机器学习大纲

李向阳　d1142845997@gmail.com

目录

1 写在前面

一直想系统的学习并总结一下机器学习的各种算法, 因此写下了这个系列, 算是一个总结, 也方便以后回顾.

本系列所有文章均是自己的学习笔记, 绝大部分内容来自机器学习方面的参考书籍、其他人写的一些博客或者 notes、以及维基百科等, 自己并没有任何原创, 大多数都是把推导的过程详细的写了出来, 以供自己理解.

虽然各个算法之间相对比较独立, 不过每个人的知识体系略有不同, 接受算法的顺序最好适合自己, 这个系列的推荐阅读顺序是

(1) 线性回归学习笔记

涉及了基本线性回归、正则化、Ridge、Lasso、Bayes 线性回归.

(2) Logistic 回归学习笔记

涉及了基本 Logistic 回归、损失函数、决策边界、正则化、(变分) 贝叶斯 Logistic 回归

(3) Softmax 回归学习笔记

涉及到了模型参数估计 (推导)、编程计算

(4) SVM 学习笔记

涉及到了基本 SVM、损失函数、核方法、软间隔与正则化、SMO 方法

(5) BP 神经网络学习笔记

涉及了基础的神经网络、反向传播算法 (推导)、损失函数

(6) CNN 学习笔记

涉及了卷积神经网络的基本概念

(7) EM 算法学习笔记

涉及了 EM 算法、混合高斯聚类模型、EM 算法的一般形式 (证明)

(8) K-Means 聚类学习笔记

涉及到了 K-Means 聚类 (快速聚类)、分级聚类 (谱系聚类)、混合高斯聚类

(9) 朴素贝叶斯学习笔记

涉及到了朴素贝叶斯分类器、判别模型与生成模型、线性判别分析 (LDA)、Gauss 判别分析、Bayes 判别

(10) 决策树学习笔记

涉及到了决策树 (ID3 算法)、C4.5 算法、CART 算法

(11) AdaBoost 学习笔记

涉及到了 AdaBoost 算法、Boosting 方法、Bagging 方法、梯度提升、提升树、随机森林

(12) 模型评价学习笔记

涉及到了查准率、查全率、ROC 曲线

(13) PCA 与 SVD 学习笔记

涉及到了主成分分析、SVD

(14) Bayes 方法学习笔记

涉及到了 Bayes 方法的基本原则

(15) MCMC 学习笔记

涉及到了 MCMC 采样方法、马尔科夫链、Gibbs 采样

(16) 隐马尔可夫模型学习笔记

涉及到了隐马尔可夫模型的基本概念

这个顺序不是严格按照我写作的次序, 有一些小的变动. 而且写作的时候可能更改某些东西, 比如先写了 Logistic 回归, 再写了 SVM, 写完 SVM 时又对 Logistic 回归有了新的理解, 所以又扩充了 Logistic 回归笔记的内容. 总之, 笔记里的东西是动态变化的 (本篇大纲也是动态变化的), 回顾时, 尽量按照这个顺序即可.

2 基本概念

2.1 关于记号

记号是一个蛋疼的东西, 混乱的记号让人作呕, 特别是对于强迫症有时无法忍受, 而且记号不明确也会影响理解. 当然, 单独理解每个算法时记号只要能够区分就行, 但是为了尽量保持本系列笔记记号的一致性, 在此还是要说一下记号.

(1) 字体:

不加粗的字母表示单独的一个数, 加粗的字母 (称为粗体或黑体) 表示一个向量 (为了方便, 一般为列向量).

有些文献中把粗体的 $\boldsymbol{x}$ 用 $\mathbf{x}$ 来表示 (或者把粗体的 $\boldsymbol{X}$ 用 $\mathbf{X}$ 表示), 我习惯仍将粗体倾斜 (虽然不倾斜可能更便于区分).

当然, 为了区分记号, 也可引入花体. 总之, 对比以下字母即可:

$$x, y, \boldsymbol{x}, \boldsymbol{y}, \mathbf{x}, \mathbf{y}, X, Y, \boldsymbol{X}, \boldsymbol{Y}, \mathbf{X}, \mathbf{Y}, \mathcal{X}, \mathcal{Y}$$

$$n, \mathrm{N}, N, \boldsymbol{N}, \mathbf{N}, \mathcal{N}, \mathbb{N}$$

$$l, \mathrm{L}, L, \mathcal{L}, \mathfrak{L}, \mathscr{L}, \ell$$

另外, 注意 beamer 中的字体样式更是多种多样.

(2) 样本:

一般有 2 种表示方式.

我最开始用粗体的 $\boldsymbol{x}$ 表示样本, 它有 n 个分量, 分量用不带粗体的下标表示, 即有 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T$, 用上标表示具体的样本, 即 $\boldsