = \frac{2 \times \text{micro-}P \times \text{micro-}R}{\text{micro-}P + \text{micro-}R}$$

		预测情况	
		正例	反例
真实	正例	TP	TN
	反例	FP	TN

第二章 模型评估与选择

2.3.3 ROC 与 AUC
 (Receiver Operating Characteristic 受试者工作特征
 Area Under ROC Curve ROC 曲线下面积)

很多学习器是为测试样本产生一个实值或概率预测，将其与分类阈值 threshold 作比较，大于阈值为正类，小于阈值为反类。

假如将实值或概率排序，“最可能”正例排最前，“最不可能”是正例排最后，分类过程相当于在这个排序中以某个截断点 cut point 将样本分为两部分。

不同任务，设定不同截断点，若更注重“查准率”，选靠前。

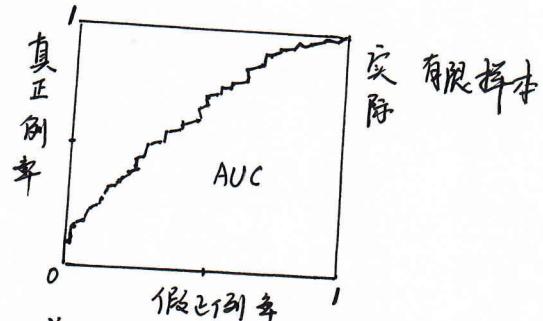
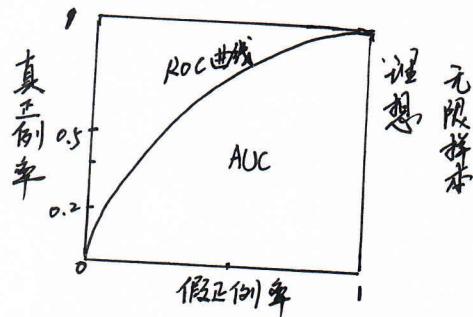
“查全率”，选靠后。

根据学习器预测结果对样例排序，按此顺序逐个把样本作为正例预测，每次计算两个值：纵轴“真正例率” True Positive Rate, TPR.

横轴“假正例率” False Positive Rate, FPR，得“ROC曲线”

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

$$FPR = \frac{FP}{TN+FP}$$



绘图过程：给定 m^+ 个正例， m^- 个反例，首先根据预测排序，然后将分类阈值设为最大，即把所有样例均预测为反例，此时 $(0,0)$ 。然后，将分类阈值依次设为每个样例预测值，依次将每个样例划分为正例。设前一个标记点坐标为 (x, y) ，当前若为正例，坐标为 $(x, y + \frac{1}{m^+})$
 当前若为反例，坐标为 $(x + \frac{1}{m^-}, y)$

公众号

【计算机视觉联盟】

学习器比较时，若一个包住另一个，可以说前者优于后者，若有交叉，则分情况，比较合理的判断是比较 ROC 曲线下面积，即 AUC。

$$AUC = \frac{1}{2} \sum_{i=1}^n (x_{i+1} - x_i)(y_i + y_{i+1})$$

损失 loss

$$L_{rank} = \frac{1}{m^+ m^-} \sum_{x \in D^+} \sum_{x' \in D^-} (\mathbb{I}(f(x^+) < f(x')) + \frac{1}{2} \mathbb{I}(f(x^+) = f(x')))$$

$$AUC = 1 - L_{rank}$$

第2章 模型评估与选择

公众号

【计算机视觉联盟】

2.3.4 代价敏感错误率与代价曲线

现实中，不同类型错误所造成后果不同。

比如：看病如果误诊

门禁如果放进了坏人

挑西瓜买到不甜的

unequal cost

为权衡不同类型错误所造成的不同损失，可将错误赋予“非均等代价”

以二分类问题，设定一个“代价矩阵” cost matrix

		二分类代价矩阵	
		预测	
真实	第0类	cost ₀₀	第1类
	第1类	cost ₁₀	cost ₁₁

损失程度相当越大，cost₀₁与cost₁₀值差别越大。
一般对比其比值。
如 5:1 等价 50:10

前面介绍的性能度量大都隐含了均等代价

$$\text{如错误率 } E(f; D) = \frac{1}{m} \sum_{i=1}^m \mathbb{I}(f(x_i) \neq y_i)$$

在非均等代价下，希望总体最小化“总体代价” total cost.

代价敏感 cost-sensitive 错误率为：

$$E(f; D; \text{cost}) = \frac{1}{m} \left(\sum_{x_i \in D^+} \mathbb{I}(f(x_i) \neq y_i) \times \text{cost}_{01} + \sum_{x_i \in D^-} \mathbb{I}(f(x_i) \neq y_i) \times \text{cost}_{10} \right)$$

非均等代价下，ROC 曲线（受试者工作特征）不能直接反映学习器的期望总代价，而“代价曲线”(cost curve) 则可达到目的。

横轴是 [0,1] 的正例概率代价

$$P(+)\text{cost} = \frac{P \times \text{cost}_{01}}{P \times \text{cost}_{01} + (1-P) \times \text{cost}_{10}}, \quad P \text{ 为样本为正例概率}$$

纵轴是 [0,1] 的均一化代价

$$\text{cost norm} = \frac{\text{FNR} \times p \times \text{cost}_{01} + \text{FPR} \times (1-p) \times \text{cost}_{10}}{P \times \text{cost}_{01} + (1-P) \times \text{cost}_{10}}$$

$$\text{FNR} = 1 - TPR$$

第2章 模型评估与选择

公众号

2.4 比较检验

为什么机器学习中性能比较非常复杂？

两个学习器不能直接比吗？

两个学习器不能直接比”。
第一、我们希望比较泛化性能，然而实验评估获得的是测试集性能。
两者对比结果未必相同。

第二. 测量集上的性能与测量集本身选择有很大关系

测试集大小，或测试集大小一致但样例差异也会导致不同结果

第三. 很多算法本身有一定的随机性, 即使相同参数设置同一个测试集多次运行, 结果可能有所不同。

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。

2.4.1 假设检验

假设检验的四步：

① 原假设：小明偷吃苹果 \Rightarrow ② 在原假设前提下计算的概率 P \Rightarrow ③ 概率 $< \alpha$ 显著水平
 备选假设：小明没吃苹果

④ 如果 $P \leq \alpha$ ，拒绝原假设，看连假设成立； $P > \alpha$ ，原假设成立。

假设检验中“假设”是对学习器泛化错误率分布的判断或猜想。

泛化错误率 $\varepsilon = \varepsilon_0$ 不知道, 要求 } \Rightarrow 用 $\hat{\varepsilon}$ 近似等于 ε_0 做为结果,
 测试错误率 $\hat{\varepsilon}$ 计算得的 }

举例]：1000个球，摸黑 0.3，白 0.7， $\varepsilon = 0.3$ 这是真分，但
你给别人后，让别人去试出来黑白倒底等于多少。

2.4.1 假设检验

泛化错误率 $\hat{\varepsilon}$ 的学习器在一个样本上犯错的概率是 ε

测试错误率 $\hat{\varepsilon}$ 意味着 m 个样本，恰有 $m \times \hat{\varepsilon}$ 个误分类。

假设泛化错误率 $\hat{\varepsilon}$ 的学习器将 m' 个样本误分类。

($m - m'$)个样本是正确分类的概率为 $\varepsilon^{m'}(1-\varepsilon)^{m-m'}$

恰好将 $\hat{\varepsilon} \times m$ 个样本误分类。

泛化错误率为 $\hat{\varepsilon}$ 的学习器被测试错误率为 $\hat{\varepsilon}$ 的概率：

$$P(\hat{\varepsilon}, \varepsilon) = C_m^{\hat{\varepsilon} \times m} \cdot \varepsilon^{\hat{\varepsilon} \times m} \cdot (1-\varepsilon)^{m - \hat{\varepsilon} \times m} \quad (2.26)$$

如何恰当理解上面公式？

泛化错误率为0.3，比如黑球

测试错误率 $\hat{\varepsilon}$ 意味着1000个样本，恰有 $1000 \cdot \hat{\varepsilon}$ 个误分

有1000个球， m' 个样本被机器分为黑色， $(m - m')$ 个被分为其它颜色。

全部都正确的概率含义： m' 个样本确实是黑色。 $(m - m')$ 个确实是其它颜色。

概率为 $\varepsilon^{m'}(1-\varepsilon)^{m-m'} = 0.3^{m'} \times 0.7^{m-m'}$

恰好选择了 $\hat{\varepsilon} \times m$ 个黑球，替换 m' 为 $\hat{\varepsilon} \times m$

$$P(\hat{\varepsilon}, \varepsilon) = C_m^{\hat{\varepsilon} \times m} \cdot \varepsilon^{\hat{\varepsilon} \times m} \cdot (1-\varepsilon)^{m - \hat{\varepsilon} \times m}$$

比如 $\hat{\varepsilon} = 0.4$ 含义：

$$P(0.4, 0.3) = C_{1000}^{400} \cdot 0.3^{400} \cdot 0.7^{600}$$

②部分是对①部分理论的解释



周志华《机器学习》西瓜书 手推笔记 (v2)

第三章

《线性模型》

作者: 王博 (Kings)、Sophia
博士微信: **Kingsplus** (添加时请备注 学校/单位+专业)
Github: <https://github.com/Sophia-11/Machine-Learning-Notes>
(**荣登趋势榜**)
公众号【计算机视觉联盟】持续更新
后台回复【**西瓜书手推笔记**】可下载 pdf 打印版本



已完结待更笔记: 《深度学习-花书手推笔记》、《无人驾驶手推笔记》、《SLAM 十四讲》
请继续关注公众号【计算机视觉联盟】最新消息或 Github

第3章 线性模型

3.1 基本形式

d 个属性描述示例 $x = (x_1, x_2, \dots, x_d)$ 其中 x_i 是 x 在第 i 个属性取值。

线性模型 (linear model) 通过属性的线性组合预测函数

$$f(x) = w_1 x_1 + w_2 x_2 + \dots + w_d x_d + b \quad (3.1)$$

↓ 向量形式

$$f(x) = w^T x + b \quad (3.2)$$

↑

$$w = (w_1, w_2, \dots, w_d)$$

w, b 学得后，模型确定

线性模型有很好的解释性，更多非线性模型可在线性模型基础上引入层级结构或高维映射可得。

3.2 线性回归

数据集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ 其中 $x_i = (x_{i1}, \dots, x_{id})$, $y_i \in R$

↓

$$D = \{x_i, y_i\}_{i=1}^m, \text{ 其中 } x_i \in R$$

“有序关系”：连续转化为连续值，如高矮 $\{1, 0\}$

高中低 $\{1, 0.5, 0\}$

“不存在有序关系”： k 个属性值， k 维向量 $(0, 0, 1)$
 $(0, 1, 0)$
 $(1, 0, 0)$ 转为向量。

线性回归试图学得

$$f(x_i) = w x_i + b \text{ 使得 } f(x_i) \approx y_i \quad (3.3)$$

公众号

【计算机视觉联盟】

第3章 线性模型

3.2 线性回归

$$f(x_i) = w x_i + b, \text{使得 } f(x_i) \approx y_i$$

↓ 如何确定 w, b ?

公众号

【计算机视觉联盟】

$$(w^*, b^*) = \underset{(w, b)}{\operatorname{arg\,min}} \sum_{i=1}^m (f(x_i) - y_i)^2$$

(3.4)

均方误差最小化

“欧式距离”

$$= \underset{(w, b)}{\operatorname{arg\,min}} \sum_{i=1}^m (y_i - w x_i - b)^2$$

↓ “最小二乘法”

$$E(w, b) = \sum_{i=1}^m (y_i - w x_i - b)^2$$

Parameter estimation

↓ 求解过程称为线性回归模型的最小二乘“参数估计”

$$\left\{ \begin{array}{l} \frac{\partial E(w, b)}{\partial w} = 2 \left(w \sum_{i=1}^m x_i^2 - \sum_{i=1}^m (y_i - b) x_i \right) \\ \frac{\partial E(w, b)}{\partial b} = 2 \left(mb - \sum_{i=1}^m (y_i - w x_i) \right) \end{array} \right. \quad (3.5)$$

$$\left\{ \begin{array}{l} \frac{\partial E(w, b)}{\partial w} = 2 \left(m b - \sum_{i=1}^m (y_i - w x_i) \right) \\ \frac{\partial E(w, b)}{\partial b} = 2 \left(mb - \sum_{i=1}^m (y_i - b) x_i \right) \end{array} \right. \quad (3.6)$$

↓ 令 (3.5), (3.6) 为零 求最优解

$$\left\{ \begin{array}{l} w = \frac{\sum_{i=1}^m y_i (x_i - \bar{x})}{\sum_{i=1}^m x_i^2 - \frac{1}{m} (\sum_{i=1}^m x_i)^2} \\ b = \dots \end{array} \right. \quad (3.7)$$

其中 $\bar{x} = \frac{1}{m} \sum_{i=1}^m x_i$

$$b = \frac{1}{m} \sum_{i=1}^m (y_i - w x_i) \quad (3.8)$$

第3章 线性模型

公众号

【计算机视觉联盟】

3.2 线性回归 多元线性回归

更一般的 d 个属性: $f(x_i) = w^T x_i + b$, 使得 $f(x_i) \approx y_i$

↓ 联想

$$f(x_i) = (w^T, b) \begin{pmatrix} x_i \\ 1 \end{pmatrix}$$

把 w 和 b 吸收为一个 $\hat{w} = (w, b)$. 数据集 D 表示为 $m \times (d+1)$ 大小矩阵.

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} & 1 \\ x_{21} & x_{22} & \cdots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{md} & 1 \end{pmatrix} = \begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \vdots & \vdots \\ x_m^T & 1 \end{pmatrix} \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

$$\hat{w}^* = \underset{\hat{w}}{\operatorname{argmin}} (y - X\hat{w})^T (y - X\hat{w}) \quad (3.9)$$

↓ 令 $E = (y - X\hat{w})^T (y - X\hat{w})$ 对 \hat{w} 求导

$$\frac{\partial E}{\partial \hat{w}} = 2X^T(X\hat{w} - y) \quad (3.10)$$

由(3.10) 式为零, 可求 \hat{w} 最优解.

$$0 = 2X^T(X\hat{w} - y)$$

$$X^T X \hat{w} - X^T y = 0$$

$$X^T X \hat{w} = X^T y$$

若 $X^T X$ 为满秩或正定矩阵.

$$(3.10) 为 \hat{w}^* = (X^T X)^{-1} X^T y \quad (3.11)$$

↓ 最终

$$f(\hat{x}_i) = \hat{x}_i^T (X^T X)^{-1} X^T y \quad (3.12)$$

如果 $X^T X$ 不满秩, 解出多个 \hat{w} , 将由算法的归一化偏好决定.

常见是引入正则化项 (regularization)

第3章 线性模型

公众号

【计算机视觉联盟】

3.2 线性回归

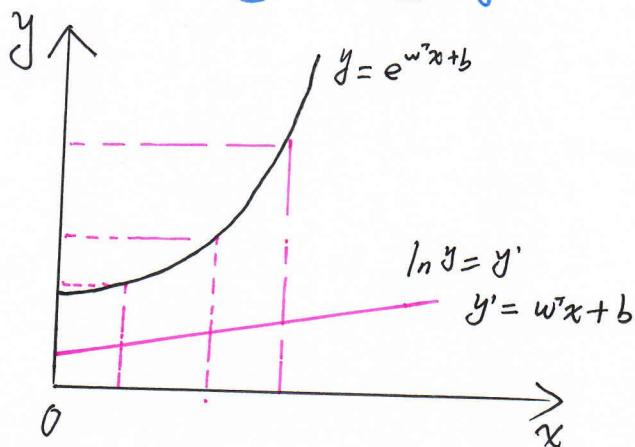
线性回归模型简写为: $y = w^T x + b$ (3.13)

对数线性回归: $\ln y = w^T x + b$

(3.14)

$$\Downarrow$$

$e^{w^T x + b}$ 逼近 y



从对数转为线性了。

广义线性模型

$$y = g^{-1}(w^T x + b) \quad (3.15)$$

$g(\cdot)$ 单调可微，“转换函数”，对数线性回归是广义的特例

3.3 对数几率回归

考虑二分类任务，输出标记 $y \in \{0, 1\}$

线性回归预测值 $z = w^T x + b$ 只需将 z 转换为 0/1

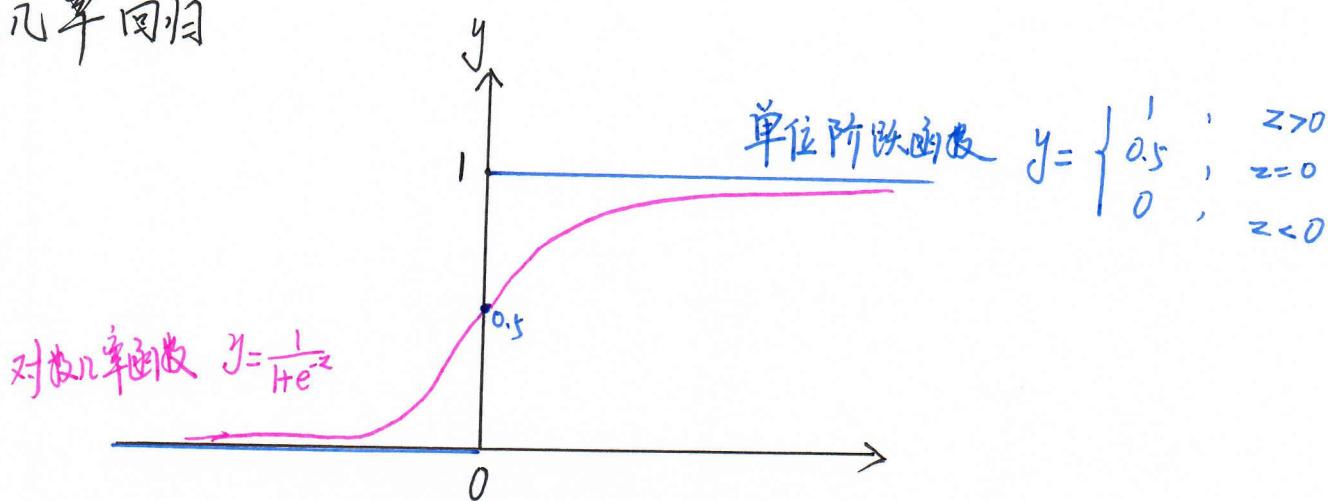
最理想的是“单位阶跃函数”(unit-step function)

$$y = \begin{cases} 0 & , z < 0 \\ 0.5 & , z = 0 \\ 1 & , z > 0 \end{cases} \quad (3.16)$$

若预测值 z 大于零判为正例，小于零判为反例。

第3章 线性模型

3.3 对数几率回归



单位阶跃函数不连续

对数几率函数 logistic function 在一定程度上近似单位阶跃函数。
是常用的替代函数：

$$y = \frac{1}{1+e^{-z}} \quad (3.17)$$

\Downarrow

代入 (3.15) $y = g^{-1}(w^T x + b)$

$$y = \frac{1}{1+e^{-(w^T x + b)}} \quad (3.18)$$

\Updownarrow

转化为

$$\ln \frac{y}{1-y} = w^T x + b \quad (3.19)$$

解读： y 视为样本正例可能性，则 $1-y$ 是其反例可能性。

“几率”反映 x 作为正例的相对可能性： $\frac{y}{1-y}$ (3.20)

对数几率： $\ln \frac{y}{1-y}$ (3.21)

(3.18) 用线性回归模型预测结果去逼近真实标记的对数几率。
称为“对数几率回归”

第3章 线性模型

3.3 对数几率回归

$$y = \frac{1}{1 + e^{-(w^T x + b)}} \Rightarrow \ln \frac{y}{1-y} = w^T x + b$$

① 将 y 视为类后验概率估计 $P(y=1|x)$

$$P(y=0|x) = 1 - P(y=1|x)$$

公众号
【计算机视觉联盟】

$$\ln \frac{P(y=1|x)}{P(y=0|x)} = w^T x + b \quad (3.22)$$



$$\left\{ \begin{array}{l} P(y=1|x) = \frac{e^{w^T x + b}}{1 + e^{w^T x + b}} \quad (3.23) \\ P(y=0|x) = \frac{1}{1 + e^{w^T x + b}} \quad (3.24) \end{array} \right.$$

$$\left\{ \begin{array}{l} P(y=1|x) = \frac{e^{w^T x + b}}{1 + e^{w^T x + b}} \quad (3.23) \\ P(y=0|x) = \frac{1}{1 + e^{w^T x + b}} \quad (3.24) \end{array} \right.$$

$$\left\{ \begin{array}{l} P(y=1|x) = \frac{e^{w^T x + b}}{1 + e^{w^T x + b}} \quad (3.23) \\ P(y=0|x) = \frac{1}{1 + e^{w^T x + b}} \quad (3.24) \end{array} \right.$$

通过“极大似然法”估计 w 和 b , 给定数据集 $\{(x_i, y_i)\}_{i=1}^m$, 对数几率回归模型最大化“对数似然” (log-likelihood)

$$L(w, b) = \sum_{i=1}^m \ln P(y_i|x_i; w, b) \quad (3.25)$$

$$\left. \begin{array}{l} \text{令 } \beta = (w; b) \\ \hat{x} = (x; 1) \end{array} \right\} \Rightarrow w^T x + b \text{ 可写为 } \beta^T \hat{x} \text{ 可用于 (3.23, 3.24)}$$

$$\left. \begin{array}{l} \text{令 } P_r(\hat{x}; \beta) = P(y=1|\hat{x}; \beta) \\ P_o(\hat{x}; \beta) = P(y=0|\hat{x}; \beta) = 1 - P_r(\hat{x}; \beta) \end{array} \right\} \Rightarrow P(y_i|x_i; w, b) = y_i P_r(\hat{x}_i; \beta) + (1 - y_i) P_o(\hat{x}_i; \beta) \quad (3.26)$$

联合 (3.26) 代入 (3.25), 用 (3.23, 3.24) 最终 (这一步推导见 A4 式, 分 $y=0, y=1$ 综合一致)

$$\text{最大化 (3.25) 等价于最小化 } L(\beta) = \sum_{i=1}^m (-y_i \beta^T \hat{x}_i + \ln(1 + e^{\beta^T \hat{x}_i})) \quad (3.27)$$

高阶导数, 但优化理论, 梯度下降, 牛顿法等.

$$\beta^* = \arg \min_{\beta} L(\beta) \quad (3.28)$$

迭代公式

$$\beta^{t+1} = \beta^t - \left(\frac{\partial^2 L(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial L(\beta)}{\partial \beta} \quad (3.29)$$

$$\beta_{\text{梯}} \left\{ \begin{array}{l} \frac{\partial L(\beta)}{\partial \beta} = - \sum_{i=1}^m \hat{x}_i (y_i - P_r(\hat{x}_i; \beta)) \quad (3.30) \\ \frac{\partial^2 L(\beta)}{\partial \beta \partial \beta^T} = \sum_{i=1}^m \hat{x}_i \hat{x}_i^T P_r(\hat{x}_i; \beta) (1 - P_r(\hat{x}_i; \beta)) \quad (3.31) \end{array} \right.$$

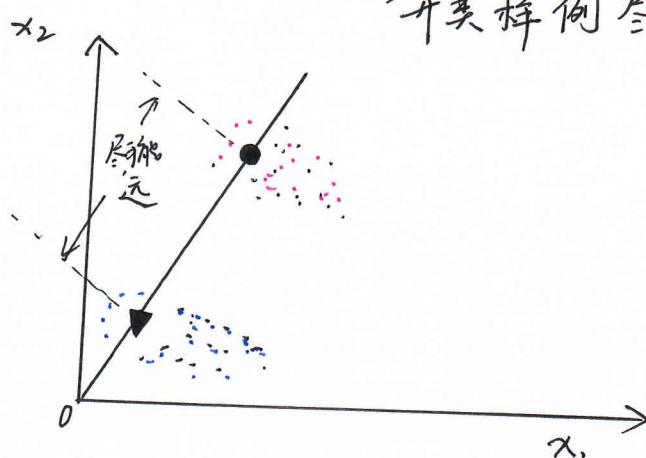
$$\beta_{\text{梯}} \left\{ \begin{array}{l} \frac{\partial L(\beta)}{\partial \beta} = - \sum_{i=1}^m \hat{x}_i (y_i - P_r(\hat{x}_i; \beta)) \quad (3.30) \\ \frac{\partial^2 L(\beta)}{\partial \beta \partial \beta^T} = \sum_{i=1}^m \hat{x}_i \hat{x}_i^T P_r(\hat{x}_i; \beta) (1 - P_r(\hat{x}_i; \beta)) \quad (3.31) \end{array} \right.$$

第3章 线性模型

3.4 线性判别分析

Linear Discriminant Analysis, LDA, 二分类：“Fisher判别分析”

LDA思想：投影到一条直线上，同类样例尽可能近
异类样例尽可能远



公众号
【计算机视觉联盟】

数据集 $D = \{(x_i, y_i)\}_{i=1}^m$, $y_i \in \{0, 1\}$, X_i , μ_i , Σ_i , $i \in \{0, 1\}$

\downarrow \downarrow \downarrow
 集合 均值向量 协方差矩阵

投影直线为 w .

两类样本中心在直线上投影为 $w^T \mu_0$, $w^T \mu_1$,

两类样本的协方差： $w^T \Sigma_0 w$, $w^T \Sigma_1 w$

① 同类样例投影点尽可能近：协方差尽可能小 $w^T \Sigma_0 w + w^T \Sigma_1 w$ 尽可能小

② 异类样例尽可能远：中心之间距离尽可能大 $\|w^T \mu_0 - w^T \mu_1\|_2^2$ 尽可能大.

同时考虑①②得最大化目标：

$$J = \frac{\|w^T \mu_0 - w^T \mu_1\|_2^2}{w^T \Sigma_0 w + w^T \Sigma_1 w} = \frac{w^T (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T w}{w^T (\Sigma_0 + \Sigma_1) w} \quad (3.32)$$

定义“类内矩阵散度”

$$S_w = \Sigma_0 + \Sigma_1$$

$$= \sum_{x \in X_0} (x - \mu_0)(x - \mu_0)^T + \sum_{x \in X_1} (x - \mu_1)(x - \mu_1)^T \quad (3.33)$$

定义类间散度矩阵

$$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T \quad (3.34)$$

$$J = \frac{w^T S_b w}{w^T S_w w} \quad (3.35)$$

第3章 线性模型

3.4 线性判别分析

$$J = \frac{w^T S_b w}{w^T S_w w} \quad (3.35)$$

LDA欲最大化目标

S_b, S_w 的“广义瑞利商”

如何求 w 呢?

令 $w^T S_w w = 1$, 最大化 J 等价于 $\min_w -w^T S_b w$ (3.36)

补充: (3.36) 拉格朗日函数为:

$$L(w, \lambda) = -w^T S_b w + \lambda (w^T S_w w - 1)$$

$$\frac{\partial L(w, \lambda)}{\partial w} = -S_b w + \lambda S_w w$$

$$\Downarrow \text{令 } = 0$$

$$S_b w = \lambda S_w w \quad (3.37)$$

公众号

【计算机视觉联盟】

$S_b w$ 方向恒为 $\mu_0 - \mu_1$

$$\text{令 } S_b w = \lambda (\mu_0 - \mu_1) \quad (3.38)$$

$$w = S_w^{-1} (\mu_0 - \mu_1) \quad (3.39)$$

S_w 怎么求?

S_w 奇异值分解: $S_w = U \Sigma V^T \Rightarrow S_w^{-1} = V \Sigma^{-1} U^T$ 即可

将 LDA 推广到多分类任务中, N 类. 第 i 类 例数为 m_i .

全局散度矩阵

$$S_t = S_b + S_w = \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T \quad (3.40)$$

类内散度矩阵

$$S_w = \sum_{i=1}^N S_{w,i} \quad (3.41)$$

$$S_{w,i} = \sum_{x \in X_i} (x - \mu_i)(x - \mu_i)^T \quad (3.42)$$

$$\max_w \frac{\text{tr}(w^T S_b w)}{\text{tr}(w^T S_w w)} \quad (3.44)$$

其中 $w \in \mathbb{R}^{d \times (N-1)}$

如何求 w 呢?

$$S_b w = \lambda S_w w \quad (3.45)$$

w 是 $S_w^{-1} S_b$ 的 $N-1$ 个最大特征值

特征值对应特征向量或矩阵.

LDA 常被视为一种经典监督降维

第3章 线性模型

公众号

3.5 多分类学习

【计算机视觉联盟】

考虑 N 个类别 $C_1, C_2 \dots C_N$, 多分类学习基本思路是“拆解法”.

拆分策略	1. “一对一” One vs. One	OvO
	2. “一对其余” One vs. Rest	OvR
	3. “多对多” Many vs. Many	MvM

给定数据集 $D = \{(x_1, y_1), (x_2, y_2) \dots (x_m, y_m)\}$, $y_i \in \{C_1, C_2 \dots C_N\}$.

OvO: N 个类别两两配对: $C_N^2 = \frac{N(N-1)}{2}$ 个二分类任务

OvR: 每次将一个类做为正例, 其他都为反例训练 N 个分类器.

如 C_1, C_2, C_3 时

OvO: C_1, C_2

C_1, C_3

C_2, C_3

OvR: $C_1 : C_2, C_3$

$C_2 : C_1, C_3$

$C_3 : C_1, C_2$

OvR 为 N 个分类器, OvO 为 $\frac{N(N-1)}{2}$ 个

OvO 存储开销, 测试时间开销比 OvR 大.

OvO 每次只用 2 个, OvR 用所有, 所以 OvO 训练时间比 OvR 小.

性能两个差不多.

MvM: 每次将若干个类作为正类, 若干个其他类作为反类.

正反构造不能随意.

常用的 MvM 技术: “纠错输出码” (Error Correcting Output Codes) ECOC

ECOC 步骤: 编码: 对 N 个类别进行 M 次划分, 每次划分一部分正, 一部分反
共产生 M 个训练集, 训练 M 个分类器.

解码: M 个分类器分别对测试样本预测, 预测组或编码.
将预测编码和类别编码比较, 返回距离最小的为最终预测.

第3章 线性模型

3.5 多分类学习

编码矩阵有多种形式，常见二元码、三元码

公众号

【计算机视觉联盟】

分离器

	f_1	f_2	f_3	f_4	f_5	
C_1	-1	+1	-1	+1	+1	
C_2	+1	-1	-1	+1	-1	
C_3	-1	+1	+1	-1	+1	
C_4	-1	-1	+1	+1	-1	

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
C_1	-1	-1	+1	+1	-1	+1	+1	4
C_2	-1				+1	-1		2
C_3	+1	+1	-1	-1	-1	+1	-1	5
C_4	-1	+1		+1	-1		+1	3

分离器

	f_1	f_2 </
--	-------	----------



周志华《机器学习》西瓜书 手推笔记 (v2)

第四章 《决策树》

作者: 王博 (Kings)、Sophia
博士微信: **Kingsplus** (添加时请备注 学校/单位+专业)
Github: <https://github.com/Sophia-11/Machine-Learning-Notes>
(**荣登趋势榜**)
公众号【计算机视觉联盟】持续更新
后台回复**【西瓜书手推笔记】**可下载 pdf 打印版本



已完结待更笔记: 《深度学习-花书手推笔记》、《无人驾驶手推笔记》、《SLAM 十四讲》
请继续关注公众号【计算机视觉联盟】最新消息或 Github

第四章 决策树

4.1 基本流程

决策树是一类常见的机器学习方法，又称“判别树”，决策过程最终结论对应了我们所希望的判定结果。

公众号

【计算机视觉联盟】

一棵决策树

一个根结点：	包含样本全集
若干个内部结点：	对各属性测试，每个结点包含的样本集合根据属性测试结果划分到子结点中。
若干个叶结点：	对应决策结果

决策树的生成是一个递归过程

有三种情形会导致递归返回

(1) 当前结点包含样本属同一类别，无需划分

(2) 当前属性集为空，所有样本在所有属性上取值相同，无法划分

(3) 当前结点包含的样本集合为空，不能划分

→ (2) 情形下，把当前结点标记为叶结点，类别 [设定为所有样本最多的类别]

→ (3) 情形下，记为叶结点，但其类别 [设定为其父结点所含样本最多的类别]。

(2) 利用当前结点的后验分布

(3) [把父结点的样本分布作为当前结点的先验分布。]

第4章 决策树

公众号

【计算机视觉联盟】

4.2 划分选择

关键在于如何选择最优划分属性

我们希望决策树分支结点所包含的样本尽可能属同一类别

即“纯度” purity 越来越高

4.2.1 信息增益

“信息熵” (information entropy) 是度量样本纯度的一种指标

假设样本集合 D 中从类样本所占比例为 P_k ($k=1, 2, \dots, |Y|$) .

$$\text{信息熵定义: } Ent(D) = - \sum_{k=1}^{|Y|} P_k \log_2 P_k \quad (4.1)$$

① 若 $P=0$ 则 $\log_2 P=0$

$Ent(D)$ 值越小，纯度越高

② $Ent(D)$ 最小值为 0

最大值为 $\log_2 |Y|$

假设离散属性 a 有 V 个可能取值 $\{a^1, a^2, \dots, a^V\}$

用 a 对 D 进行划分，产生 V 个分支结点。

第 V 个分支包含 a^V 的样本，记 D^V

考虑样本数目，赋予权重 $\frac{|D^V|}{|D|}$

利用属性 a 对样本 D 进行划分获得“信息增益” (information gain)

$$Gain(D, a) = Ent(D) - \sum_{v=1}^V \frac{|D^V|}{|D|} Ent(D^V) \quad (4.2)$$

注意，这里只是 a 对 D ，还可能计算 b 对 D ， c 对 D ...

一般而言，信息增益越大，属性 a 划分“纯度提升”越大，效果越好。

4.1 节中算法选择属性 $a_* = \underset{a \in A}{\operatorname{argmax}} Gain(D, a)$

迭代二分器

ID3 决策树学习算法是以信息增益为准则的。 (Iterative Dichotomiser)

第4章 决策树

公众号

【计算机视觉联盟】

4.2.1 信息增益

实际例子

西瓜数据集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜	↑	17个瓜
	↓ 3个取值	↓ 3个	↓ 3个	↓ 3个	↓ 3个	↓ 2个	8个好瓜 9个坏瓜		

学习一棵能预测是不是好瓜的决策树。 $|y|=2$.

数据集一共有 17 个西瓜，正例 8 个，反例 9 个

$$P_1 = \frac{8}{17} \quad P_2 = \frac{9}{17}$$

信息熵： $Ent(D) = -\sum_{k=1}^2 P_k \log_2 P_k = -\left(\frac{8}{17} \log_2 \frac{8}{17} + \frac{9}{17} \log_2 \frac{9}{17}\right) = 0.998$

计算当前属性集合 {色泽, 根蒂, 敲声, 纹理, 脐部, 触感} 每个属性信息增益.

$\begin{cases} \text{青绿} : \text{形成子集 } D^1 \Rightarrow \text{包含 6 个样例} \\ \text{乌黑} : D^2 = \{\text{乌黑}\} \Rightarrow \text{包含 6 个样例} \\ \text{浅白} : D^3 = \{\text{浅白}\} \Rightarrow \text{包含 5 个样例} \end{cases}$	$\begin{cases} \text{正例} : P_1 = \frac{3}{6} \\ \text{反例} : P_2 = \frac{3}{6} \end{cases}$	$信息熵 Ent(D^1) = -\left(\frac{3}{6} \log_2 \frac{3}{6} + \frac{3}{6} \log_2 \frac{3}{6}\right) = 1.000$
$\begin{cases} \text{正例} : P_1 = \frac{4}{6} \\ \text{反例} : P_2 = \frac{2}{6} \end{cases}$	$Ent(D^2) = -\left(\frac{4}{6} \log_2 \frac{4}{6} + \frac{2}{6} \log_2 \frac{2}{6}\right) = 0.918$	
$\begin{cases} \text{正例} : P_1 = \frac{1}{5} \\ \text{反例} : P_2 = \frac{4}{5} \end{cases}$	$Ent(D^3) = -\left(\frac{1}{5} \log_2 \frac{1}{5} + \frac{4}{5} \log_2 \frac{4}{5}\right) = 0.722$	

色泽的信息增益为：

$$\begin{aligned} Gain(D, \text{色泽}) &= Ent(D) - \sum_{v=1}^3 \frac{|D^v|}{|D|} Ent(D^v) \\ &= 0.998 - \left(\frac{6}{17} \times 1.000 + \frac{6}{17} \times 0.918 + \frac{5}{17} \times 0.722 \right) \\ &= 0.109 \end{aligned}$$

$\begin{cases} \text{同理 其他信息增益} \\ \text{Gain}(D, \text{根蒂}) = 0.143 \\ \text{Gain}(D, \text{敲声}) = 0.141 \\ \text{Gain}(D, \text{纹理}) = 0.381 \\ \text{Gain}(D, \text{脐部}) = 0.289 \\ \text{Gain}(D, \text{触感}) = 0.006 \end{cases}$	$\text{Gain}(D, \text{根蒂}) = 0.143$
	$\text{Gain}(D, \text{敲声}) = 0.141$
	$\text{Gain}(D, \text{纹理}) = 0.381$
	$\text{Gain}(D, \text{脐部}) = 0.289$
	$\text{Gain}(D, \text{触感}) = 0.006$

✓ 最大，选为划分属性

第四章 决策树

4.2.1 信息增益

纹理信息增益最大，选为划分属性

公众号
【计算机视觉联盟】

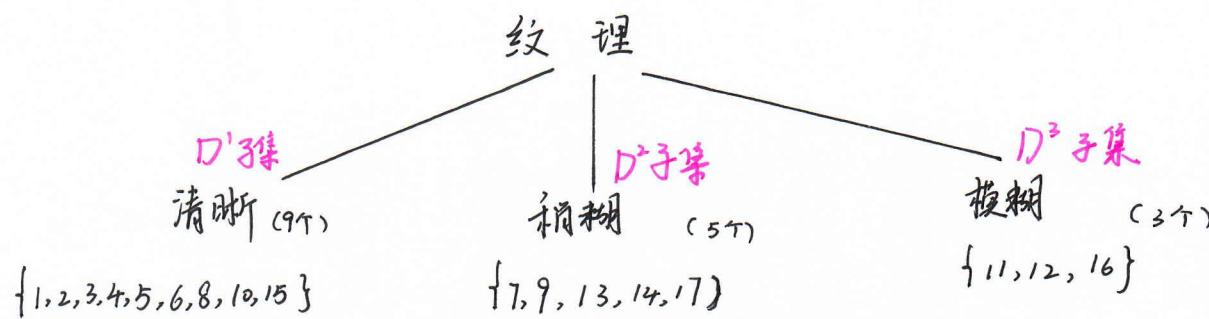


图 4.3

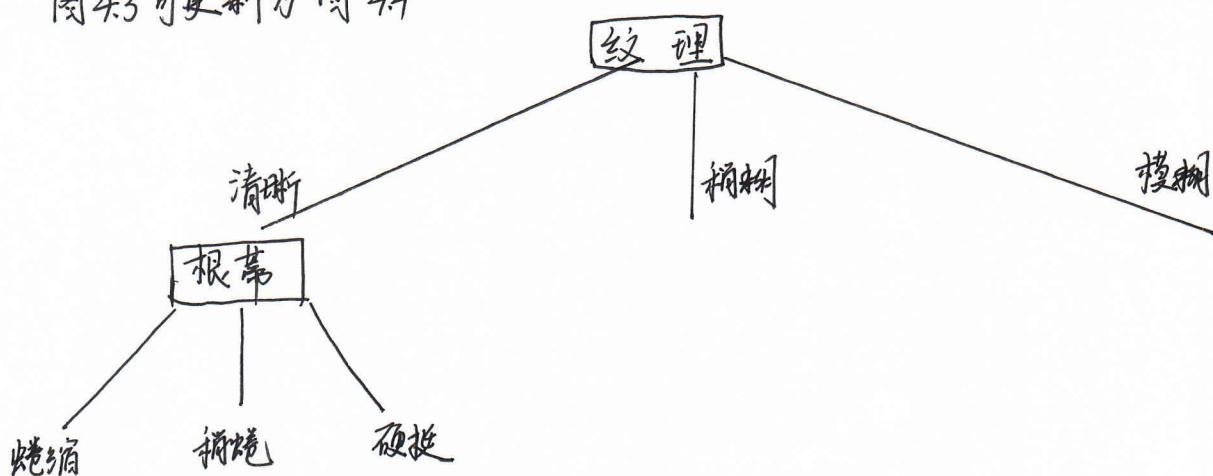
认真看这三句
 ↓
 1. 清晰为第一个分支结点为例
 包含 9 个样例，集合为 D'
 属性集合为 {色泽, 根蒂, 敲声, 膨部, 触感} 5 种

第1步还是先计算信息熵 $Ent(D')$, $Ent(D^2)$, $Ent(D^3)$

↓
 基于 D' 计算的各属性的信息增益

$Gain(D', \text{色泽}) = 0.043$ $Gain(D', \text{根蒂}) = 0.458 \checkmark$ 选其一作为属性 $Gain(D', \text{敲声}) = 0.331$ $Gain(D', \text{膨部}) = 0.458 \checkmark$ $Gain(D', \text{触感}) = 0.458 \checkmark$

图 4.3 可更新为 图 4.4



一步一步计算，一步一步更新，最后得到决策树

第4章 决策树

公众号

【计算机视觉联盟】

4.2 划分选择

4.2.2 增益率

信息增益准则对可取值数目较多的属性有所偏好.

C4.5 决策树算法不直接使用信息增益，而是使用“增益率”(gain ratio)

增益率定义: $\text{Gain-ratio}(D, a) = \frac{\text{Gain}(D, a)}{\text{IV}(a)}$ (4.3)

属性 a “固有值”
(intrinsic value) $\text{IV}(a) = -\sum_{v=1}^V \frac{|D_v|}{|D|} \log_2 \frac{|D_v|}{|D|}$ (4.4)

a 可能取值越多 (V 越大), 则 $\text{IV}(a)$ 通常也会越大

{ 触感有 2 个取值, $\text{IV}(\text{触感}) = 0.874$, $V = 2$

{ 色泽有 3 个取值, $\text{IV}(\text{色泽}) = 1.580$, $V = 3$

{ 编号有 17 个, $\text{IV}(\text{编号}) = 4.088$, $V = 17$ 无意义

增益率准则对可取值数目较少的属性有所偏好.

C4.5 并不是直接使用增益率: 先找信息增益高于平均水平的, 再选择增益率最高的.

4.2.3 基尼指数

CART 决策树使用“基尼指数”选择划分属性.

数据集 D 的纯度可用基尼值度量: $\text{Gini}(D) = \sum_{k=1}^{|Y|} \sum_{k' \neq k} P_k P_{k'} = 1 - \sum_{k=1}^{|Y|} P_k^2$ (4.5)

直观看, $\text{Gini}(D)$ 反映了从 D 中随机挑两个样本, 其类别标记不一致的概率.

$\text{Gini}(D)$ 越小, 纯度越高.

属性 a 的基尼指数定义为 $\text{Gini-index}(D, a) = \sum_{v=1}^V \frac{|D_v|}{|D|} \text{Gini}(D_v)$ (4.6)

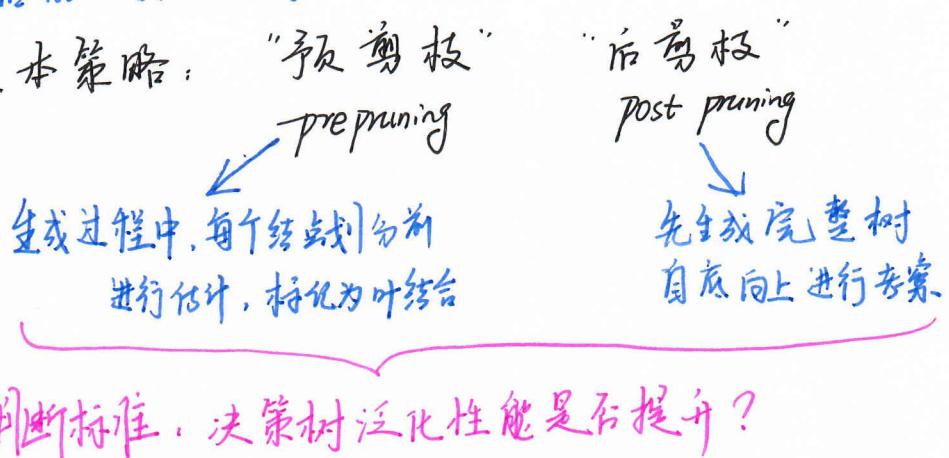
属性集合 A 中, 选得划分后基尼指数最小的属性作为最优先划分属性.

$$a_* = \arg \min_{a \in A} \text{Gini-index}(D, a)$$

4.3 剪枝处理

剪枝(pruning)是用来解决“过拟合”，比如分支过多，把训练集自身的一些特点当作所有数据的一般性质。

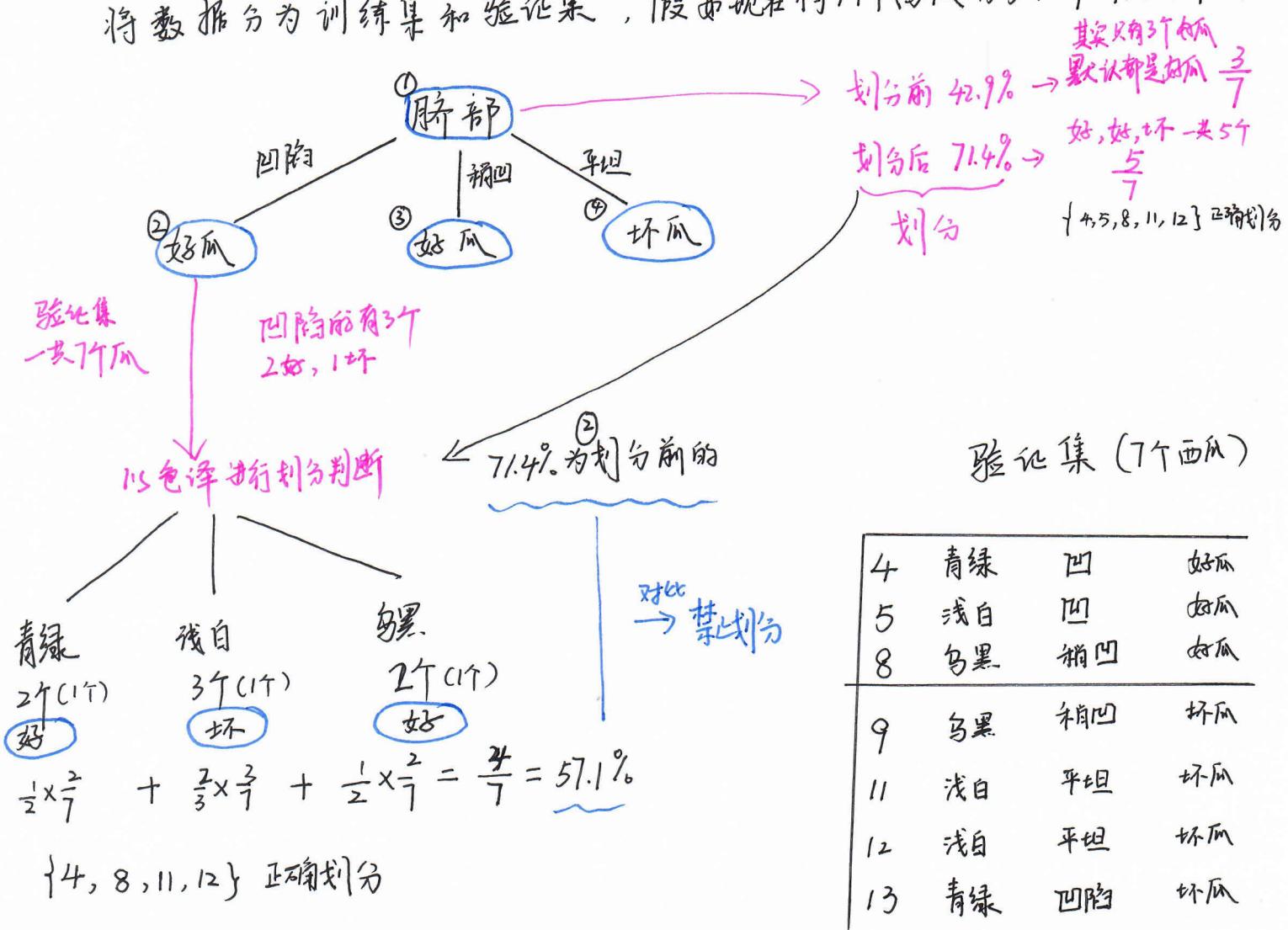
决策树剪枝的基本策略：“预剪枝” “后剪枝”



4.3.1 预剪枝

预剪枝要对划分前后的泛化能力进行估计

将数据分为训练集和验证集，假如现在将17个西瓜分为10个训练，7个验证



第4章 决策树

公众号

【计算机视觉联盟】

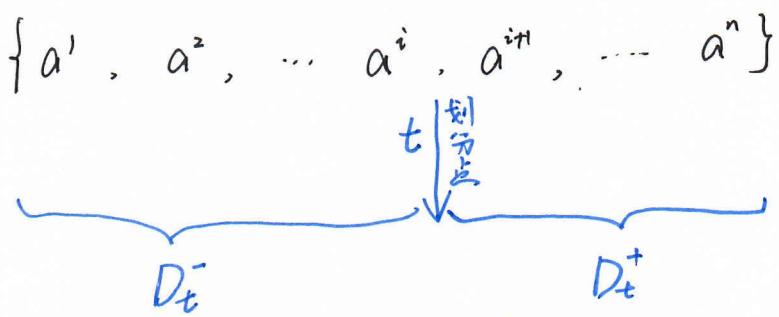
4.4 连续与缺失值

4.4.1 连续值处理

由于连续属性的可取值数目不再有限，连续属性离散化技术可派上用场。比如二分法(bi-partition)对连续属性进行处理(4.5决策树机制)

给定样本集 D ，连续属性 a

假设 a 在 D 上出现了 n 个不同取值，从小到大排序



$[a^i, a^{i+1}]$ 中取任系产生划分为结果相同
(记得 a 是连续属性)

对于连续属性 a ，我们可考虑包含 $(n-1)$ 个元素的候选划分集合。

$$(n-1) \text{ 个空隙} \quad T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \leq i \leq n-1 \right\} \quad (4.7)$$



把区间 $[a^i, a^{i+1}]$ 的中位点作为候选划分点。

回顾 4.2 “信息增益” $\text{Gain}(D, a) = \text{Ent}(D) - \sum_{v=1}^V \frac{|D_v|}{|D|} \text{Ent}(D_v)$ (4.2)



$$\text{Gain}(D, a) = \max_{t \in T_a} \text{Gain}(D, a, t)$$

$$= \max_{t \in T_a} \text{Ent}(D) - \sum_{t \in T_a} \frac{|D_t^+|}{|D|} \text{Ent}(D_t^+) \quad (4.8)$$

$\text{Gain}(D, a, t)$ 是样本集 D 基于划分点 t 二分后的信息增益。

取使 $\text{Gain}(D, a, t)$ 最大化的划分点

第四章 决策树

公众号

【计算机视觉联盟】

4.4 连续与缺失值

4.4.2 缺失值处理

比如一些缺失属性值进行训练样例。

(1) 如何在属性缺失的情况下进行划分与属性选择?

(2) 给定划分属性, 若该属性上的值缺失, 如何对样本进行划分?

数据集, “-”表缺失属性值

编号	色泽	根蒂	敲声	纹理	脐部	触感	西瓜
1	-						
2							
3			-				
4							
5	-						
6						-	
7					-		
8							
9		-					
10				-			
11					-		
12			-				
13	-						
14							
15						-	
16					-		
17		-					

给定训练集 D 和属性 a

$\{1, 2, \dots, 17\}$ {色泽, 根蒂, 敲声, 纹理, 脐部, 触感}

令 \tilde{D} 表示 D 中在属性 a 上没有缺失值的样本子集

\downarrow $\{4, 14, 16\}$ (*注意: 这里我默认 a 为 6 个属性都有, 实际请看第 3 页的例子, 取 a^1 或 a^2 为某属性而省)

假定属性 a 有 V 个可取值 $\{a^1, a^2, \dots, a^V\}$

令 \tilde{D}^v 表示 \tilde{D} 中在属性 a 上取值为 a^v 的样本子集 $\Rightarrow \tilde{D} = \bigcup_{v=1}^V \tilde{D}^v$

(比如 a^1 = 色泽 \tilde{D}^1 中 {4 是正例, {14, 16} 反例})

\tilde{D}_k 表示 \tilde{D} 中属于第 k 类 ($k = 1, 2, \dots, |y|$) 的样本子集 $\Rightarrow \tilde{D} = \bigcup_{k=1}^{|y|} \tilde{D}_k$

\downarrow 比如色泽属性下有 3 个取值 $|y|=3$

4.4 连续与缺失值

4.4.2 缺失值处理

每个样本 x 赋予一个权重 w_x . 定义

$$\rho = \frac{\sum_{x \in D} w_x}{\sum_{x \in D} w_x} \quad (4.9)$$

$|y|$ 为某属性下又有几个取值
(比如 故事属性下有3个)

$$\tilde{P}_k = \frac{\sum_{x \in \tilde{D}_k} w_x}{\sum_{x \in \tilde{D}} w_x} \quad (1 \leq k \leq |y|) \quad (4.10)$$

V 为 a 的属性取值 (假设 a 有6个取值)

$$\tilde{\gamma}_v = \frac{\sum_{x \in \tilde{D}^v} w_x}{\sum_{x \in \tilde{D}} w_x} \quad (1 \leq v \leq V) \quad (4.11)$$

ρ 为无缺失值样本所占比例

\tilde{P}_k 为无缺失值样本中第 k 类所占比例

$\tilde{\gamma}_v$ 表示无缺失值样本中在属性 a 上取值 a^v 的样本所占比例

显然 $\sum_{k=1}^{|y|} \tilde{P}_k = 1 \quad \sum_{v=1}^V \tilde{\gamma}_v = 1$

将信息增益的计算式 (4.2)

$$Gain(D, a) = Ent(D) - \sum_{v=1}^V \frac{|D^v|}{|D|} Ent(D^v)$$

↓ 推广为:

$$Gain(D, a) = \rho \times Gain(\tilde{D}, a) = \rho \times \left(Ent(\tilde{D}) - \sum_{v=1}^V \tilde{\gamma}_v Ent(\tilde{D}^v) \right) \quad (4.12)$$

$$Ent(\tilde{D}) = - \sum_{k=1}^{|y|} \tilde{P}_k \log_2 \tilde{P}_k$$

- 决策
策略
1. 若样本 x 在划分属性 a 上取值已知, 则将 x 划入与其取值对应的子结点, 且样本权值在子结点中保持 w_x
 2. 若样本 x 在划分属性 a 上取值未知, 将 x 划入所有子结点, 且样本权值在与属性值 a^v 对应的子结点中调整为 $\tilde{\gamma}_v \cdot w_x$, 让同一个样本以不同的概率划到不同的子结点中.

第四章 决策树

公众号

【计算机视觉联盟】

4.4 连续与缺失值

4.4.2 缺失值处理

举例，参见前文。 (4.5 算法与上页一样的解决方案)

D 一共有 17 个样例 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17}。
各样例权值均为 1。

以属性“色泽”为例，属性上无缺失值的样例子集

$$\tilde{D} = \underbrace{\{2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17\}}_{6 好} \quad \underbrace{\{13\}}_{8 不好} \quad \text{共 14 个}$$

$$\tilde{D} \text{ 的信息熵为: } Ent(\tilde{D}) = - \sum_{k=1}^2 \tilde{P}_k \log_2 \tilde{P}_k$$

$$= - \left(\frac{6}{14} \log_2 \frac{6}{14} + \frac{8}{14} \log_2 \frac{8}{14} \right) = 0.985$$

\tilde{D}^1	\tilde{D}^2	\tilde{D}^3	子集	$Ent(\tilde{D}^1) = - \left(\frac{2}{4} \log_2 \frac{2}{4} + \frac{2}{4} \log_2 \frac{2}{4} \right) = 1.000$
青绿	乌黑	浅白		$Ent(\tilde{D}^2) = - \left(\frac{4}{6} \log_2 \frac{4}{6} + \frac{2}{6} \log_2 \frac{2}{6} \right) = 0.918$
4 个	6 个	4 个		$Ent(\tilde{D}^3) = - \left(\frac{0}{4} \log_2 \frac{0}{4} + \frac{4}{4} \log_2 \frac{4}{4} \right) = 0.000$
2 好 + 2 坏	4+2	0+4		

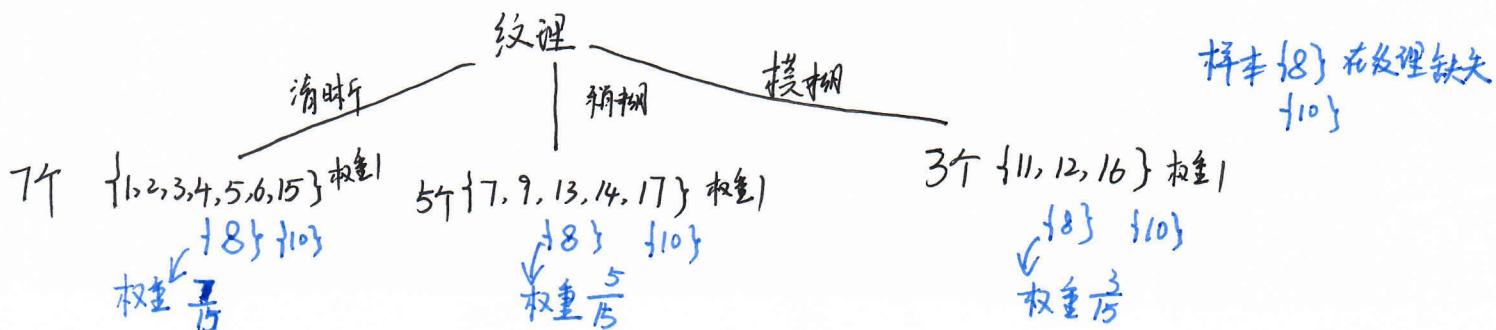
样本子集 D 上属性“色泽”的信息增益为

$$Gain(D, \text{色泽}) = Ent(D) - \sum_{i=1}^3 \tilde{P}_i Ent(\tilde{D}^i) = 0.985 - \left(\frac{4}{14} \times 1 + \frac{6}{14} \times 0.918 + \frac{4}{14} \times 0 \right) = 0.306$$

样本集 D 上属性“纹理”的信息增益为

$$Gain(D, \text{纹理}) = P \times Gain(D, \text{纹理}) = \frac{14}{17} \times 0.306 = 0.252$$

同理可得 $Gain(D, \text{纹理}) = 0.424 \dots$ “纹理”所有属性取得了最大信息增益。



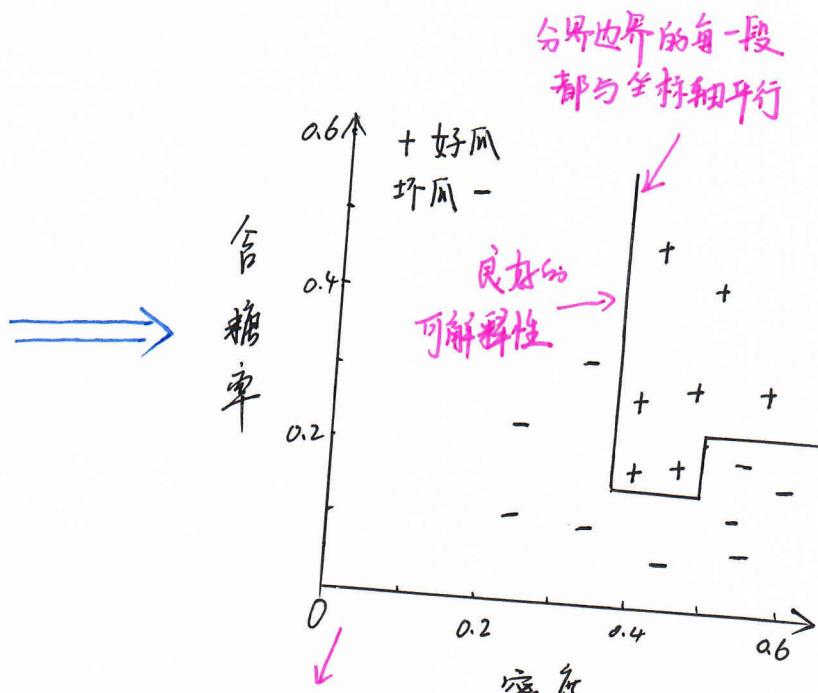
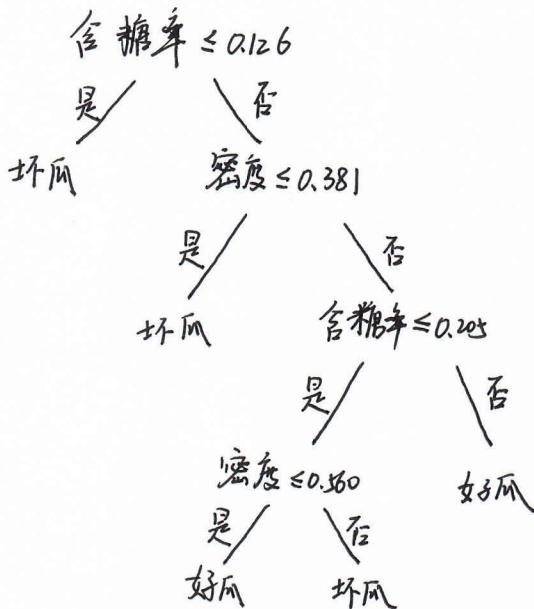
4.5 多变量决策树

若我们把每个属性视为坐标空间中的一个坐标轴，则 d 个属性描述的样本就对应了 d 维空间中的一个数据点，寻找不同样本的分类边界

决策树形成的分类边界特点：轴平行 (axis-parallel)

分类边界由若干个与坐标轴平行的平面组成。

举例：



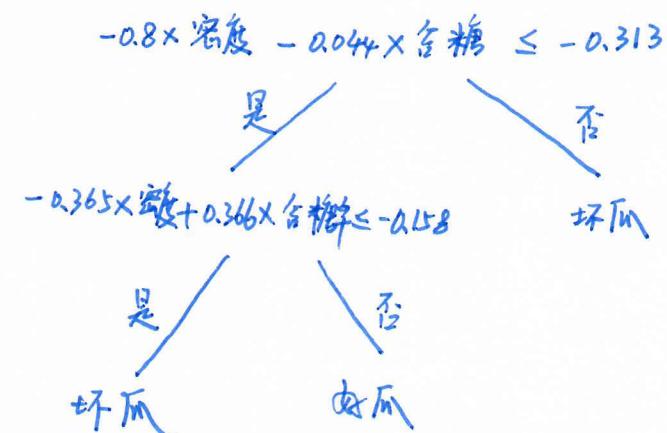
若能使用斜的划分边界，则决策树模型将大为简化

“多变量决策树”就是能实现这样的“斜划分”甚至更复杂划分的决策树。

↑ 又称

斜决策树 (oblique decision tree)

在此类决策树中，非叶结点不再是仅对某个属性，而是对属性线性组合测试；每一个非叶结点是一个形如 $\sum_{i=1}^d w_i a_i = t$ 的线性分类器。 w_i 是权重， w_i 和 t 可学习得到。不再是为每个非叶结点寻找一个最优划分属性，而是试图建立一个合适的线性分类器。





周志华《机器学习》西瓜书 手推笔记 (v2)

第五章 《神经网络》

作者: 王博 (Kings)、Sophia
博士微信: **Kingsplus** (添加时请备注 学校/单位+专业)
Github: <https://github.com/Sophia-11/Machine-Learning-Notes>
(**荣登趋势榜**)

公众号【计算机视觉联盟】持续更新
后台回复【**西瓜书手推笔记**】可下载 pdf 打印版本



已完结待更笔记: 《深度学习-花书手推笔记》、《无人驾驶手推笔记》、《SLAM 十四讲》
请继续关注公众号【计算机视觉联盟】最新消息或 Github

第5章 神经网络

公众号

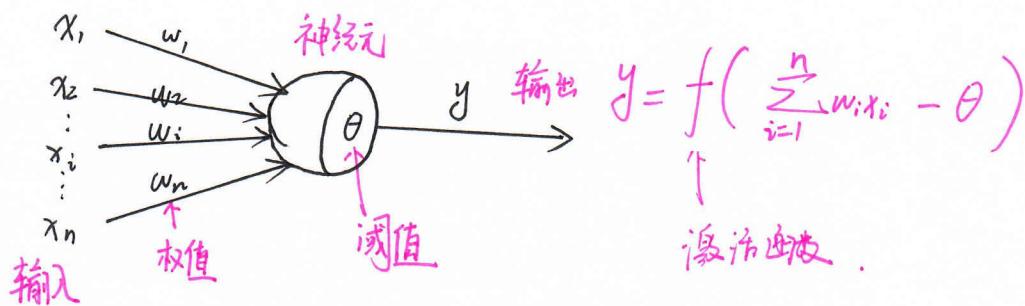
【计算机视觉联盟】

5.1 神经元模型

机器学习中谈论神经网络指“神经网络学习”

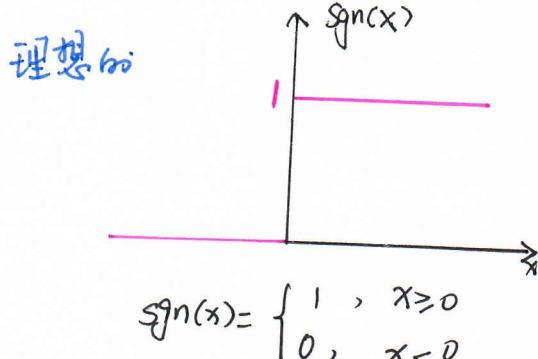
神经网络基本成分是神经元 (neuron) 模型 (unit).

1943年, McCulloch and Pitts: M-P 神经元模型

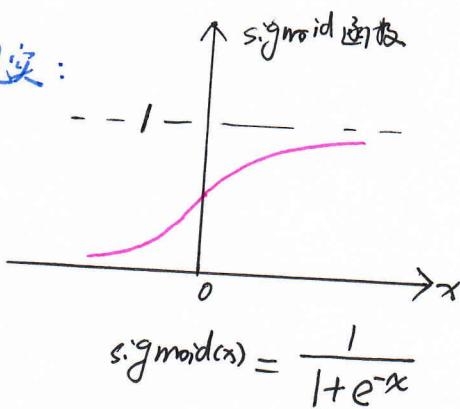


M-P 神经元模型

激活函数



现实:



阶跃函数

要么激活, 1; 要么抑制, 0

但不连续 不光滑

Sigmoid 函数

0~1 之间

可微 连续

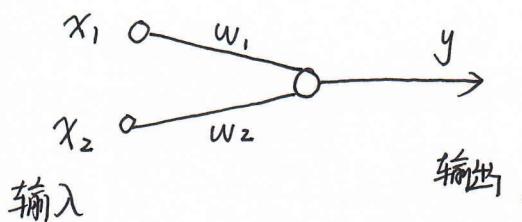
把许多这样神经元按一定层次连接就得到神经网络

第5章 神经网络

5.2 感知机与多层网络

threshold logic unit

↓
Perceptron 感知机由两层神经元组成，又称“阈值逻辑单元”



公众号
【计算机视觉联盟】

感知机可实现与、或、非运算， $y = f(\sum w_i x_i - \theta)$

“与运算” $x_1 \wedge x_2$

$$f \text{ 为阶跃函数 } \quad \text{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

$$w_1 = w_2 = 1, \theta = 2 \text{ 则 } y = f(1 \cdot x_1 + 1 \cdot x_2 - 2)$$

仅在 $x_1 = x_2 = 1$ 时 $y = 1$

“或运算” $x_1 \vee x_2$

$$w_1 = w_2 = 1, \theta = 0.5 \text{ 则 } y = f(1 \cdot x_1 + 1 \cdot x_2 - 0.5)$$

当 $x_1 = 1$ 或 $x_2 = 1$ 时 $y = 1$

“非运算” $\neg x_1$

$$w_1 = -0.6, w_2 = 0, \theta = -0.5 \text{ 则 } y = f(-0.6 \cdot x_1 + 0.5)$$

当 $x_1 = 1$ 时 $y = 0$ ， $x_1 = 0$ 时 $y = 1$

更一般的，给定数据集，权重 $w_i (i=1, 2 \dots n)$ 阈值 θ 可通过学习得到

θ 可看做 w_{n+1} ，输入永远是 -1 。

若当前感知机输出为 \hat{y} ，感知机权重调整

$$w_i \leftarrow w_i + \Delta w_i \quad (5.1)$$

$$\Delta w_i = \eta(y - \hat{y}) x_i \quad (5.2)$$

学习率 $\eta \in (0, 1)$

单层感知机只能“与”“或”“非”，非线性“异或”解决不了

两层感知机可解决“异或” 引出多层前馈神经网络

第5章 神经网络

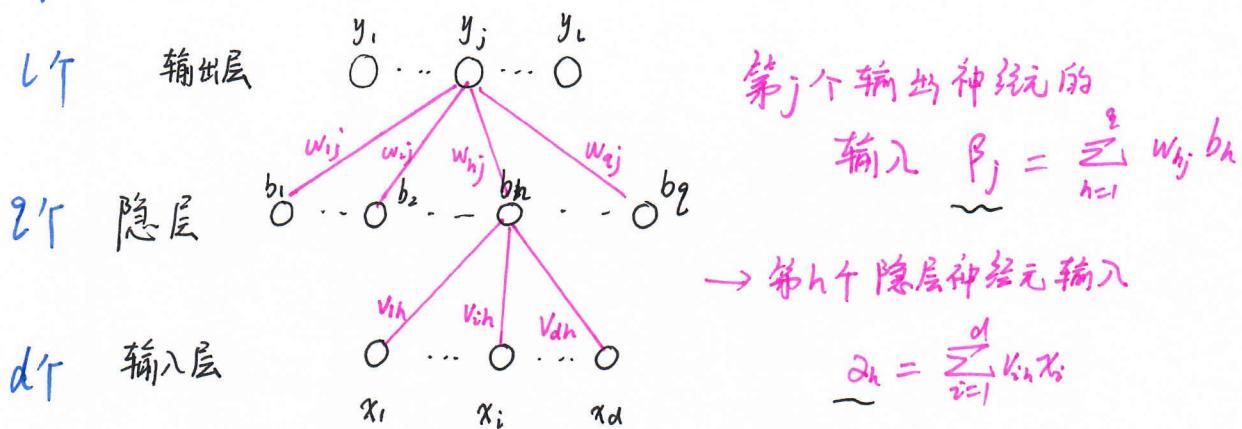
公众号

【计算机视觉联盟】

5.3 误差逆传播算法

error Back Propagation BP 是其中杰出代表

通常说“BP网络”一般指用BP算法训练多层前馈神经网络



训练例 (x_k, y_k) 假定输出为 $\hat{y}_k = (\hat{y}_1^k, \hat{y}_2^k, \dots, \hat{y}_l^k)$

$$\text{即 } \hat{y}_j^k = f(\beta_j - \theta_j) \quad (5.3)$$

网络上的均方误差为 $E_k = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j^k - y_j^k)^2 \quad (5.4)$

需要多少参数呢? $d \times l$ 个权值 $q \times l$ 个权值

q 个阈值 l 个阈值

$$d \times l + q \times l + q + l = (d+1+l)q + l \text{ 个阈值.}$$

任意参数更新公式为 $v \leftarrow v + \Delta v \quad (5.5)$

BP算法基于梯度下降策略, 以目标负梯度方向对参数调整.

以 w_{hj} 为例:

$$\Delta w_{hj} = -\eta \frac{\partial E_k}{\partial w_{hj}} \quad (5.6)$$

$$\frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}} \quad (5.7)$$

最终影响 E_k 再影响 \hat{y}_j^k 先影响 β_j

第5章 神经网络

5.3 误差逆传播算法

$$\frac{\partial E_k}{\partial w_{nj}} = \underbrace{\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j}}_{(5.3) (5.4)} \cdot \underbrace{\frac{\partial \beta_j}{\partial w_{nj}}}_{(5.7)}$$

$$(5.9) f'(x) = f(x)(1-f(x))$$

(5.7)

$$\frac{\partial \beta_j}{\partial w_{nj}} = b_h \quad (5.8)$$

$$\begin{aligned} g_j &= -\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} = -(\hat{y}_j^k - y_j^k) f'(\beta_j - \theta_j) \\ &= \hat{y}_j^k (1 - \hat{y}_j^k) (y_j^k - \hat{y}_j^k) \quad (5.10) \end{aligned}$$

公众号
【计算机视觉联盟】

合并 (5.10) (5.8) \Rightarrow (5.7) \Rightarrow (5.6)

$$\Delta W_{nj} = -\eta \frac{\partial E_k}{\partial w_{nj}} = \eta g_j b_h \quad (5.11)$$

同理 $\left\{ \begin{array}{l} \Delta \theta_j = -\eta g_j \quad (5.12) \\ \Delta V_{ih} = \eta e_h x_i \quad (5.13) \\ \Delta V_h = -\eta e_h \quad (5.14) \end{array} \right. \quad \left. \begin{array}{l} \text{中的 } e_h \\ \dots \end{array} \right.$

$$\begin{aligned} e_h &= -\frac{\partial E_k}{\partial b_h} \cdot \frac{\partial b_h}{\partial z_h} \\ &= -\sum_{j=1}^L \frac{\partial E_k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial b_h} f'(\alpha_h - \gamma_h) \\ &= \sum_{j=1}^L w_{nj} g_j f'(\alpha_h - \gamma_h) \\ &= b_h (1 - b_h) \sum_{j=1}^L w_{nj} g_j \quad (5.15) \end{aligned}$$

BP目标是最小化训练集 D 上的累积误差

$$E = \frac{1}{m} \sum_{k=1}^m E_k \quad (5.16)$$

如何缓解 BP 网络过拟合？

(1) “早停”：数据分训练集和验证集，训练集用于计算梯度、更新权、阈值。
验证集来估计误差，若训练集误差降低但验证集升高，则停止训练。

(2) “正则化”：在误差目标函数上增加一个可用于描述网络复杂度部分，如权与阈值的平方。

$$E = \lambda \frac{1}{m} \sum_{k=1}^m E_k + (1-\lambda) \sum_i w_i^2$$

$\lambda \in [0, 1]$ 用于对神经误差和网络复杂度两项折中

第5章 神经网络

5.4 全局最小与局部极小

↓
global minimum ↓
local minimum

公众号
【计算机视觉联盟】

对 w^* 和 θ^* ，若存在 $\varepsilon > 0$ 使得

$$f(w; \theta) \in \{(w; \theta) | \| (w; \theta) - (w^*; \theta^*) \| \leq \varepsilon\}$$

都有 $E(w; \theta) \geq E(w^*; \theta^*)$ 成立，则 $(w^*; \theta^*)$ 为局部极小解。

若任意 $(w; \theta)$ 都有 $E(w; \theta) \geq E(w^*; \theta)$ ，则为全局最小解。

人们常采用以下策略来试图“跳出”局部极小，从而进一步接近全局最小：

(1) 以多组不同参数值初始化多个神经网络，按标准方法训练后，取其中误差最小的解作为最终参数。

相当于从多个不同的初始点开始搜索，陷入不同的局部极小，从而选择有可能获得更接近全局最小的结果。

(2) 使用“模拟退火”技术。 (simulated annealing)

模拟退火每一步都以一定概率接受比当前解更差的结果。

每步迭代中，接受“次优解”的概率要随时间推移而降低，保证算法稳定性。

(3) 使用随机梯度下降

在计算梯度时加入了随机因素。

即使陷入局部极小点，计算的梯度仍可能不为零。

遗传算法 (genetic algorithms) 也可用来训练神经网络以更好地逼近全局最小。

上述方法理论上不够，重启方式

第5章 神经网络

公众号

【计算机视觉联盟】

5.5 其他常见神经网络

5.5.1 RBF 网络

RBF (Radial Basis Function, 径向基函数) 网络是一种单隐层前馈神经网络，使用径向基函数作为隐层神经元激活函数。输出层则是对隐层神经元输出的线性组合。

假定输入为 d 维向量 x , 输出为实值。
→ 隐层神经元个数

RBF 可表示为：

$$\phi(x) = \sum_{i=1}^q w_i \rho(x, c_i)$$

权重
↓
中心
↓
径向基函数
↓
定义

(5.18)

某部沿径向对称的标量函数，通常定义为样本 x 到数据中心 c_i 之间
欧式距离的单调函数。

高斯径向基函数形如：

$$\rho(x, c_i) = e^{-\beta_i \|x - c_i\|^2}$$

(5.19)

具有足够多隐层神经元的 RBF 网络能以任意精度逼近连续函数。

两步：①确定神经元中心 c_i . 随机采样. 聚类

②利用 BP 确定 w_i 和 β_i .

5.5.2 ART 网络

什么是竞争型学习？

竞争型学习 (competitive learning) 是神经网络一种常用的无监督学习策略。输出神经元相互竞争，有一时刻仅有一个竞争获得胜利的神经元激活，其它被抑制。

这种称为“胜者通吃” (winner-take-all) [规则]

第5章 神经网络

公众号

【计算机视觉联盟】

5.5 其他常见神经网络

5.5.2 ART网络

ART (Adaptive Resonance Theory, 自适应谐振网络理论) 是竞争性学习代表。网络由 比较层、识别层、识别阈值和重置模块构成。

↓ ↓
接收输入样本 每个神经元对应一个模式类，神经元数目可在训练过程中动态增长。
增加新的模式类

- 竞争最简单的方求
1. 计算输入向量与每个识别神经元所对应的模式类的代表向量之间的距离
距离最小者胜
 2. 获胜神经元向其他识别层神经元发送信号，抑制激活
 3. 输入与获胜神经元的相似度大于阈值，归类该属性
 4. 重新连接权重，后期相似样本计算更大相似度
 5. 若相似度小于阈值，增设一个新的神经元。当前输入为代表向量。

识别阈值

阈值高，输入样本会被细分为多种类别	stability - plasticity dilemma
阈值低，产生比较少，比较粗略的类	

ART比较好地缓解了竞争型学习中的“可塑性-稳定性窘境”。

可塑性指有学习新知识的能力

稳定性指学习新知识时要保持对旧知识的记忆。

于是优点：可进行增量学习 或在线学习。

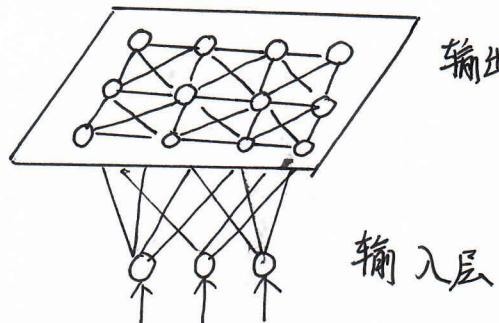
incremental learning

online learning

5.5 其他常见神经网络

5.5.3 SOM 网络 (Self-Organizing Map)

SOM 网络是一种竞争学习型的无监督神经网络，能将高维映射到低维，同时保持高维拓扑结构。高维相似点映射为输出层邻近神经元



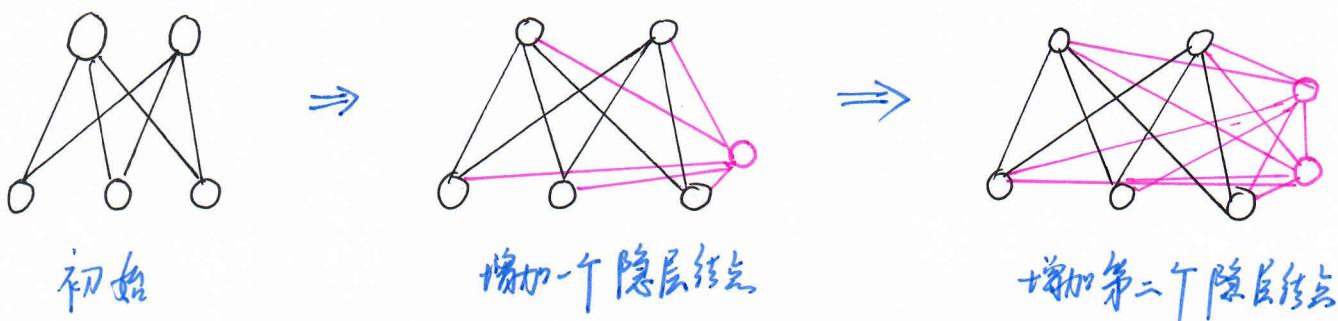
输出层：神经元以矩阵式排列二维空间
每个神经元都有一个权向量。

步骤：接收样本，每个输出层神经元会计算该样本与自身携带的权向量的距离
距离最近获胜，称最佳匹配单元。

周围神经元权向量调整，使得权向量与当前输入样本距离缩小。
不断迭代，直至收敛。

5.5.4 级联相关网络 (Cascade-Correlation)

结构自适应网络将网络结构也当作学习目标，希望找到合适网络结构。
重要代表。



级联：建立层次连接的结构，开始时只有输入输出层；随着训练新的隐层加入。

与一般前馈神经网络相比，级联相关网络无需设置网络层次、隐层神经元数目，且训练速度快。但数据较少时容易过拟合。

第5章 神经网络

5.5 其他常见神经网络

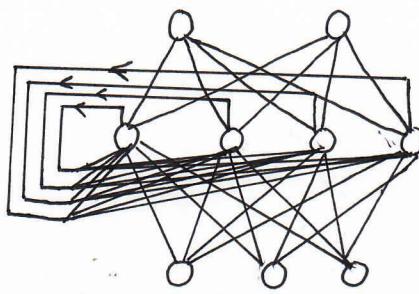
5.5.5 Elman 网络

公众号

【计算机视觉联盟】

“递归神经网络” (recurrent neural networks) 允许出现环形结构，从而让一些神经元的输出反馈回来作为输入信号。

使得 t 时刻输出状态不仅与 t 时刻输入有关，还与 $t-1$ 时刻网络状态有关，从而能处理与时间有关的动态变化。



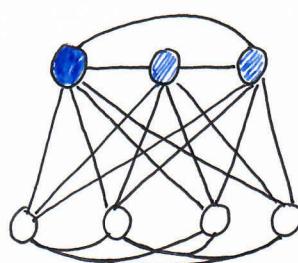
Elman 网络结构

与多层前馈网络相似，但隐层神经元输出被反馈回来，与下一时刻输入一起作为下一次隐层输入。sigmoid 激活函数。BP 算法。

5.5.6 Boltzmann 机

神经网络有一类模型是为网络定义一个“能量”，能量最小即为理想。训练就是为最小化此能量函数。

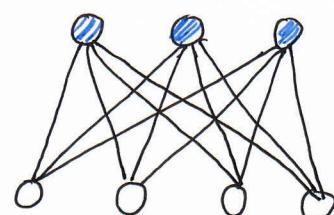
Boltzmann 机就是一种“基于能量的模型” (energy-based-model)



(a) Boltzmann 机

隐层

显层



(b) 受限 Boltzmann 机：RBM
应用

神经元分两层

显层	用于数据的输入和输出
隐层	数据的内在表达

第5章

神经网络

公众号

【计算机视觉联盟】

5.5 其他常见神经网络

5.5.6 Boltzmann 机

Boltzmann 机中神经元都是布尔型，取 0, 1 状态
 抑制。激活

令向量 $s \in \{0, 1\}^n$ 表示 n 个神经元状态， w_{ij} 表示神经元 i, j 权。

θ_i 表示神经元 i 的阈值。状态向量 s 所对应 Boltzmann 机能量意义为：

$$E(s) = - \sum_{i=1}^{n-1} \sum_{j=i+1}^n w_{ij} s_i s_j - \sum_{i=1}^n \theta_i s_i \quad (5.20)$$

若神经元以任意不依赖输入层顺序更新，则最终将达到 Boltzmann 分布。
 此时状态向量 s 出现的概率将仅由其他能量 所有可能状态向量的能量确定：

$$P(s) = \frac{e^{-E(s)}}{\sum_t e^{-E(t)}} \quad (5.21)$$

训练过程：每个样本视为一个状态向量，使其出现概率尽可能大。

标准 Boltzmann 机复杂度高，现实中常采用受限 Boltzmann 机 (RBM)

RBM 常用“对比散度”(Contrastive Divergence, CD) 进行训练。

假设 d 个显层神经元

q 个隐层神经元

v 和 h 分别表示显层与隐层状态向量。

$$P(v|h) = \prod_{i=1}^d P(v_i|h) \quad (5.22)$$

$$P(h|v) = \prod_{j=1}^q P(h_j|v) \quad (5.23)$$

每个样本：先根据 (5.23) 算隐层神经元状态分布概率

根据概率分布采样得 h'

类似从 (5.22) 从 h 产生 v'

... 采样 h'

权更新公式：

$$\Delta W = \eta (vh^\top - v'h'^\top) \quad (5.24)$$

5.6 深度学习

(1) 典型的深度学习模型就是很深层的神经网络，增加隐层数目

(2) 然而，多隐层神经网络难以直接用经典算法（如BP）进行训练，因为误差在多隐层内逆传播时，往往会“发散”（diverge）而不能收敛。

(3) 无监督逐层训练是多隐层网络训练的有效手段。

(unsupervised layer-wise training)

训练时将上一层隐节点的输出作为输入，而本层隐节点的输出作为下一层隐节点的输入，称为“预训练”(pre-training)；预训练完成后，对整个网络“微调”。

(4) 深度置信网络 (deep belief network, DBN) (Hinton 2006) 每层都是一个受限 Boltzmann

整个网络可视为若干个 RBM 堆叠而成，使用无监督逐层训练时，第一层 RBM 训练
然后第二层预训练，... 完成后，利用 BP 算法对整个网络训练。

(5) “预训练+微调”可分为将大量参数分组，局部较优聚类为主全局寻优。
有效节省训练开销。

(6) “权共享”可节省训练开销，让一组神经元使用相同的连接权。

此策略在 CNN 中发挥了重要作用

(7) “深度学习”又为“特征学习”或“表示学习”

通过多层处理，逐渐将初始“低层”特征转化为“高层”表示，用“简单模型”
完成复杂的分类学习任务。



周志华《机器学习》西瓜书 手推笔记 (v2)

第六章

《支持向量机》

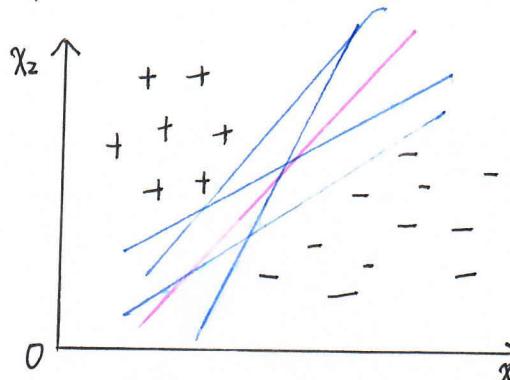
作者: 王博 (Kings)、Sophia
博士微信: **Kingsplus** (添加时请备注 学校/单位+专业)
Github: <https://github.com/Sophia-11/Machine-Learning-Notes>
(**荣登趋势榜**)
公众号【计算机视觉联盟】持续更新
后台回复**【西瓜书手推笔记】**可下载 pdf 打印版本



已完结待更笔记: 《深度学习-花书手推笔记》、《无人驾驶手推笔记》、《SLAM 十四讲》
请继续关注公众号【计算机视觉联盟】最新消息或 Github

6.1 间隔与支持向量

给定训练样本 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, $y_i \in \{-1, +1\}$, 分类学习最基本的想法是基于训练集 D 在样本空间找到一个划分超平面, 将不同类别样本分开.



应该找位于两类训练样本“正中间”的划分超平面, 如红色的.
泛化能力最强, 灵活性最强

划分超平面可通过如下线性方程描述:

$$w^T x + b = 0 \quad (6.1)$$

$w = (w_1; w_2; \dots; w_d)$ 为法向量

位移量, 决定超平面与原点之间距离

划分超平面由法向量 w 和位移 b 确定, 记为 (w, b)

任意点 x 到超平面 (w, b) 的距离可写为:

$$\gamma = \frac{|w^T x + b|}{\|w\|} \quad (6.2)$$

假设超平面 (w, b) 可将训练样本正确分类

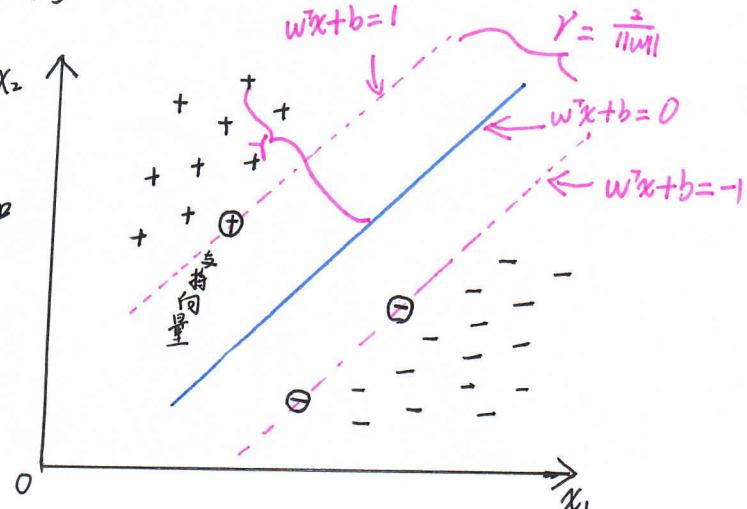
$$\left. \begin{array}{l} \text{即 } (x_i, y_i) \in D, \text{ 若 } y_i = +1 \text{ 则 } w^T x_i + b > 0 \\ \text{若 } y_i = -1 \text{ 则 } w^T x_i + b < 0 \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} w^T x_i + b \geq 1, y_i = +1 \\ w^T x_i + b \leq -1, y_i = -1 \end{array} \right. \quad (6.3)$$

“支持向量” support vector
两个异类支持向量到超平面距离和

为

$$\gamma = \frac{2}{\|w\|} \quad (6.4)$$

“间隔”



6.1 间隔与支持向量

欲找到具有“最大间隔”(maximum margin)的划分超平面，也就是要找满足约束的 w 和 b ，使得 γ 最大。即：

$$\max_{w,b} \frac{2}{\|w\|} \quad \text{s.t. } y_i (w^T x_i + b) \geq 1, \quad i=1, 2, \dots, m \quad (6.5)$$

↓ 重写

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad \text{s.t. } y_i (w^T x_i + b) \geq 1, \quad i=1, 2, \dots, m \quad (6.6)$$

↓ 约束

6.2 对偶问题

求解(6.6)来得到最大划分为对偶模型。

$$f(x) = w^T x + b \quad (6.7)$$

使用拉格朗日乘子法
得“对偶问题”

↓ 对(6.6)每条约束添加拉格朗日乘子 $\alpha_i \geq 0$

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i (w^T x_i + b)) \quad (6.8)$$

其中
 $\alpha = (\alpha_1; \alpha_2; \dots; \alpha_m)$

↓ 令 $\mathcal{L}(w, b, \alpha)$ 对 w 和 b 偏导为零

$$w = \sum_{i=1}^m \alpha_i y_i x_i \quad (6.9)$$

$$b = \sum_{i=1}^m \alpha_i y_i \quad (6.10)$$

将(6.9)代入(6.8)
消去 w 和 b

↓ 考虑(6.10)约束 得(6.6)变形的对偶问题。

$$\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (6.11)$$

$$\text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i=1, 2, \dots, m$$

第6章 支持向量机

公众号

【计算机视觉联盟】

6.2 对偶问题

解出 α 后，求出 w 与 b 即可得模型

$$f(x) = w^T x + b = \sum_{i=1}^m \alpha_i y_i x_i^T x + b \quad (6.12)$$

从对偶问题 (6.11) $\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j$ 解出的 α_i 是

(6.8) 中 $L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i (w^T x_i + b))$ 中的拉格朗日乘子，恰对应着训练样本 (x_i, y_i) 。

上述过程满足 KKT 条件

Karush-Kuhn-Tucker

$$\begin{cases} \alpha_i \geq 0 \\ y_i f(x_i) - 1 \geq 0 \\ \alpha_i (y_i f(x_i) - 1) = 0 \end{cases} \quad (6.13)$$

对任意训练样本 (x_i, y_i) 总有 $\alpha_i = 0$ 或 $y_i f(x_i) = 1$

若 $\alpha_i = 0$ ，则样本不会在 (6.12) 中出现，不会对 $f(x)$ 有影响

若 $\alpha_i > 0$ ，则必有 $y_i f(x_i) = 1$ ，对应样本点位于最大间隔边界上，是支持向量。

支持向量机性质：

训练完成后，大部分训练样本都不被保留，最终模型仅与支持向量有关

如何求解 (6.11)？

二次规划算法正比于训练样本身数，会造成较大开销。

SMO (Sequential Minimal Optimization) 是高效算法，著名代表。

SMO 基本思路：

先固定 α_i 之外的所有参数，然后求 α_i 上的极值。

由于存在约束 $\sum_{i=1}^m \alpha_i y_i = 0$ ，若固定 α_i 之外其他变量，则 α_i 可由其他导出。

于是，SMO 每次选择两个变量 α_i 和 α_j ，并固定其它参数。

6.2 对偶问题

参数初始化后，SMO 不断执行如下两个步骤至收敛：

- 选取一对需更新的变量 α_i 和 α_j ；
- 固定 α_i 和 α_j 以外的参数，求解 (6.11) 获更新后 α_i 和 α_j 。

KKT 条件违背的程度越大，则变量更新后可能导致的目标函数值减幅越大
使选取的两变量所对应样本之间的间隔最大。

SMO 高效因为在固定其他参数后，优化两个参数的过程能做到非常高效。

仅考虑 α_i 和 α_j 时，(6.11) 约束可写为

$$\alpha_i y_i + \alpha_j y_j = C, \quad \alpha_i \geq 0, \quad \alpha_j \geq 0 \quad (6.14)$$

$$C = - \sum_{k \neq i,j} \alpha_k y_k \quad (6.15)$$

是使 $\sum_{i=1}^m \alpha_i y_i = 0$ 成立的常数。

用 $\alpha_i y_i + \alpha_j y_j = C$ (6.16) 消去式 (6.11) 中变量 α_j ，仅有的约束是 $\alpha_i \geq 0$ 。

如何确定偏移项 b ？

任意 (x_s, y_s) 都有 $y_s f(x_s) = 1$ 即

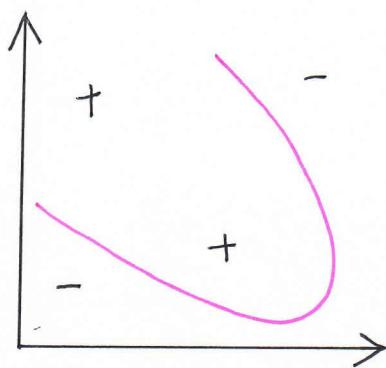
$$y_s \left(\sum_{i \in S} \alpha_i y_i x_i^\top x_s + b \right) = 1 \quad (6.17)$$

$S = \{i | \alpha_i > 0, i=1, 2, \dots, m\}$ 为所有支持向量下标集

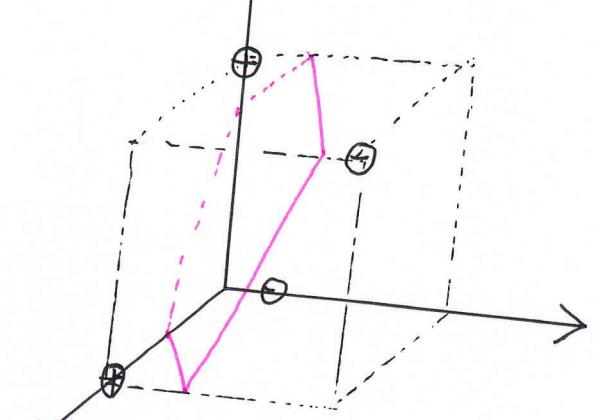
使用所有支持向量求解的平均值

$$b = \frac{1}{|S|} \sum_{s \in S} \left(y_s - \sum_{i \in S} \alpha_i y_i x_i^\top x_s \right) \quad (6.18)$$

6.3 核函数



$x \rightarrow \phi(x)$
映射高维



可将样本从原始空间映射到一个更高维的特征空间

如果原始空间是有限维，属性数有限，一定存在一个高维特征空间使样本可分

令 $\phi(x)$ 表示将 x 映射后的特征向量，特征空间中划分超平面对应模型：

$$f(x) = w^T \phi(x) + b \quad (6.19)$$

$$\begin{cases} \min_{w, b} \frac{1}{2} \|w\|^2 \\ \text{s.t. } y_i (w^T \phi(x_i) + b) \geq 1, \quad i=1, 2, \dots, m \end{cases} \quad (6.20)$$

对偶问题： $\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j)$

$$\begin{cases} \text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i=1, 2, \dots, m \end{cases} \quad (6.21)$$

求解 (6.21) 涉及计算 $\phi(x_i)^T \phi(x_j)$ ，是 x_i 和 x_j 映射到特征空间之后的内积，由于维数高，计算可能有困难。设想一个函数

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \phi(x_i)^T \phi(x_j) \quad (6.22)$$

即 x_i 与 x_j 在特征空间的内积等于通过 $k(\cdot, \cdot)$ 计算的结果

(给 $x_i, x_j, k(\cdot, \cdot)$ 函数即可计算)

6.3 核函数

式(6.21) 可重写

$$\begin{cases} \max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ \text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i=1, 2, \dots, m \end{cases} \quad (6.23)$$

求解后可得：

$$f(x) = w^\top \phi(x) + b = \sum_{i=1}^m \alpha_i y_i \phi(x_i)^\top \phi(x) + b = \sum_{i=1}^m \alpha_i y_i k(x, x_i) + b \quad (6.24)$$

函数 $k(\cdot, \cdot)$ 是“核函数”(kernel function)

(6.24) 显示模型最优解可通过训练样本的核函数展开，亦称“支持向量展开”

若已知后验映射 $\phi(\cdot)$ 具体形式 \Rightarrow 推写出核函数 $k(\cdot, \cdot)$

定理 6.1 (核函数)：

令 X 为输入空间， $k(\cdot, \cdot)$ 是定义在 $X \times X$ 的 对称函数。 k 是核函数。当且仅当对于任意数据 $D = \{x_1, x_2, \dots, x_m\}$ ，“核矩阵” K 总是半正定的。

$$K = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_j) & \dots & k(x_1, x_m) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ k(x_i, x_1) & \dots & k(x_i, x_j) & \dots & k(x_i, x_m) \\ \vdots & & \vdots & & \vdots \\ k(x_m, x_1) & \dots & k(x_m, x_j) & \dots & k(x_m, x_m) \end{bmatrix}$$

只要一个对称函数所对应的矩阵半正定，它总能作为核函数使用

对于一个半正定核矩阵，总能找到一个与之对应的映射 ϕ

任意一个核函数都隐式地定义了一个称为“再生核希尔伯特空间”

(Reproducing Kernel Hilbert Space, RKHS) 的特征空间

我们希望样本在特征空间内线性可分，因此特征空间的好坏对支持向量机的性能至关重要。

第6章 支持向量机

公众号

【计算机视觉联盟】

6.3 核函数

常用核函数

线性核

$$k(x_i, x_j) = x_i^T x_j$$

$d=1$ 退化为线性核

多项式核

$$k(x_i, x_j) = (x_i^T x_j)^d$$

$d \geq 1$ 为多项式次数

高斯核 (Gaussian) $k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\delta^2}\right)$ $\delta > 0$ 为高斯核的带宽 (width)

拉普拉斯核

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{\delta}\right) \quad \delta > 0$$

Sigmoid 核 $k(x_i, x_j) = \tanh(\beta x_i^T x_j + \theta)$ \tanh 为双曲正切函数, $\beta > 0, \theta < 0$

(1) 若 k_1 和 k_2 为核函数, 对任意正数 γ_1, γ_2 其线性组合

$$\gamma_1 k_1 + \gamma_2 k_2 \quad (6.25) \quad \text{也是核函数}$$

(2) 若 k_1 和 k_2 为核函数, 则核函数直积

$$k_1 \otimes k_2(x, z) = k_1(x, z) k_2(x, z) \quad (6.26) \quad \text{也是核函数}$$

(3) 若 k_1 为核函数, 则对于任意函数 $g(x)$

$$k(x, z) = g(x) k_1(x, z) g(z) \quad (6.27) \quad \text{也是核函数}$$

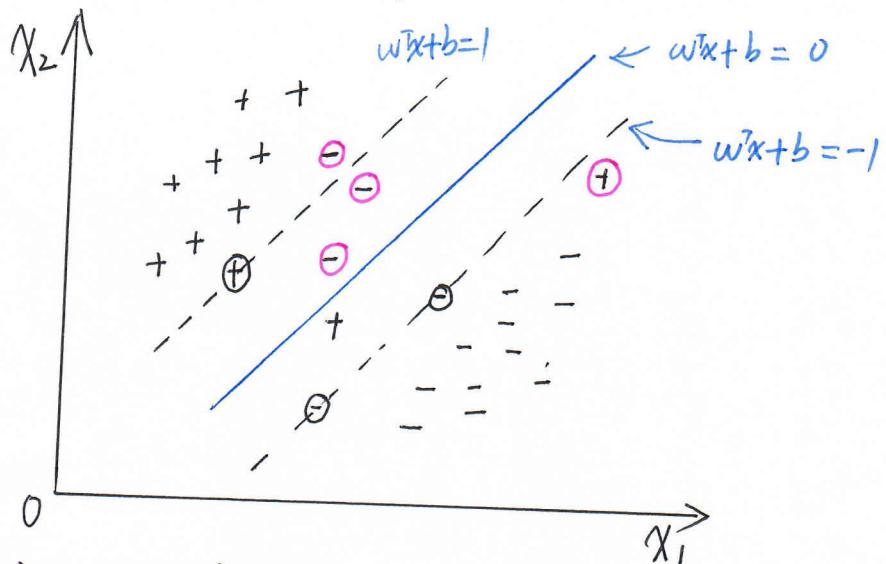
第6章 支持向量机

公众号

【计算机视觉联盟】

6.4 软间隔与正则化

前面讨论中，假定存在一个超平面可将不同类样本完全分开。然而，现实中很难出现这种完美情况，缓解该问题办法是允许支持向量机在一些样本上出错，引出“软间隔” soft margin 概念。



软间隔允许某些样本不满足约束

$$y_i(w^T x_i + b) \geq 1 \quad (6.28)$$

最大化间隔的同时，不满足约束的样本尽可能少，优化目标可写为：

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m l_{0/1}(y_i(w^T x_i + b) - 1) \quad (6.29)$$

C 是一个 $C > 0$ 常数
无穷大时，所有样本均满足(6.28)
取有限值时，允许一些样本不满足约束

$$l_{0/1}(z) = \begin{cases} 1, & \text{if } z < 0 \\ 0, & \text{otherwise.} \end{cases} \quad (6.30)$$

$l_{0/1}$ 非凸、非连续，数学性质不好。

常采用一些函数代替 $l_{0/1}$ ，称为“替代损失”(surrogate loss)。

常用
替代
损失
函数

① hinge 损失

$$l_{\text{hinge}}(z) = \max(0, 1-z) \quad (6.31)$$

② 指数损失(exponential loss)

$$l_{\text{exp}}(z) = \exp(-z) \quad (6.32)$$

③ 对数损失(logistic loss)

$$l_{\text{log}}(z) = \log(1 + \exp(-z)) \quad (6.33)$$

第6章 支持向量机

公众号

【计算机视觉联盟】

6.4 软间隔与正则化

若采用 hinge 损失 $l_{\text{hinge}}(z) = \max(0, 1-z)$

$$\text{则} (6.29) \quad \min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m l_{\text{hinge}}(y_i(w^T x_i + b) - 1)$$

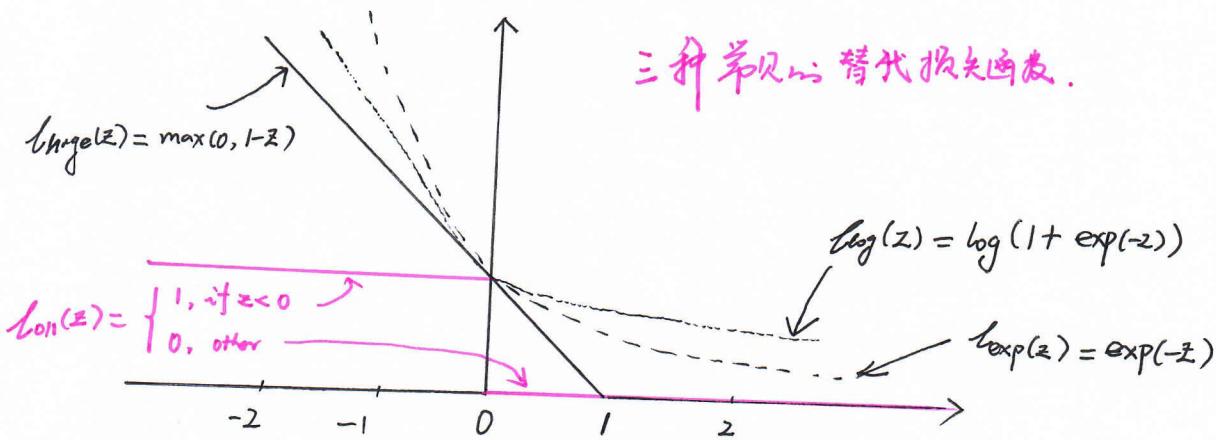
变成

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \max(0, 1 - y_i(w^T x_i + b)) \quad (6.34)$$

\Downarrow 引入“松弛变量” $\xi_i \geq 0$

$$\min_{w,b,\xi_i} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \quad (6.35)$$

常用的“软间隔支持向量机” s.t. $y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i=1, 2, \dots, m$



通过拉格朗日乘子法可得 (6.35) 的拉格朗日函数

$$L(w, b, \alpha, \xi, \mu) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i + \sum_{i=1}^m \alpha_i (1 - \xi_i - y_i(w^T x_i + b)) - \sum_{i=1}^m \mu_i \xi_i \quad (6.36)$$

$\alpha_i \geq 0, \mu_i \geq 0$ 是拉格朗日乘子

令 $L(w, b, \alpha, \xi, \mu)$ 对 w, b, ξ_i 的偏导为零可得：

$$w = \sum_{i=1}^m \alpha_i y_i x_i \quad (6.37)$$

$$0 = \sum_{i=1}^m \alpha_i y_i \quad (6.38)$$

$$C = \alpha_i + \mu_i \quad (6.39)$$

第6章 支持向量机

6.4 软间隔与正则化

将 (6.37) ~ (6.39) 代入 (6.36) 得 (6.35) 对偶问题

公众号
【计算机视觉联盟】

$$\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (6.40)$$

$$\text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, m$$

类似 (6.13) 对软间隔支持向量机, KKT 条件要求

$$\begin{cases} \alpha_i \geq 0, \quad \gamma_i \geq 0 \\ y_i f(x_i) - 1 + \gamma_i \geq 0 \\ \alpha_i (y_i f(x_i) - 1 + \gamma_i) = 0 \\ \gamma_i \geq 0, \quad n_i \gamma_i = 0 \end{cases} \quad (6.41)$$

对任意训练样本 (x_i, y_i) , 总有 $\alpha_i = 0$ 或 $y_i f(x_i) = 1 - \gamma_i$

样本不会对 $f(x)$ 有影响

$\alpha_i > 0$, 必有 $y_i f(x_i) = 1 - \gamma_i$
该样本是支持向量.

回顾 (6.39) $C = \alpha_i + n_i$

$$\left\{ \begin{array}{l} \text{若 } \alpha_i < C, \text{ 则 } n_i > 0, \text{ 则 } \gamma_i = 0, \text{ 样本在最大间隔边界上} \\ \alpha_i = C \text{ 且 } n_i = 0, \begin{cases} \text{若 } \gamma_i \leq 1 \text{ 则样本落在最大间隔内部} \\ \text{若 } \gamma_i > 1 \text{ 则样本被错误分类} \end{cases} \end{array} \right.$$

能否用对率损失函数 $\log(1 + e^{-y_i f(x)})$ 来替代损失函数?

(1) 支持向量机与对率回归优化目标相近, 通常性能相当

(2) 优势在于输出具有自然的概率意义

(3) 对率回归可直接用于多分类任务.

(4) 对率回归的解依赖于更多的训练样本, 效率开销过大

6.4 软间隔与正则化

不同梯度损失函数模型都具有共性：优化目标中的第一次用来描述划分超平面“间隔”大小，第二次 $\sum_{i=1}^m l(f(x_i), y_i)$ 用来表述训练集上误差。一般形式：

$$\min_f \underbrace{\mathcal{L}(f)}_f + C \underbrace{\sum_{i=1}^m l(f(x_i), y_i)}_{\text{经验风险}} \quad (6.42)$$

↓
结构风险
描述模型 f 的性质

↓
经验风险
描述模型与训练数据的契合程度

用于对“结构风险”、“经验风险”的折中

① (6.42) 称为“正则化”问题

$\mathcal{L}(f)$ 称为正则化项

C 正则化常数

L_p 范数 (m 项) 常用的正则化项， L_2 范数 $\|w\|_2$ ，倾向于 w 的分量取值尽量均衡
即非零分量个数尽量稠密。

L_1 范数 $\|w\|_1$ 和 L_∞ 范数 $\|w\|_\infty$ ，倾向于 w 分量尽量稀疏。
即非零分量个数尽量少。

② 从经验风险最小化角度， $\mathcal{L}(f)$ 表达我们希望获得具有任何性质的模型

为引入领域知识和用户意图提供了途径；

该信息有助于削减假设空间，从而降低了最小化训练误差的过拟合风险。

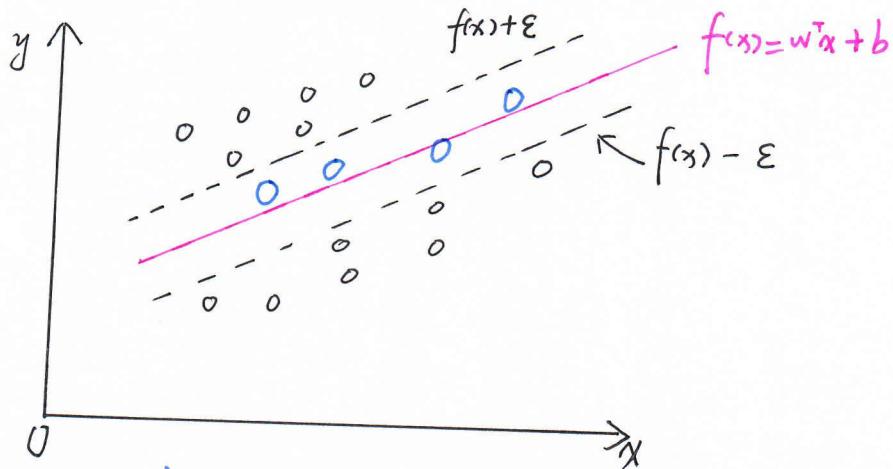
第6章 支持向量机

公众号

【计算机视觉联盟】

6.5 支持向量回归

给定训练样本 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, $y_i \in R$, 希望学得一个形如 (6.7) $f(x) = w^T x + b$ 的回归模型, 使得 $f(x)$ 与 y 尽可能相近, w, b 待求
支持向量回归 (Support Vector Regression, SVR) 假设我们能容忍 $f(x)$
与 y 之间最多有 ε 的偏差. 仅当 $f(x)$ 与 y 之间差的绝对值大于 ε 时才计算损失.



相当于以 $f(x)$ 为中点, 构建了一个宽度为 2ε 的间隔带,
若落入此带, 则认为被预测正确, 带中不计算损失.

SVR 问题可形式化为

$$\min_{w, b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m l_\varepsilon(f(x_i) - y_i) \quad (6.43)$$

正则化常数

l_ε 和 ε 不敏感损失函数

$$l_\varepsilon(z) = \begin{cases} 0 & , \text{ if } |z| \leq \varepsilon \\ |z| - \varepsilon & , \text{ otherwise.} \end{cases} \quad (6.44)$$

引入松弛变量 ξ_i 和 $\hat{\xi}_i$

(6.43) 重写为

$$\min_{w, b, \xi_i, \hat{\xi}_i} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\xi_i + \hat{\xi}_i) \quad (6.45)$$

$$\text{s.t. } f(x_i) - y_i \leq \varepsilon + \xi_i$$

$$y_i - f(x_i) \leq \varepsilon + \hat{\xi}_i$$

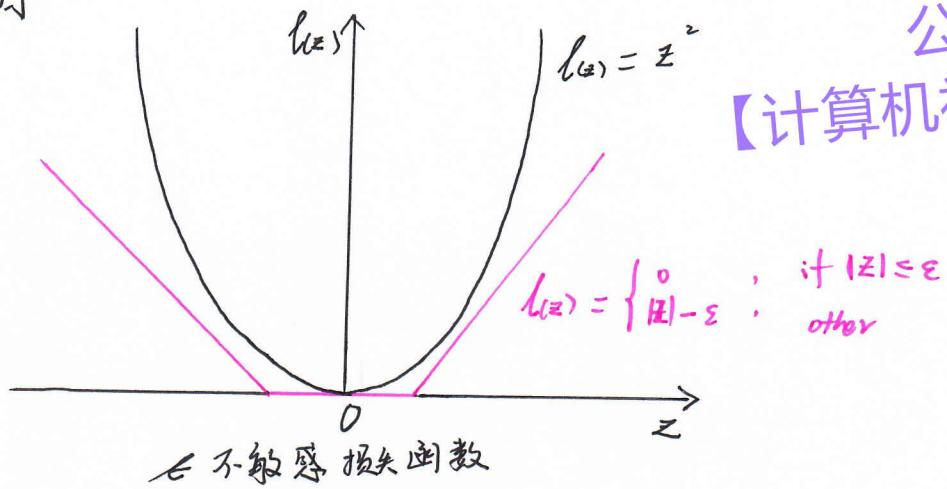
$$\xi_i \geq 0, \hat{\xi}_i \geq 0, \quad i=1, 2, \dots, m$$

第6章 支持向量机

6.5 支持向量回归

公众号

【计算机视觉联盟】



类似(6.36)引入拉格朗日乘子 $\eta_i \geq 0, \hat{\eta}_i \geq 0, \alpha_i \geq 0, \hat{\alpha}_i \geq 0$ 得(6.45)拉格朗日函数.

$$\begin{aligned} L(w, b, \alpha, \hat{\alpha}, \xi, \hat{\xi}, \eta, \hat{\eta}) = & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\xi_i + \hat{\xi}_i) - \sum_{i=1}^m \eta_i \xi_i - \sum_{i=1}^m \hat{\alpha}_i \hat{\xi}_i \\ & + \sum_{i=1}^m \alpha_i (f(x_i) - y_i - \epsilon - \xi_i) + \sum_{i=1}^m \hat{\alpha}_i (y_i - f(x_i) - \epsilon - \hat{\xi}_i) \end{aligned} \quad (6.46)$$

令 $L(w, b, \alpha, \hat{\alpha}, \xi, \hat{\xi}, \eta, \hat{\eta})$ 对 w, b, ξ_i 和 $\hat{\xi}_i$ 的偏导为零得:

$$w = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) x_i \quad (6.47)$$

$$0 = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) \quad (6.48)$$

$$C = \alpha_i + \eta_i \quad (6.49)$$

$$C = \hat{\alpha}_i + \hat{\eta}_i \quad (6.50)$$

$$\max_{\alpha, \hat{\alpha}} \sum_{i=1}^m y_i (\hat{\alpha}_i - \alpha_i) - \epsilon (\hat{\alpha}_i + \alpha_i) - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j) x_i^T x_j \quad (6.51)$$

SVR对偶问题

$$\text{s.t. } \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) = 0, \quad 0 \leq \alpha_i, \hat{\alpha}_i \leq C$$

上述过程需满足KKT条件

$$\begin{cases} \alpha_i (f(x_i) - y_i - \epsilon - \xi_i) = 0 \\ \hat{\alpha}_i (y_i - f(x_i) - \epsilon - \hat{\xi}_i) = 0 \\ \alpha_i \hat{\alpha}_i = 0, \quad \xi_i \hat{\xi}_i = 0 \\ (C - \alpha_i) \xi_i = 0, \quad (C - \hat{\alpha}_i) \hat{\xi}_i = 0 \end{cases} \quad (6.52)$$

第6章 支持向量机

6.5 支持向量回归

观察(6.52) 仅当样本 (x_i, y_i) 不落入 ε 间隔带中，相应的 α_i 和 $\hat{\alpha}_i$ 才能取非零值。

约束 $f(x_i) - y_i - \varepsilon - \xi_i = 0$ 和 $y_i - f(x_i) - \varepsilon - \hat{\xi}_i = 0$ 不能同时成立

因此 α_i 和 $\hat{\alpha}_i$ 中至少有一个为零

公众号

将(6.47)代入(6.7) $f(x) = w^T x + b$ 中，则 SVR 解形如

$$f(x) = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) x_i^T x + b \quad (6.53)$$

能使 $(\hat{\alpha}_i - \alpha_i) \neq 0$ 的样本即为 SVR 支持向量，必落在 ε 间隔带外。

SVR 支持向量仅是训练样本一部分，称具有稀疏性

由 KKT 条件(6.52)可得，每个样本 (x_i, y_i) 都有 $(C - \alpha_i)\xi_i = 0$

$$\text{且 } \alpha_i(f(x_i) - y_i - \varepsilon - \xi_i) = 0$$

于是，在得到 α_i 后，若 $0 < \alpha_i < C$ ，则必有 $\xi_i = 0$ ，进而

$$b = y_i + \varepsilon - \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) x_i^T x \quad (6.54)$$

理论上可选任意满足 $0 < \alpha_i < C$ 的样本通过式(6.54)求 b 。

实际中选取多个或所有满足条件 $0 < \alpha_i < C$ 的样本求解 b 后取平均值。

若考虑特征映射形式(6.19) $f(x) = w^T \phi(x) + b$ ，则式(6.47)升级为：

$$w = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) \phi(x_i) \quad (6.55)$$

将式(6.55)代入(6.19) 则 SVR 可表示为

$$f(x) = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) \underbrace{k(x, x_i)}_{\downarrow} + b \quad (6.56)$$

$k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ 为核函数

第6章 支持向量机

公众号

【计算机视觉联盟】

6.6 核方法

$$\text{回顾(6.24)} \quad f(x) = \sum_{i=1}^m \alpha_i y_i k(x, x_i) + b$$

$$(6.56) \quad f(x) = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) k(x, x_i) + b$$

若不考虑偏移 b , 无论 SVM 还是 SVR, 模型总能表示为 核函数 $k(x, x_i)$ 线性组合

定理 6.2 (表示定理) 令 H 为核函数 k 对应的再生核希尔伯特空间。
 $\|h\|_H$ 表示 H 空间中关于 h 的范数. 对任意单调递增函数 $\varphi: [0, \infty] \rightarrow \mathbb{R}$
 和任意非负损失函数 $\ell: \mathbb{R}^m \rightarrow [0, \infty]$ 优化问题

$$\min_{h \in H} F(h) = \varphi(\|h\|_H) + \ell(h(x_1), h(x_2), \dots, h(x_m)) \quad (6.57)$$

↓ 解可写为

$$h^*(x) = \sum_{i=1}^m \alpha_i k(x, x_i) \quad (6.58)$$

表示定理对损失函数没有限制.

对正则化项 φ 只要求单调递增, 不要求 φ 是凸函数.

优化问题 (6.57) 的最优解 $\stackrel{\text{优}}{\downarrow}$ 可表示为 核函数 $k(x, x_i)$ 的线性组合.

可通过“核化”将线性学习器拓展为非线性学习器

“核线性判别分析” (Kernelized Linear Discriminant Analysis, KLDA)

先假设某种映射中: $X \rightarrow F$ 将样本映射到一个特征空间 F . 然后在 F 中执行线性判别分析, 求:

$$h(x) = w^T \phi(x) \quad (6.59)$$

$$\text{类似(3.35)} \quad J = \frac{w^T S_b w}{w^T S_w w} \quad \text{LDA 最大化目标.}$$

即 S_b 与 S_w 为“广义瑞利商”.

↓

类间散度

KLDA 学习标

$$\max_w J(w) = \frac{w^T S_b^\phi w}{w^T S_w^\phi w} \quad (6.50)$$

类内散度

6.6 核方法

令 X_i 表示第 $i \in \{0, 1\}$ 类样本集合，其样本数为 m_i 。总样本数为 $m = m_0 + m_1$ ，第 i 类样本在特征空间 F 的均值为

$$\mu_i^\phi = \frac{1}{m_i} \sum_{x \in X_i} \phi(x) \quad (6.61)$$

两个散度矩阵分别为：

$$S_b^\phi = (\mu_0^\phi - \mu_1^\phi)(\mu_0^\phi - \mu_1^\phi)^T \quad (6.62)$$

$$S_w^\phi = \sum_{i=0}^1 \sum_{x \in X_i} (\phi(x) - \mu_i^\phi)(\phi(x) - \mu_i^\phi)^T \quad (6.63)$$

ϕ 难以具体知道，因此用核函数 $k(x, x_i) = \phi(x_i^\top) \phi(x)$ 在隐式表达。

把 $J(\omega)$ 表示为 (6.57) 核函数，令 $\alpha_i = 0$ 。由表示定理

$h(x)$ 可写为

$$h(x) = \sum_{i=1}^m \alpha_i k(x, x_i) \quad (6.64)$$

$$\omega = \sum_{i=1}^m \alpha_i \phi(x_i) \quad (6.65)$$

令 $K \in R^{m \times m}$ 为核函数 k 所对应的核矩阵。 $(K)_{ij} = k(x_i, x_j)$

$$\text{取 } \begin{cases} \hat{\mu}_0 = \frac{1}{m_0} K I_0 & (6.66) \\ \hat{\mu}_1 = \frac{1}{m_1} K I_1 & (6.67) \end{cases}$$

$$M = (\hat{\mu}_0 - \hat{\mu}_1)(\hat{\mu}_0 - \hat{\mu}_1)^T \quad (6.68)$$

$$N = K K^T - \sum_{i=0}^1 m_i \hat{\mu}_i \hat{\mu}_i^T \quad (6.69)$$

$$(6.60) 等价为: \max_{\alpha} J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T N \alpha} \quad (6.70)$$



周志华《机器学习》西瓜书 手推笔记 (v2)

第七章

《贝叶斯分类》

作者: 王博 (Kings)、Sophia
博士微信: **Kingsplus** (添加时请备注 学校/单位+专业)
Github: <https://github.com/Sophia-11/Machine-Learning-Notes>
(**荣登趋势榜**)
公众号【计算机视觉联盟】持续更新
后台回复**【西瓜书手推笔记】**可下载 pdf 打印版本



已完结待更笔记: 《深度学习-花书手推笔记》、《无人驾驶手推笔记》、《SLAM 十四讲》
请继续关注公众号【计算机视觉联盟】最新消息或 Github

7.1 贝叶斯决策论

对分类任务而言，在所有相关概率已知的理想情形下，贝叶斯决策论考虑如何基于这些概率和误判损失来选择最优的类别标记。

假设有 N 种可能类别标记即 $\mathcal{Y} = \{c_1, c_2, \dots, c_N\}$

λ_{ij} 是将一个真实标记 g_j 样本误分为 c_i 所产生的损失。

基于后概率 $P(c_i|x)$ 可获得样本 x 分类为 c_i 所产生的期望损失。

即在样本 x 上的“条件风险”

$$R(c_i|x) = \sum_{j=1}^N \lambda_{ij} P(g_j|x) \quad (7.1)$$

寻找一个判定准则 $h: X \rightarrow \mathcal{Y}$ 以最小化总体风险

$$R(h) = E_x [R(h(x)|x)] \quad (7.2)$$

贝叶斯判定准则 (Bayes decision rule)

为最小化总体风险，只需在每个样本上选择那个能使条件风险 $R(c_i|x)$ 最小的类别标记，即：

$$h^*(x) = \arg \min_{c \in \mathcal{Y}} R(c|x) \quad (7.3)$$

h^* 称为贝叶斯最优分类器。

总体风险 $R(h^*)$ 称为贝叶斯风险。

$1 - R(h^*)$ 反映了分类器所能达到的最性能。即模型精度上限。

具体而言，若目标是最小化分类错误率，则误判损失 λ_{ij} 可写为

$$\lambda_{ij} = \begin{cases} 0, & \text{if } i=j \\ 1, & \text{other} \end{cases}$$

条件风险 $R(c|x) = 1 - P(c|x)$

第7章 贝叶斯分类

7.1 贝叶斯决策论

最小化分类错误率的贝叶斯最优分类器为

$$h^*(x) = \arg \max_{c \in Y} P(c|x)$$

对每个样本 x , 选后验概率 $P(c|x)$ 最大的类别标记

机器学习所要实现的是基于有限训练样本集尽可能准确地估计出后验概率 $P(c|x)$, 主要有两种策略

(1) “判别式模型”: 给定 x , 可通过直接建模 $P(c|x)$ 来预测 c .

(2) “生成式模型”: 先对联合概率分布 $P(x, c)$ 建模, 然后再获得 $P(c|x)$.

【举例】: 决策树, BP, 支持向量机

生成式模型, 必须考虑

$$P(c|x) = \frac{P(x, c)}{P(x)} \quad (7.7)$$

$$P(c|x) = \frac{P(c) P(x|c)}{P(x)} \quad (7.8)$$

↑
“先验”
↓
“似然”
↑
归一化的“证据”因子.

※

因此, 估计 $P(c|x)$ 转化为如何基于训练数据 D 来估计先验 $P(c)$ 和似然 $P(x|c)$

类的先验概率 $P(c)$ 表达了各类样本所占比例. 根据大数定律, 训练集包含充分独立样本, $P(c)$ 可通过各类样本出现频率估计.

类的条件概率 $P(x|c)$: 由于涉及关于 x 所有属性的联合概率, 直接根据样本估计可遇到困难, 使用频率来估计不行, 因为“未被观测到”与“出现概率为零”通常是不同的.

公众号

【计算机视觉联盟】

7.2 极大似然估计

估计类条件概率记关于类别 c 的类条件概率为 $P(x|c)$, 假设 $P(x|c)$ 具有确定的形状并且被参数向量 θ_c 唯一确定, 则我们的任务就是利用训练集 D 估计参数 θ_c . 将 $P(x|c)$ 记为 $P_{(x|\theta_c)}$

概率模型的训练过程就是参数估计的过程

参数估计两种不同方案: ① 频率主义学派认为参数虽然未知, 但客观存在, 可通过优化似然函数等准则确定参数值
 ② 贝叶斯派认为参数是未观察到的随机变量, 其本身可有分布, 因此可假设服从一个先验分布, 然后基于观测到的数据来计算参数的后验分布

极大似然估计 MLE, 根据数据采样来估计概率分布

令 D_c 表示训练集 D 的第 c 类样本集合, 假设样本独立同分布.

参数 θ_c 对数据集 D_c 的似然是

$$P(D_c | \theta_c) = \prod_{x \in D_c} P(x | \theta_c) \quad (7.9)$$

对 θ_c 进行极大似然估计, 就是去寻找最大化似然 $P(D_c | \theta_c)$ 的参数值 $\hat{\theta}_c$

(7.9) 式会变成下述, 通常为对数似然 (log-likelihood)

$$\begin{aligned} LL(\theta_c) &= \log P(D_c | \theta_c) \\ &= \sum_{x \in D_c} \log P(x | \theta_c) \end{aligned} \quad (7.10)$$

此时 θ_c 的极大似然估计 $\hat{\theta}_c$ 为

$$\hat{\theta}_c = \underset{\theta_c}{\operatorname{argmax}} LL(\theta_c) \quad (7.11)$$

第7章 贝叶斯分类

公众号

【计算机视觉联盟】

7.2 极大似然估计

在连续属性下，假设概率密度函数 $P(x|c) \sim N(\mu_c, \sigma^2)$

μ_c, σ^2 极大似然为

$$\hat{\mu}_c = \frac{1}{|D_c|} \sum_{x \in D_c} x \quad (7.12)$$

$$\hat{\sigma}^2 = \frac{1}{|D_c|} \sum_{x \in D_c} (x - \hat{\mu}_c)(x - \hat{\mu}_c)^T \quad (7.13)$$

估计结果的准确性严重依赖所假设的概率分布形式是否符合潜在的真实数据分布。

7.3 朴素贝叶斯分类器

贝叶斯公式来估计后验概率 $P(c|x)$ 困难在于 $P(c|x)$ 是所有属性的联合概率，难以从有限训练样本直接估计。

朴素贝叶斯分类器采用“属性条件独立性假设”对已知类别，假设所有属性相互独立，假设每个属性独立地对分类结果发生影响。

基于属性条件独立性假设，式(7.8)重写为

$$P(c|x) = \frac{P(c) P(x|c)}{P(x)} = \frac{P(c)}{P(x)} \prod_{i=1}^d P(x_i|c) \quad (7.14)$$

d为属性数目
x在第i个属性取值

对所有类别而言 $P(c)$ 都相同，基于式(7.6)的贝叶斯判定准则有：

$$h_{nb}(x) = \arg \max_{c \in \mathcal{Y}} P(c) \prod_{i=1}^d P(x_i|c) \quad (7.15)$$

朴素贝叶斯分类器表达式

朴素贝叶斯分类器的训练过程就是基于训练集 D 来估计先验概率 $P(c)$ ，并为每个属性估计条件概率 $P(x_i|c)$

第7章 贝叶斯分类

公众号

【计算机视觉联盟】

7.3朴素贝叶斯分类器

D_c 表示训练集 D 中第 c 类集合，样本充足， c 类先验概率：

$$P(c) = \frac{|D_c|}{|D|} \quad (7.16)$$

离散属性而言， D_{c,x_i} 表示 D_c 中在第 i 个属性上取值为 x_i 的样本组成集合，则条件概率 $P(x_i|c)$ ：

$$P(x_i|c) = \frac{|D_{c,x_i}|}{|D_c|} \quad (7.17)$$

连续属性可用概率密度函数。假定 $P(x_i|c) \sim N(\mu_{c,i}, \sigma_{c,i}^2)$

$$P(x_i|c) = \frac{1}{\sqrt{2\pi} \sigma_{c,i}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma_{c,i}^2}\right) \quad (7.18)$$

西瓜数据集

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖量	好瓜
1	青	蜷	浊	凹	硬	清			
2	乌	蜷	沉	凹	硬	清			
3	乌	蜷	浊	凹	硬	清			
4	青	蜷	沉	凹	硬	清			
5	浅	蜷	浊	凹	硬	清			
6	青	稍	浊	稍	软	清			
7	浅	稍	浊	稍	软	稍			
8	乌	稍	浊	稍	硬	清			
9	乌	稍	沉	稍	硬	稍			
10	青	硬	清	平	软	清			
11	浅	硬	清	平	硬	模			
12	浅	蜷	浊	平	软	模			
13	青	稍	浊	凹	硬	稍			
14	浅	稍	沉	凹	硬	清			
15	乌	稍	浊	稍	软	模			
16	浅	蜷	浊	平	硬	稍			
17	青	蜷	沉	稍	硬	稍			

好瓜

不好瓜

第7章 贝叶斯分类

公众号

【计算机视觉联盟】

7.3 朴素贝叶斯分类器

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖量	好瓜
1	青	蜷	浊	清	凹	硬	0.697	0.460	?

估计类先验概率 $P(c)$

$$P(\text{好瓜}) = \frac{8}{17} \approx 0.471$$

$$P(\text{不是好瓜}) = \frac{9}{17} \approx 0.529$$

为每个属性估计条件概率 $P(x_i | c)$

$$P_{\text{青|是}} = P(\text{色泽=青} | \text{好瓜}) = \frac{3}{8} = 0.375$$

$$P_{\text{青|否}} = P(\text{色泽=青} | \text{不是好瓜}) = \frac{3}{9} \approx 0.333$$

$$P_{\text{蜷|是}} = P(\text{蜷} | \text{好瓜}) = \frac{5}{8} = 0.625$$

$$P_{\text{蜷|否}} = P(\text{蜷} | \text{否}) = \frac{3}{9} \approx 0.333$$

$$P_{\text{浊|是}} = P(\text{浊} | \text{好瓜}) = \frac{6}{8} = 0.75$$

$$P_{\text{浊|否}} = P(\text{浊} | \text{否}) = \frac{4}{9} \approx 0.444$$

$$P_{\text{清|是}} = P(\text{清} | \text{好瓜}) = \frac{7}{8} = 0.875$$

$$P_{\text{清|否}} = P(\text{清} | \text{否}) = \frac{2}{9} \approx 0.222$$

$$P_{\text{凹|是}} = P(\text{凹} | \text{好瓜}) = \frac{6}{8} = 0.75$$

$$P_{\text{凹|否}} = P(\text{凹} | \text{否}) = \frac{2}{9} \approx 0.222$$

$$P_{\text{硬|是}} = P(\text{硬} | \text{好瓜}) = \frac{6}{8} = 0.75$$

$$P_{\text{硬|否}} = P(\text{硬} | \text{否}) = \frac{6}{9} \approx 0.667$$

$$P_{\text{密度:0.697|是}} = P(\text{密度}=0.697 | \text{好瓜=是})$$

$$= \frac{1}{\sqrt{2\pi} \cdot 0.129} \exp\left(-\frac{(0.697 - 0.574)^2}{2 \cdot 0.129^2}\right) \approx 0.959$$

$$P_{\text{密度:0.697|否}} = P(\text{密度}=0.697 | \text{否})$$

$$= \frac{1}{\sqrt{2\pi} \cdot 0.195} \exp\left(-\frac{(0.697 - 0.476)^2}{2 \cdot 0.195^2}\right) \approx 0.103$$

$$P_{\text{含糖量:0.460|是}} = \frac{1}{\sqrt{2\pi} \cdot 0.101} \exp\left(-\frac{(0.460 - 0.279)^2}{2 \cdot 0.101^2}\right) \approx 0.788$$

$$P_{\text{含糖量:0.460|否}} = \frac{1}{\sqrt{2\pi} \cdot 0.108} \exp\left(-\frac{(0.460 - 0.154)^2}{2 \cdot 0.108^2}\right)$$

$$\approx 0.066$$

第7章 贝叶斯分类

公众号

【计算机视觉联盟】

7.3 朴素贝叶斯分类器

$$P(\text{好瓜}=\text{是}) \times P_{\text{青}}=\text{是} \times P_{\text{蜡}}=\text{是} \times P_{\text{油}}=\text{是} \times P_{\text{凹}}=\text{是} \times P_{\text{硬}}=\text{是} \times P_{\text{密度}}=\text{是} \times P_{\text{糖分}}=\text{是}$$

$$\approx 0.038$$

$$P(\text{好瓜}=\text{否}) \times P_{\text{青}}=\text{否} \times P_{\text{蜡}}=\text{否} \times P_{\text{油}}=\text{否} \times P_{\text{凹}}=\text{否} \times P_{\text{硬}}=\text{否} \times P_{\text{密度}}=\text{否} \times P_{\text{糖分}}=\text{否} \approx 6.8 \times 10^{-5}$$

由于 $0.038 > 6.8 \times 10^{-5}$ 判别为好瓜

需注意，若某个属性值在训练中没有与某个类同时出现过，则

$$P_{\text{清脆}}=\text{是} = P(\text{清脆} | \text{好瓜}) = \frac{0}{8} = 0$$

此时乘积永远是0。避免这种情况，在估计概率值时通常进行“平滑”，常用“拉普拉斯修正”。

令 N 表示训练集 D 中可能的类别数， N_i 表示第 i 个属性可能取值数。

$$(7.16) \quad P(c) = \frac{|D_c|}{|D|} \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{记为} \quad \hat{P}(c) = \frac{|D_c| + 1}{|D| + N}$$

$$(7.17) \quad P(x_i | c) = \frac{|D_{c,x_i}|}{|D_c|} \quad \left. \begin{array}{l} \\ \end{array} \right\} \quad \hat{P}(x_i | c) = \frac{|D_{c,x_i}| + 1}{|D_c| + N_i}$$

本节中 先验概率为

$$\hat{P}(\text{好瓜}=\text{是}) = \frac{8+1}{17+2} \approx 0.474 \quad \hat{P}(\text{好瓜}=\text{否}) = \frac{9+1}{17+2} \approx 0.526$$

类似

$$\hat{P}_{\text{青}}=\text{是} = \hat{P}(\text{青}=\text{是} | \text{好瓜}) = \frac{3+1}{8+3} \approx 0.364$$

$$\hat{P}_{\text{青}}=\text{否} = \frac{3+1}{9+3} \approx 0.333$$

7.4 半朴素贝叶斯分类器

人们尝试对属性条件独立性假设进行一定程度的放松。

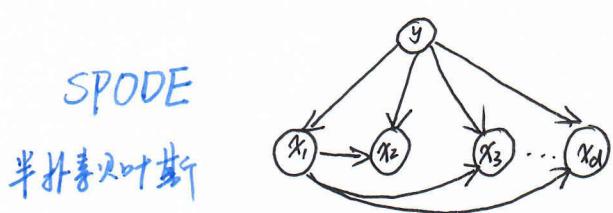
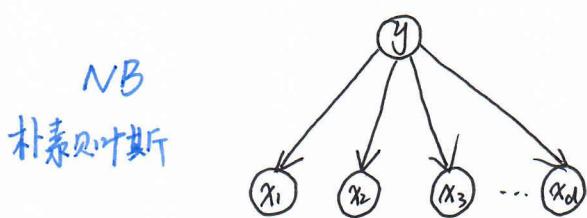
半朴素贝叶斯分类器基本想法：适当考虑一部分属性间的相互依赖信息，从而既不需进行完全联合概率计算，又不至于彻底忽略了比较强的属性依赖关系。

“独依赖估计” (One-Dependant Estimator, ODE) 是半朴素贝叶斯分类器最常用的策略，假设每个属性在类别之外最多依赖于一个其他属性，即：

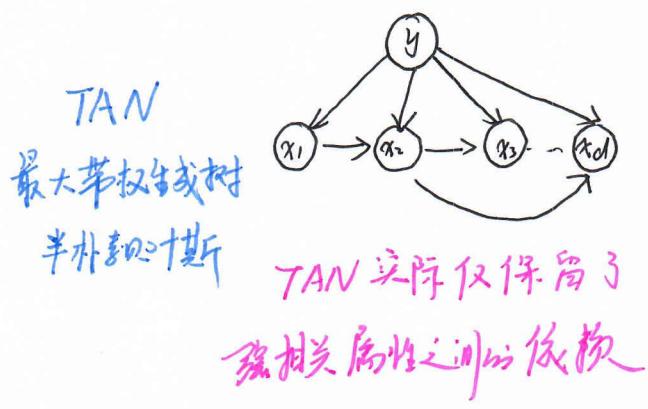
$$P(c|x) \propto P(c) \prod_{i=1}^d P(x_i|c, p_{x_i}) \quad (7.21)$$

属性 x_i : 所依赖的属性， x_i : 文属性

如何确定每个属性的文属性？不同策略的独依赖分类器又不同。



所有属性都依赖于同一个属性，称为“超父”
 x_1 是超父属性



① 计算任意两属性间条件互信息

$$I(x_i, x_j|y) = \sum_{x_i, x_j, c \in V} P(x_i, x_j|c) \log \frac{P(x_i, x_j|c)}{P(x_i|c) P(x_j|c)}$$

② 以属性为结点构建完全图，距离两结点权重为 $I(x_i, x_j|y)$

③ 构建此完全图的最大带权生成树，选根变量，将边置为

④ 加入类别结点 y ，增加从 y 到每个属性有向边

第7章 贝叶斯分类

公众号

【计算机视觉联盟】

7.4 半朴素贝叶斯分类器

AODE (Averaged One-Dependent Estimator) 尝试将每个属性作为超父构建 SPODE，然后将那些具有足够训练数据支持的 SPODE 集成为最终结果，即

$$P(c|x) \propto \sum_{i=1}^d P(c, x_i) \prod_{j=1}^d P(x_j | c, x_i) \quad (7.23)$$

$|D_{x_i}| \geq m$

D_{x_i} 是第 i 个属性取值为 x_i 样本集， m 为阈值常数

$$\hat{P}(c, x_i) = \frac{|D_{c, x_i}| + 1}{|D| + N_i} \quad (7.24)$$

$$\hat{P}(x_j | c, x_i) = \frac{|D_{c, x_i, x_j}| + 1}{|D_{c, x_i}| + N_j} \quad (7.25)$$

举例： $\hat{P}_{\text{是, 滴响}} = \hat{P}(\text{腋下=是, 高声=滴响}) = \frac{6+1}{17+3} = 0.35$

样本数

一共 3 个属性

$\hat{P}_{\text{凹|是, 滴}} = \hat{P}(\text{腋下=凹} | \text{腋下=是, 高声=滴}) = \frac{3+1}{6+3} = 0.444$

样本数

几种属性取值

AODE 元零模型选择，既能通过预计算节省预测时间，也能采取懒惰学习方法在预测时再进行计数，易于实现增量学习。

第7章 贝叶斯分类

公众号

【计算机视觉联盟】

7.5 贝叶斯网

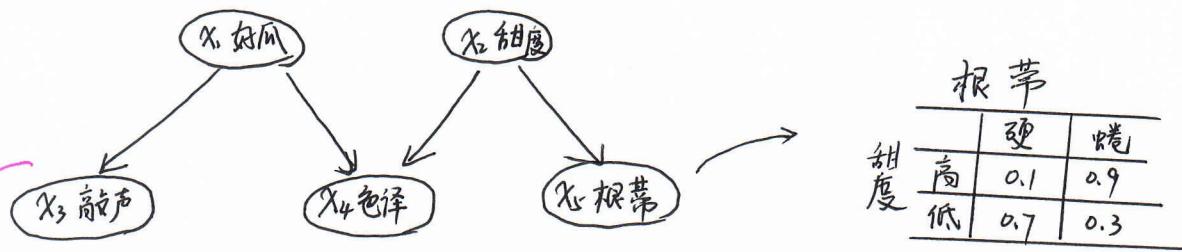
贝叶斯网 (Bayesian network) 又称“信念图”(belief network)，借助有向无环图 (Directed Acyclic Graph, DAG) 刻画属性依赖关系，并使用条件概率表 (Conditional Probability Table, CPT) 来描述属性联合概率分布。

一个贝叶斯网 B 由结构 G 和参数 θ 两部分构成， $B = \langle G, \theta \rangle$

网络结构 G 是一个有向无环图，其每个结点对应于一个属性，两属性有直接依赖关系则由一条边连接。

参数 θ 定量描述这种依赖关系。属性 x_i 在 G 的父结点为 π_i ，则 θ 包含了每个属性的条件概率表 $\theta_{x_i|\pi_i} = P_B(x_i|\pi_i)$

举例：



7.5.1 结构

贝叶斯网结构有效地表达了属性间的条件独立性。给定父结点集，贝叶斯网假设每个属性与它的非后裔属性独立。

$B = \langle G, \theta \rangle$ 将属性 x_1, x_2, \dots, x_d 的联合概率分布定义为

$$P_B(x_1, x_2, \dots, x_d) = \prod_{i=1}^d P_B(x_i | \pi_i) = \prod_{i=1}^d \theta_{x_i|\pi_i} \quad (7.26)$$

联合概率分布定义为：

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1) P(x_2) P(x_3|x_1) P(x_4|x_1, x_2) P(x_5|x_2)$$

x_3 和 x_4 在给定 x_1 取值独立 $x_3 \perp x_4 | x_1$

x_4 和 x_5 在给定 x_2 取值独立 $x_4 \perp x_5 | x_2$

第7章 贝叶斯分类

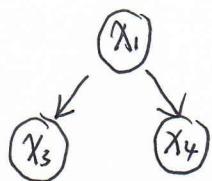
公众号

【计算机视觉联盟】

7.5 贝叶斯网

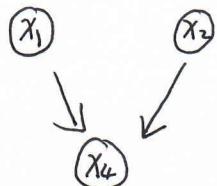
7.5.1 结构

贝叶斯网中三个变量之间的典型依赖关系



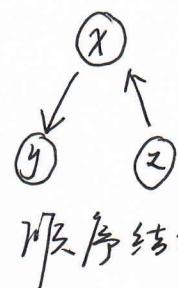
同父结构

给定 X_1 ,
 X_3, X_4 条件独立



V型结构

给定 X_4 取值, X_1, X_2 必独立
若 X_4 不变, X_1, X_2 相互独立



给定化值
 y 与 z 条件独立.

证明 ↓ “边际独立性”

$$P(x_1, x_2) = \sum_{x_4} P(x_1, x_2, x_4)$$

$$= \sum_{x_4} P(x_4 | x_1, x_2) P(x_1) P(x_2)$$

$$= P(x_1) P(x_2) \quad \text{记 } x_1 \perp\!\!\! \perp x_2$$

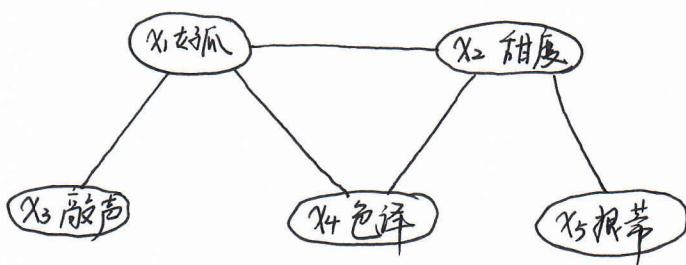
(7.27)

为了分析有向图中变量间的条件独立性, 可使用“有向完备”.

先把有向图转变为一个无向图:

- 找出有向图中的所有 V 型结构, 在 V 型结构两个父结点之间加上一条无向边
- 将所有有向边改为无向边.

由此产生的无向图称为“道德图”, 父结点相连的过程称为“道德化”.



第7章 贝叶斯分类

7.5 贝叶斯网

公众号
【计算机视觉联盟】

7.5.2 学习

贝叶斯网学习的首要任务就是根据训练数据集来找出结构最“恰当”的贝叶斯网。

“评分搜索”先定义一个评分函数，以此来评估贝叶斯网与训练数据的契合程度，基于评分函数在寻找结构最优的贝叶斯网

“最小描述长度” MDL 准则

每个贝叶斯网描述了一个在训练数据上的概率分布，自有一套编码机制能使那些经常出现的样本有更短的编码。选择综合长度最短的网。

给定训练集 $D = \{x_1, x_2, \dots, x_m\}$ ，贝叶斯网 $B = \langle G, \theta \rangle$ 在 D 上的评分函数：

$$S(B|D) = \underbrace{f(\theta)|B|}_{\text{新参数 \(\theta\) 所需字节数}} - \underbrace{LL(B|D)}_{\text{网参数个数}} \quad (7.28)$$

$$LL(B|D) = \sum_{i=1}^m \log P_B(x_i) \quad (7.29)$$

$S(B|D)$ 第1项是描述网 B 字节数 第2项是 B 对应概率分布 P_B 的字节数

$$f(\theta)=1 \text{ 得 AIC 评分函数} \quad AIC(B|D) = |B| - LL(B|D) \quad (7.30)$$

$$f(\theta)=\frac{1}{2}\log m \text{ 得 BIC 评分函数} \quad BIC(B|D) = \frac{\log m}{2}|B| - LL(B|D) \quad (7.31)$$

$f(\theta)=0$ 评分函数退化为负对数似然。

若网 $B = \langle G, \theta \rangle$ 中 G 固定，则 $S(B|D)$ 第1项为常数。参数 $\theta_{x_i|z_i}$ 可直接由数据集 D 得到：

$$\hat{\theta}_{x_i|z_i} = \hat{P}_\theta(x_i|z_i) \quad (7.32)$$

第7章 贝叶斯分类

7.6 EM算法

前面训练样本都是“完整”的，实际上并不一定都是“完整”

未观测变量学名“隐变量” 令 X 表示已观测变量集

Z 表示 隐变量集

Θ 表示模型参数

公众号

【计算机视觉联盟】

若对 Θ 做极大似然，则应最大化对数似然

$$LL(\Theta|X, Z) = \ln P(X, Z|\Theta) \quad (7.34)$$

Z 是隐变量，对 Z 计算期望，来最大化已观测数据对数“边际似然”

$$LL(\Theta|Z) = \ln P(X|\Theta) = \ln \sum_z P(X, z|\Theta) \quad (7.35)$$

EM算法常用来估计参数隐变量：

* 若 Θ 已知，可根据训练数据推断出最优隐变量 Z 的值(E步)

* 若 Z 的值已知，则可方便对参数 Θ 做极大似然估计(M步)

以初始值 Θ^0 为起点，对(7.35)进行收敛：

① 基于 Θ^t 推 Z 的期望，记 Z^t

② 基于已观测 X 和 Z^t 对 Θ 做极大似然估计，记 Θ^{t+1}

循环

EM
算法
原理

若不是取 Z 的期望，而是基于 Θ^t 计算隐变量 Z 的概率分布 $P(Z|X, \Theta^t)$

E步：以当前参数 Θ^t 推隐变量分布 $P(Z|X, \Theta^t)$ ，并计算对数似然 $LL(\Theta|X, Z)$ 关于 Z 的期望

$$Q(\Theta|\Theta^t) = E_{Z|X, \Theta^t} LL(\Theta|X, Z) \quad (7.36)$$

M步：寻找参数最大化期望似然，即

$$\Theta^{t+1} = \underset{\Theta}{\operatorname{argmax}} Q(\Theta|\Theta^t) \quad (7.37)$$



周志华《机器学习》西瓜书 手推笔记 (v2)

第八章 《集成学习》

作者: 王博 (Kings)、Sophia
博士微信: **Kingsplus** (添加时请备注 学校/单位+专业)
Github: <https://github.com/Sophia-11/Machine-Learning-Notes>
(**荣登趋势榜**)

公众号【计算机视觉联盟】持续更新
后台回复【**西瓜书手推笔记**】可下载 pdf 打印版本

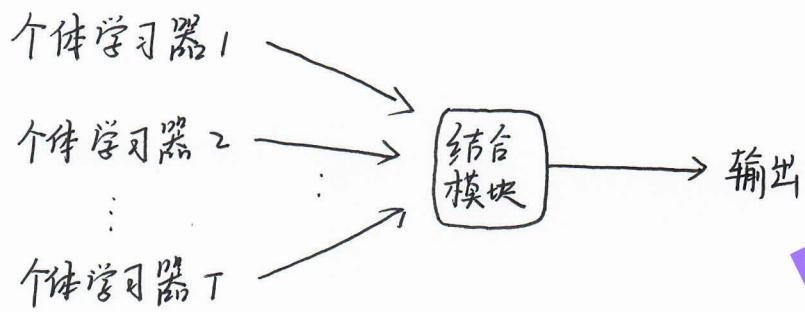


已完结待更笔记: 《深度学习-花书手推笔记》、《无人驾驶手推笔记》、《SLAM 十四讲》
请继续关注公众号【计算机视觉联盟】最新消息或 Github

第8章 集成学习

8.1 个体与集成

集成学习 (ensemble learning) 通过构建并结合多个学习器来完成学习任务，有时也被称为多分类器系统、基于委员会的学习。



同质“集成”只包含同种类型的个体学习器，同质集成中的个体学习器亦称“基学习器”，相应的学习算法称为“基学习算法”

异质“集成”由不同学习算法生成，不再有基学习法，称“组件学习器”

集成学习常获得比单一学习器显著优越的泛化性能，对“弱狗器”尤为明显

要想获得好的集成，个体学习器应“好而不同”，即个体学习器要有一定的“准确性”，并且也有差异“多样性”

以二分类问题 $y \in \{-1, +1\}$ 和函数 f 为例，假定基分类器错误率为 ε ，对每个基分类器 h_i 有：

$$P(h_i(x) \neq f(x)) = \varepsilon \quad (8.1)$$

假设集成通过投票对 T 个基分类器判断，则超过半数为正确

$$H(x) = \text{sign}\left(\sum_{i=1}^T h_i(x)\right) \quad (8.2)$$

集成错误率为：

$$P(H(x) \neq f(x)) = \sum_{k=0}^{\lfloor T/2 \rfloor} \binom{T}{k} (1-\varepsilon)^k \varepsilon^{T-k} \leq \exp(-\frac{1}{2} T(1-2\varepsilon)^2) \quad (8.3)$$

随集成个体分类器数目 T 增大，集成的错误率将指数级下降，最终趋于零。

第8章 集成学习

集成学习分两大类：①个体学习器存在强依赖关系，必须串行生成序列

化方法。如：Boosting

②不存在强依赖关系，可同时生成的并行化方法。
如 Bagging 和随机森林

8.2 Boosting

Boosting 工作机制：先从初始训练集中训练出一个基学习器，再根据基学习器表现对训练样本分布调整，使先前错的样本后续得到更大关注，基于调整后的样本训练下一个基学习器，反复直到达到指定值 T，最终将 T 个学习器加权结合。

Boosting 族算法最著名的是 AdaBoost。比较容易理解的是基“加性模型”即基学习器线性组合

$$H(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (8.4)$$

最小化指数损失函数

$$L_{\text{exp}}(H|P) = E_{x \sim P} [e^{-f(x)H(x)}] \quad (8.5)$$

若 $H(x)$ (8.4) 能令指数损失函数最小化，考虑 (8.5) 对 (8.4) 求偏导

$$\frac{\partial L_{\text{exp}}(H|P)}{\partial H(x)} = -e^{-H(x)} P(f(x)=1|x) + e^{H(x)} P(f(x)=-1|x) \quad (8.6)$$

$$\Downarrow \xi = 0$$

$$H(x) = \frac{1}{2} \ln \frac{P(f(x)=1|x)}{P(f(x)=-1|x)} \quad (8.7)$$

$$\text{sign}(H(x)) = \text{sign}\left(\frac{1}{2} \ln \frac{P(f(x)=1|x)}{P(f(x)=-1|x)}\right) = \begin{cases} 1, & P(f(x)=1|x) > P(f(x)=-1|x) \\ -1, & P(f(x)=1|x) < P(f(x)=-1|x) \end{cases} = \arg \max_{y \in \{-1, 1\}} P(f(x)=y|x) \quad (8.8)$$

第8章 集成学习

8.2 Boosting

在AdaBoost算法中，第一个基分类器 h_1 是通过直接将基学习算法用于初步数据分布而得；此后迭代生成 h_t 和 α_t 。当基分类器 h_t 基于分布 D_t 产生后，权重 α_t 应使 $\alpha_t h_t$ 最小化指数损失函数

$$\begin{aligned}
 L_{\text{exp}}(\alpha_t h_t | D_t) &= E_{x \sim D_t} [e^{-f(x)\alpha_t h_t(x)}] \\
 &= E_{x \sim D_t} [e^{-\alpha_t I(f(x) = h_t(x))} + e^{\alpha_t I(f(x) \neq h_t(x))}] \\
 &= e^{-\alpha_t} P_{x \sim D_t}(f(x) = h_t(x)) + e^{\alpha_t} P_{x \sim D_t}(f(x) \neq h_t(x)) \\
 &= e^{-\alpha_t}(1 - \varepsilon_t) + e^{\alpha_t} \varepsilon_t \quad (8.9) \\
 &\downarrow \\
 &\varepsilon_t = P_{x \sim D_t}(h_t(x) \neq f(x))
 \end{aligned}$$

考虑指数损失函数的导数

$$\frac{\partial L_{\text{exp}}(\alpha_t h_t | D_t)}{\partial \alpha_t} = -e^{-\alpha_t}(1 - \varepsilon_t) + e^{\alpha_t} \varepsilon_t \quad (8.10)$$

$$\Downarrow = 0$$

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right) \quad (8.11)$$

AdaBoost 算法在获得 H_{t-1} 之后样本分布 D_t 进行调整，使下一轮基学习器 h_t 能纠正一些 H_{t-1} 错误，理想是纠正所有错误即最小化。

$$\begin{aligned}
 L_{\text{exp}}(H_{t-1} + h_t | D) &= E_{x \sim D} [e^{-f(x)(H_{t-1}(x) + h_t(x))}] \\
 &= E_{x \sim D} [e^{-f(x)H_{t-1}(x)} e^{-f(x)h_t(x)}] \quad (8.12)
 \end{aligned}$$

$$f^2(x) = h_t^2(x) = 1 \quad \Downarrow \text{使用 } e^{-f(x)h_t(x)} \text{ 泰勒展开}$$

$$\begin{aligned}
 L_{\text{exp}}(H_{t-1} + h_t | D) &\approx E_{x \sim D} [e^{-f(x)H_{t-1}(x)} \left(1 - f(x)h_t(x) + \frac{f^2(x)h_t^2(x)}{2} \right)] \\
 &= E_{x \sim D} [e^{-f(x)H_{t-1}(x)} \left(1 - f(x)h_t(x) + \frac{1}{2} \right)] \quad (8.13)
 \end{aligned}$$

第8章 集成学习

8.2 Boosting

理想的基学习器 $h_t(x) = \underset{h}{\operatorname{argmin}} E_{\text{exp}}(H_{t-1} + h | D)$

$$= \underset{h}{\operatorname{argmin}} E_{\text{exp}} \left[e^{-f(x) H_{t-1}(x)} (1 - f(x) h(x) + \frac{1}{2}) \right]$$

$$= \underset{h}{\operatorname{argmin}} E_{\text{exp}} \left[e^{-f(x) H_{t-1}(x)} f(x) h(x) \right]$$

$$= \underset{h}{\operatorname{argmax}} E_{\text{exp}} \left[\frac{e^{-f(x) H_{t-1}(x)}}{E_{\text{exp}} [e^{-f(x) H_{t-1}(x)}]} f(x) h(x) \right] \quad (8.14)$$

这一项是常数

令 D_t 表示一个分布

$$D_t(x) = \frac{e^{-f(x) H_{t-1}(x)}}{E_{\text{exp}} [e^{-f(x) H_{t-1}(x)}]} \quad (8.15)$$

根据数学期望，(8.14), (8.15) 两式等价于：

$$\begin{aligned} h_t(x) &= \underset{h}{\operatorname{argmax}} E_{\text{exp}} \left[\frac{e^{-f(x) H_{t-1}(x)}}{E_{\text{exp}} [e^{-f(x) H_{t-1}(x)}]} f(x) h(x) \right] \\ &= \underset{h}{\operatorname{argmax}} E_{\text{exp}_{D_t}} [f(x) h(x)] \end{aligned} \quad (8.16)$$

由 $f(x), h(x) \in \{-1, +1\}$ 有

$$f(x) h(x) = 1 - 2 \mathbb{I}(f(x) \neq h(x)) \quad (8.17)$$

理想的基学习器

$$h_t(x) = \underset{h}{\operatorname{argmin}} E_{\text{exp}_{D_t}} [\mathbb{I}(f(x) \neq h(x))] \quad (8.18)$$

理想的 h_t 将在分布 D_t 下最小化分类误差。

$$D_t \text{ 与 } D_{t+1} \text{ 关系: } D_{t+1}(x) = \frac{D(x) e^{-f(x) H_{t-1}(x)}}{E_{\text{exp}} [e^{-f(x) H_{t-1}(x)}]} = \frac{D(x) e^{-f(x) H_{t-1}(x)}}{E_{\text{exp}} [e^{-f(x) H_t(x)}]} e^{-f(x) \alpha_t + h_t(x)}$$

$$\text{样本分布更新公式} = D_t(x) \cdot e^{-f(x) \alpha_t + h_t(x)} \frac{E_{\text{exp}} [e^{-f(x) H_{t-1}(x)}]}{E_{\text{exp}} [e^{-f(x) H_t(x)}]} \quad (8.19)$$

公众号
【计算机视觉联盟】

看清楚

$D_t(x), D(x)$

替换

8.3 Bagging 与随机森林

个体学习器不存在强依赖关系，可同时生成的并行化方法

欲得到泛化性能强的集成，个体学习器应尽可能有较大差异。一个数据集，可产生若干子集，每个子集训练出一个基学习器，使用相互重叠的采样子集。

8.3.1 Bagging

基于自助采样法，给定包含 m 个样本的数据集，使得下次采样该样本仍能被选中。经过 m 个随机采样，得到 m 个样本的采样集。（取出再放回重采，比如100个球，先取1个放采样集中，然后再放回，再从100个取1个放采样集）

初始训练集中有的样本在采样集中多次出现，有的从未出现。
初始训练集中约有63.2%样本出现在采样集中。

Bagging 基本流程：

可采样出 T 个含 m 个训练样本的采样集，然后基于每个采样集训练出一个基学习器，再将这些学习器结合。

结合时 $\left\{ \begin{array}{l} \text{对分类任务使用简单投票法，集取一样随机取一个} \\ \text{对回归任务使用简单平均法} \end{array} \right.$

Bagging 的复杂度大致为 $T(O(m) + O(s))$

\uparrow $O(s)$ 很小
 T 不大 \downarrow
 假设基学习器的计算复杂度为 $O(m)$

Bagging 集成与基学习算法的复杂度同阶

标准 AdaBoost 只适用于二分类任务

Bagging 能不修改地用于多类、回归任务

第8章 集成学习

公众号

【计算机视觉联盟】

8.3 Bagging 与随机森林

8.3.1 Bagging

自助采样过程给 Bagging 带来的优点：由于每个基学习器只使用了初始训练集中约 63.2% 的样本，剩下约 36.8% 的样本可用作验证集来对泛化性能进行“包外估计”

D_t 表示 h_t 实际使用训练样本集。

$H^{oob}(x)$ 表示对样本 x 的包外预测，仅针对未使用的部分

$$H^{oob}(x) = \arg \max_{y \in Y} \sum_{t=1}^T I(h_t(x) = y) \cdot I(x \notin D_t) \quad (8.20)$$

Bagging 泛化误差

包外估计为

$$\epsilon^{oob} = \frac{1}{|D|} \sum_{(x,y) \in D} I(H^{oob}(x) \neq y) \quad (8.21)$$

包外估计 \Rightarrow
 { 基学习器是决策树，可辅助剪枝，或辅助对零训练样本结点处理
 基学习器是神经网络，辅助早期停止以减小过拟合风险。}

Bagging 主要关注降低方差，它在不剪枝决策树、神经网络等易受样本扰动的学习器上效用明显

8.3.2 随机森林 (Random Forest, RF)

随机森林是 Bagging 一个扩展变体。

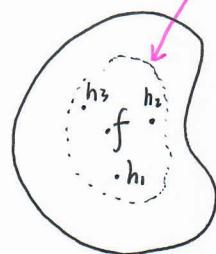
RF 以决策树为基学习器构建 Bagging 集成

在决策树训练过程中引入了随机属性选择

- {① 传统决策树在选择划分属性时是在当前结点的属性集合选一个最优属性。
② RF 中，对某决策树的每个结点，先从该结点属性集合随机选择一个包含 K 个属性的子集，然后再从子集中选择最优属性划分。
③ 一般推荐 $K = \log_2 d$ ④ $K=1$ ，随机选择一个属性用于划分
⑤ 一般推荐 $K = \sqrt{d}$

8.4 结合策略

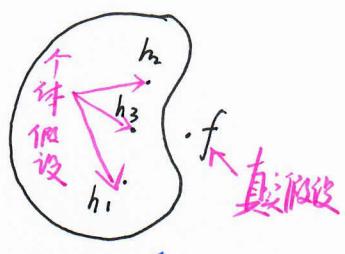
- 学习器结合带来的三个好处
- ① 从统计方面，单学习器可能因误选而导致泛化性能不佳，结合多个学习器会减小这一风险
 - ② 从计算方面，多次运行之后结合，可降低陷入编程局部极小点风险
 - ③ 从表示方面，结合多个学习器，相应的假设空间有所扩大，有可能学到更好的近似。



① 统计方面



② 计算方面



③ 表示原因

T个基学习器 $\{h_1, h_2, \dots, h_T\}$

8.4.1 平均法

对数值型输出 $h_i(x) \in R$, 最常见的结合策略是使用平均法 (averaging)

· 简单平均法

$$H(x) = \frac{1}{T} \sum_{i=1}^T h_i(x) \quad (8.22)$$

· 加权平均法

$$H(x) = \sum_{i=1}^T w_i h_i(x) \quad (8.23)$$

$$w_i \geq 0, \sum_{i=1}^T w_i = 1$$

第8章 集成学习

公众号

【计算机视觉联盟】

8.4 结合策略

8.4.2 投票法

学习器 h_i 将从类别标记集合 $\{c_1, c_2 \dots c_N\}$ 中被训练出一个标记。

将 h_i 在样本 x 上的预测输出表示为 N 维向量 $(h_i^1(x), h_i^2(x), \dots, h_i^N(x))$.

$h_i^j(x)$ 是 h_i 在类别 c_j 上的输出。

· 绝对多数投票法 (majority voting)

$$H(x) = \begin{cases} c_j & , \text{ if } \sum_{i=1}^T h_i^j(x) > 0.5 \sum_{k=1}^N \sum_{i=1}^T h_i^k(x) ; \\ \text{reject} & , \text{ otherwise.} \end{cases} \quad (8.24)$$

标记得票过半，被识别为该标记。否则拒绝识别

· 相对多数投票法 (plurality voting)

$$H(x) = \operatorname{argmax}_j \sum_{i=1}^T h_i^j(x) \quad (8.25)$$

得票最多的为标记，若相同票数，随机选取一个

· 加权投票法 (weighted voting)

$$H(x) = \operatorname{argmax}_j \sum_{i=1}^T w_i h_i^j(x) \quad (8.26)$$

不同类型个体学习器可能产生不同类型的 $h_i^j(x)$ 的值，常见的有：

* 类标记： $h_i^j(x) \in \{0, 1\}$ ， h_i 将样本 x 被判为 c_j 的取值为 1，或为 0。
称为“硬投票”

* 类概率： $h_i^j(x) \in [0, 1]$ ，相当于后验概率 $P(c_j|x)$ 的一个估计。
称为“软投票”

8.4 结合策略

8.4.3 学习法

当训练数据很多时，一种更为强大的结合策略是使用“学习法”，即通过另一个学习器进行结合。

Stacking 是典型代表

将个体学习器称为初级学习器，用于结合的学习器称为次级学习器或元学习器。

① *Stacking* 先从初始数据集训练出初级学习器

“生成”一个新数据集用于训练次级学习器

② 初级学习器的输出被当作样例输入特征

而初始样本的标记仍被当作样例标记

③ 次级训练集是利用初级学习器产生的

一般使用交叉验证或留一法这样的方式，用训练初级学习器未使用
的样本来产生次级学习器的训练样本

将初级学习器的输出置概率作为次级学习器的输入属性。

用多响应线性回归 (Multi-response Linear Regression, MLR) 作为次

级学习算法效果较好，在MLR中使用不同的属性集更佳。

贝叶斯模型平均 (Bayes Model Averaging, BMA) 基于后验概率来为
不同模型赋予权重，可视为加权平均法的一种特殊实现。

Stacking 通常优于BMA，鲁棒性好，对近似误差敏感。

8.5 多样性

8.5.1 误差一分歧分解

欲构建泛化能力强的集成，个体学习器互“如何不同”

假定用个体学习器 h_1, h_2, \dots, h_T 通过加权平均法结合产生的集成来完成回归学习任务 $f: R^d \rightarrow R$.

对示例 x , 定义学习器 h_i 的“分歧”为

$$A(h_i|x) = (h_i(x) - H(x))^2 \quad (8.27)$$

集成的分歧是：

$$\bar{A}(h|x) = \sum_{i=1}^T w_i A(h_i|x) \quad (8.28)$$

反映了个体学习器的多样性

$$= \sum_{i=1}^T w_i (h_i(x) - H(x))^2$$

个体学习器 h_i 和集成 H 的平方误差：

$$E(h_i|x) = (f(x) - h_i(x))^2$$

$$E(H|x) = (f(x) - H(x))^2$$

令 $\bar{E}(h|x) = \sum_{i=1}^T w_i \cdot E(h_i|x)$ 表示个体学习器误差的加权均值

$$\begin{aligned} \bar{A}(h|x) &= \sum_{i=1}^T w_i E(h_i|x) - E(H|x) \\ &= \bar{E}(h|x) - E(H|x) \end{aligned} \quad (8.31)$$

令 $p(x)$ 为样本概率密度

$$\sum_{i=1}^T w_i \int A(h_i|x) p(x) dx = \sum_{i=1}^T w_i \int E(h_i|x) p(x) dx - \int E(H|x) p(x) dx \quad (8.32)$$

个体学习器 h_i 在全样本上的泛化误差 $E_i = \int E(h_i|x) p(x) dx$ (8.33)

分歧项 $A_i = \int A(h_i|x) p(x) dx$ (8.34)

集成的泛化误差 $E = \int E(H|x) p(x) dx$ (8.35)

第8章 集成学习

公众号

【计算机视觉联盟】

8.5 多样性

(8.33 ~ 8.35) → 代入 (8.32)

令 $\bar{E} = \sum_{i=1}^T w_i E_i$ 表示个体学习器泛化的加权均值

$\bar{A} = \sum_{i=1}^T w_i A_i$ 表示个体学习器的加权分歧值，有：

$$E = \bar{E} - \bar{A} \quad (8.36)$$

“误差一分支分解”

8.5.2 多样性度量

度量集成个体分类器的多样性，即估算个体学习器多样化程度。

比如考虑个体分类器的 两两相似 / 不相似性。

给定数据集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ，对二分类任务，

$y_i \in \{-1, +1\}$ 分类器 h_i 与 h_j 互斥] 结果取表

		$h_i = +1$	$h_i = -1$
$h_j = +1$	a	c	
	b	d	

a 表示 h_i, h_j 的预测为正类的样本数目。 $a+b+c+d = m$.

· 不合度量 (disagreement measure)

$$dis_{ij} = \frac{b+c}{m} \quad (8.37)$$

值域 $[0, 1]$ 值越大 多样性越大

· 相关系数

$$\rho_{ij} = \frac{ad - bc}{\sqrt{(a+b)(a+c)(c+d)(b+d)}} \quad (8.38)$$

值域 $[-1, 1]$ ， h_i, h_j 元素为 0，正相关为正 负相关为负

· Q-统计量 (Q-statistic)

$$Q_{ij} = \frac{ad - bc}{ad + bc} \quad (8.39)$$

Q_{ij} 与 ρ_{ij} 等号相同， $|Q_{ij}| \leq |\rho_{ij}|$

8.5 多样性

 κ -统计量 (κ -statistic)

$$\kappa = \frac{P_1 - P_2}{1 - P_2}$$

(8.40)

P_1 = $\frac{a+d}{m}$ (8.41)
两分类器一致概率

P_2 = $\frac{(a+b)(a+c) + (c+d)(b+d)}{m^2}$ (8.42)
两分类器偶然一致概率

完全一致 $\kappa = 1$ κ 通常为非负值偶然一致 $\kappa = 0$

8.5.3 多样性增强

如何有效地增大多样性大的个体学习器，主要是对数据样本、输入属性、输出表示、算法参数进行扰动。

- ① 数据样本扰动
 - 通常基于采样法，Bagging 中自助采样
 - AdaBoost 使用序列采样
 - 对“不稳定性学习器”：决策树、神经网络 有效
 - 对“稳定”没有效：线性学习器、支持向量机、朴素贝叶斯、k 近邻学习器
- ② 输入属性扰动
 - 随机子空间从初始属性集中抽取若干个属性子集。
 - 再基于每个属性子集训练一个基础器。
 - 若属性很少，不适用
- ③ 输出表示扰动
 - 对类标记变动，如“翻转法”随机改变一些样本标记。
 - 对输出表示转化，如“输出限制法”将分类输出转化为回归输出后构建个体学习器。
 - 原化劣评价，ECOC 该一分为二
- ④ 算法参数扰动
 - 神经网络：神经元数、权值。
 - “负相关法”显示正则化项
 - 决策树 属性选择机制



周志华《机器学习》西瓜书 手推笔记 (v2)

第九章 《聚类》

作者: 王博 (Kings)、Sophia
博士微信: **Kingsplus** (添加时请备注 学校/单位+专业)
Github: <https://github.com/Sophia-11/Machine-Learning-Notes>
(**荣登趋势榜**)
公众号【计算机视觉联盟】持续更新
后台回复**【西瓜书手推笔记】**可下载 pdf 打印版本



已完结待更笔记: 《深度学习-花书手推笔记》、《无人驾驶手推笔记》、《SLAM 十四讲》
请继续关注公众号【计算机视觉联盟】最新消息或 Github

第9章 聚类

9.1 聚类任务

聚类试图将数据集中的样本划分为若干个通常是不相交的子集，每个子集称为“簇”。

假定样本集 $D = \{x_1, x_2, \dots, x_m\}$ 包含 m 个无标记样本。
 每个样本 $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ 是一个 n 维向量。
 聚类将样本集 D 划分为 k 个不相交的簇 $\{C_l \mid l = 1, 2, \dots, k\}$
 其中 $C_i \cap_{l \neq i} C_l = \emptyset$ 且 $D = \bigcup_{i=1}^k C_i$ 。
 用 $\lambda_j \in \{1, 2, \dots, k\}$ 表示“簇”标记。聚类结果可用包含 m 个元素的簇标记向量
 $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ 表示

9.2 性能度量

亦称聚类“有效性指标” (Validity Index)

聚类结果的“簇内相似度”高且“簇间相似度”低。

性能度量大致两类 $\begin{cases} ① \text{与“参考模型”比，“外部指标”} \\ ② \text{直接考虑结果，“内部指标”} \end{cases}$

对数据集 $D = \{x_1, x_2, \dots, x_m\}$

聚类给出的簇划分 $C = \{C_1, C_2, \dots, C_k\}$

$\lambda \sim$ 簇标记向量

参考模型给出的簇划分 $C^* = \{C_1^*, C_2^*, \dots, C_s^*\}$

λ^*

$$a = |SSI|, SSI = \{(x_i, x_j) \mid \lambda_i = \lambda_j, \lambda_i^* = \lambda_j^*, i < j\} \quad (9.1)$$

两两配对

$$b = |SDI|, SDI = \{(x_i, x_j) \mid \lambda_i = \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\} \quad (9.2)$$

定义

$$c = |DSI|, DSI = \{(x_i, x_j) \mid \lambda_i \neq \lambda_j, \lambda_i^* = \lambda_j^*, i < j\} \quad (9.3)$$

$$d = |DDI|, DDI = \{(x_i, x_j) \mid \lambda_i \neq \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\} \quad (9.4)$$

$$a+b+c+d = C_m^2 = \frac{m(m-1)}{2}$$

9.2 性能度量

聚类性能度量外部指标：

- Jaccard 系数 (Jaccard Coefficient, J_C)

$$[0,1] \quad J_C = \frac{a}{a+b+c} \quad (9.5)$$

- FM 指数 (Fowlkes and Mallows Index, FMI)

$$FMI = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}} \quad (9.6)$$

- 越大越好
• Rand 指数 (Rand Index, RI)

$$RI = \frac{2ac+bd}{m(m-1)} \quad (9.7)$$

考虑聚类结果的簇划分 $C = \{C_1, C_2, \dots, C_k\}$ 定义

样本间平均距离 $\text{avg}(C) = \frac{2}{|C|(|C|-1)} \sum_{1 \leq i < j \leq |C|} \text{dist}(x_i, x_j) \quad (9.8)$

样本间最近距离 $\text{diam}(C) = \max_{1 \leq i \leq j \leq |C|} \text{dist}(x_i, x_j) \quad (9.9)$
计算两样本之间距离

最近距离 $\text{dim}(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} \text{dist}(x_i, x_j) \quad (9.10)$

簇 C_i 与簇 C_j 中心点距离 $d_{cen}(C_i, C_j) = \text{dist}(\gamma_i^k, \gamma_j^l)$
 $\gamma_i^k = \frac{1}{|C_i|} \sum_{1 \leq i \leq |C_i|} x_i$
簇中心点

聚类性能度量的内部指标

- DB 指数 (Davies-Bouldin Index, DBI)

$$DBI = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left(\frac{\text{avg}(C_i) + \text{avg}(C_j)}{d_{cen}(\gamma_i^k, \gamma_j^l)} \right) \quad (9.12)$$

越小越好

- Dunn 指数 (Dunn Index, DI)

$$DI = \min_{1 \leq i \leq k} \left\{ \min_{j \neq i} \left(\frac{\text{dmin}(C_i, C_j)}{\max_{1 \leq i \leq k} \text{diam}(C_i)} \right) \right\} \quad (9.13)$$

越大越好

第9章 聚类

公众号

【计算机视觉联盟】

9.3 距离计算

函数 $dist(\cdot, \cdot)$ “距离度量” 基本性质	非负性	$dist(x_i, x_j) \geq 0$	(9.14)
	同一性	$dist(x_i, x_j) = 0$ 当且仅当 $x_i = x_j$	(9.15)
	对称性	$dist(x_i, x_j) = dist(x_j, x_i)$	(9.16)
	直递性	$dist(x_i, x_j) \leq dist(x_i, x_k) + dist(x_k, x_j)$	(9.17)

给定样本 $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ 最常用的是“闵可夫斯基距离”
 $x_j = (x_{j1}, x_{j2}, \dots, x_{jn})$

$$dist_{mk}(x_i, x_j) = \left(\sum_{u=1}^n |x_{iu} - x_{ju}|^p \right)^{\frac{1}{p}}, p \geq 1 \text{ 且是所有}$$
(9.18)

$p=2$ 时 欧氏距离 (Euclidean Distance)

$$dist_{ed}(x_i, x_j) = \|x_i - x_j\|_2 = \sqrt{\sum_{u=1}^n |x_{iu} - x_{ju}|^2}$$
(9.19)

$p=1$ 时 曼哈顿距离 (Manhattan distance)

$$dist_{man}(x_i, x_j) = \|x_i - x_j\|_1 = \sum_{u=1}^n |x_{iu} - x_{ju}|.$$
(9.20)

属性划分

“连续属性”
“离散属性”

属性划分

“有序属性”	1, 2, 3	闵可夫斯基距离可用
“无序属性”	飞机, 火车, 轮船	闵可夫斯基距离不可用

对无序属性采用 VDM.

令 m_{ua} 表示属性 u 上取值为 a 的样本数.

$m_{u,a,i}$ 表第 i 个样本簇中在属性 u 上取值为 a 的样本数.

属性 u 上两个离散值 a 与 b 之间的 VDM 距离为:

$$VDM_p(a, b) = \left| \frac{m_{u,a,i}}{m_{u,a}} - \frac{m_{u,b,i}}{m_{u,b}} \right|^p$$
(9.21)

9.3 距离计算

将闵可夫斯基距离和VDM结合即可处理混合属性

n_c 个有序属性， $n-n_c$ 个无序属性。即：

$$\text{Minko VDM}_p(x_i, x_j) = \left(\sum_{u=1}^{n_c} |x_{iu} - x_{ju}|^p + \sum_{u=n_c+1}^n \text{VDM}_p(x_{iu}, x_{ju}) \right)^{\frac{1}{p}} \quad (9.22)$$

样本权重不同，“加权距离”

加权闵可夫斯基距离：

$$\text{dist}_{wmk}(x_i, x_j) = (w_1 \cdot |x_{i1} - x_{j1}|^p + \dots + w_n \cdot |x_{in} - x_{jn}|^p)^{\frac{1}{p}} \quad (9.23)$$

\downarrow
w_i 为权重 $\sum_{i=1}^n w_i = 1$

9.4 原型聚类

“基于原型的聚类”，通过一组原型刻画。

9.4.1 k均值算法

样本集 $D = \{x_1, x_2, \dots, x_m\}$ ， k -means 算法针对聚类所得簇划分

$C = \{C_1, C_2, \dots, C_k\}$ 最小化平方误差

$$E = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|_2^2 \quad (9.24)$$

\downarrow
 $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$ 均值向量.

越小，内部相似度越高

这是一个NP问题，采用贪心策略，通过迭代化近似求解

算法：① 假设簇数 $k=3$ ，随机选3个样本做为中心 μ_1, μ_2, μ_3

② 对每一个样本，计算与 μ_1, μ_2, μ_3 距离 分类

$$\begin{aligned} C_1 &= \{x_5, x_6, \dots\} \\ C_2 &= \{x_1, x_2, \dots\} \\ C_3 &= \{x_8, x_9, \dots\} \end{aligned}$$

③ 对 C_1, C_2, C_3 分别求新的均值向量 μ'_1, μ'_2, μ'_3

不断重复迭代，得到最终划分。

第9章 聚类

公众号

【计算机视觉联盟】

9.4 原型聚类

9.4.2 学习向量量化

Learning Vector Quantization, LVQ.

假设样本有类别标记，学习过程利用样本的监督信息来辅助聚类

给定样本集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$

每个样本 x_j 由 n 个属性描述的特征向量 $(x_{j1}, x_{j2}, \dots, x_{jn})$, $y_j \in Y$ 是 x_j 的类别标记
学得一组 n 维原型向量 $\{P_1, P_2, \dots, P_q\}$, 类标记 $t_i \in Y$

LVQ 关键在于如何更新原型向量

对样本 x_j , 若最近的原型向量 P_i^* 与 x_j 的类别标记相同, 则令 P_i^* 向 x_j 方向靠近。

此时新原型向量为

$$P' = P_i^* + \eta \cdot (x_j - P_i^*) \quad (9.25)$$

P' 与 x_j 之间距离为

$$\begin{aligned} \|P' - x_j\|_2 &= \|P_i^* + \eta \cdot (x_j - P_i^*) - x_j\|_2 \\ &= (1-\eta) \cdot \|P_i^* - x_j\|_2 \end{aligned} \quad (9.26)$$

类似的, 若 P_i^* 与 x_j 的类别标记不同, 则更新后的原型向量与 x_j 之间的距离增大为 $(1+\eta) \|P_i^* - x_j\|_2$, 从而更远离 x_j .

学得一组原型向量 $\{P_1, P_2, \dots, P_q\}$ 后, 可实现对样本空间 X 的簇划分.

任意样本 x , 划入最近簇中:

每个 P_i 定义了与之相关的区域 R_i . 区域中样本与 P_i 距离不大于与其它 P_i' 的距离

$$R_i = \{x \in X \mid \|x - P_i\|_2 \leq \|x - P_i'\|_2, i' \neq i\} \quad (9.27)$$

形成了对样本空间 X 的簇划分 $\{R_1, R_2, \dots, R_q\}$ 称为 Voronoi 划分

第9章 聚类

公众号

【计算机视觉联盟】

9.4 原型聚类

9.4.3 高斯混合聚类

Mixture of Gaussian 采用概率模型

对 n 维样本空间 X 中的随机向量 x , 若 x 服从高斯分布, 概率密度函数为:

$$P(x) = \frac{1}{\sqrt{2\pi} |\Sigma|} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (9.28)$$

Σ 是 $n \times n$ 协方差矩阵 记为 $P(x|\mu, \Sigma)$

定义高斯混合分布

$$P_m(x) = \sum_{i=1}^k \alpha_i \cdot P(x|\mu_i, \Sigma_i) \quad (9.29)$$

$\alpha_i > 0$ 混合系数 $\sum_{i=1}^k \alpha_i = 1$

样本生成过程:

根据 $\alpha_1, \alpha_2, \dots, \alpha_k$ 定义的先验分布选择高斯混合成分. i 为第 i 个成员概率
根据概率密度进行采样, 生成相应的样本.

生成训练集 $D = \{x_1, x_2, \dots, x_m\}$

令随机变量 $z_j \in \{1, 2, \dots, k\}$ 表生成样本 x_j 的高斯混合成分.

z_j 的先验概率 $P(z_j=i)$ 对应于 α_i ($i=1, 2, \dots, k$)

z_j 的后验分布对应于

$$\begin{aligned} P_m(z_j=i|x_j) &= \frac{P(z_j=i) \cdot P_m(x_j|z_j=i)}{P_m(x_j)} \\ &= \frac{\alpha_i \cdot P(x_j|\mu_i, \Sigma_i)}{\sum_{l=1}^k \alpha_l \cdot P(x_j|\mu_l, \Sigma_l)} \quad (9.30) \end{aligned}$$

记为 γ_{ji} ($i=1, 2, \dots, k$)

第9章 聚类

公众号

【计算机视觉联盟】

9.4 原形聚类

9.4.3 高斯混合聚类

若高斯混合分布 $P_m(x) = \sum_{i=1}^k \alpha_i \cdot p(x|\mu_i, \Sigma_i)$ 已知，高斯混合聚类把样本集 D 划分为 k 个簇， $C = \{C_1, C_2, \dots, C_k\}$

每个样本 x_j 的簇标记 λ_j 如下：

$$\lambda_j = \arg \max_{i \in \{1, 2, \dots, k\}} \gamma_{ji} \quad (9.31)$$

模型参数 $\{(\alpha_i, \mu_i, \Sigma_i) | 1 \leq i \leq k\}$ 如何求解？

样本集 D ，可采用极大似然估计。最大化对数似然：

$$\begin{aligned} LL(D) &= \ln \left(\prod_{j=1}^m P_m(x_j) \right) \\ &= \sum_{j=1}^m \ln \left(\sum_{i=1}^k \alpha_i \cdot p(x_j | \mu_i, \Sigma_i) \right) \end{aligned} \quad (9.32)$$

若参数 $\{(\alpha_i, \mu_i, \Sigma_i) | 1 \leq i \leq k\}$ 能使 $\frac{\partial LL(D)}{\partial \mu_i} = 0$

$$\sum_{j=1}^m \frac{\alpha_i \cdot p(x_j | \mu_i, \Sigma_i)}{\sum_{l=1}^k \alpha_l \cdot p(x_j | \mu_l, \Sigma_l)} (\lambda_j - \lambda_i) = 0$$

由(9.30), $\gamma_{ji} = P_m(z_j=i|x_j)$

$$\mu_i = \frac{\sum_{j=1}^m \gamma_{ji} x_j}{\sum_{j=1}^m \gamma_{ji}} \quad (9.34)$$

$$\frac{\partial LL(D)}{\partial \Sigma_i} = 0$$

$\alpha_i \geq 0, \sum_{i=1}^k \alpha_i = 1$

LL(D) 在矩阵形式

$$\Sigma_i = \frac{\sum_{j=1}^m \gamma_{ji} (x_j - \mu_i)(x_j - \mu_i)^T}{\sum_{j=1}^m \gamma_{ji}} \quad (9.35)$$

$$LL(D) + \lambda \left(\sum_{i=1}^k \alpha_i - 1 \right) \quad (9.36)$$

$$\frac{\partial}{\partial \lambda} (9.36) = 0$$

$$\sum_{j=1}^m \frac{P(x_j | \mu_i, \Sigma_i)}{\sum_{l=1}^k \alpha_l \cdot P(x_j | \mu_l, \Sigma_l)} + \lambda = 0 \quad (9.37)$$

$$\alpha_i = \frac{1}{m} \sum_{j=1}^m \gamma_{ji}$$

样本加权平均

高斯混合模型 EM 算法：

E步：计算每个样本属于每个高斯成分的后验概率 γ_{ji} ；

M步：根据(9.34)、(9.35)、(9.38)更新

模型参数 $\{(\alpha_i, \mu_i, \Sigma_i) | 1 \leq i \leq k\}$

混合级由样本所属成分平均后概率平均之

9.5 密度聚类

DBSCAN 是基于一组“邻域”参数 $(\varepsilon, \text{MinPts})$ 来刻画样本分布的紧密程度。给定数据集 $D = \{x_1, x_2 \dots x_m\}$

- ε -邻域：对 $x_j \in D$ ，其 ε 邻域包含样本集 D 中与 x_j 的距离不大于 ε 的样本。

$$\text{即 } N_\varepsilon(x_j) = \{x_i \in D \mid \text{dist}(x_i, x_j) \leq \varepsilon\};$$

- 核心对象 (core object)：若 x_j 的 ε 邻域至少包含 MinPts 个样本。

即 $|N_\varepsilon(x_j)| \geq \text{MinPts}$, 则 x_j 是一个核心对象

- 密度直达 (directly density-reachable)：若 x_j 位于 x_i 的 ε 邻域中，且 x_i 是核心对象
称 x_j 由 x_i 密度直达

- 密度可达 (density-reachable)：对 x_i 与 x_j ，若存在样本序列为 $P_1, P_2 \dots P_n$

其中 $P_1 = x_i$, $P_n = x_j$ 且 P_{i+1} 由 P_i 密度直达，称 x_j 由 x_i 密度可达

- 密度相连 (density-connected)：对 x_i 与 x_j ，若存在 x_k 使得 x_i 与 x_j 均由 x_k
密度直达，称 x_i 与 x_j 密度相连

DBSCAN 的簇度义为：给定邻域参数 $(\varepsilon, \text{MinPts})$ ，簇 $C \subseteq D$ 是满足
以下性质的非空样本子集：

连接性： $x_i \in C, x_j \in C \Rightarrow x_i$ 与 x_j 密度相连 (9.39)

最大性： $x_i \in C, x_j$ 由 x_i 密度可达 $\Rightarrow x_j \in C$ (9.40)

9.6 层次聚类

最小距离： $d_{\min}(C_i, C_j) = \min_{x \in C_i, z \in C_j} \text{dist}(x, z)$ (9.41)

最大距离： $d_{\max}(C_i, C_j) = \max_{x \in C_i, z \in C_j} \text{dist}(x, z)$ (9.42)

平均距离： $d_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{z \in C_j} \text{dist}(x, z)$ (9.43)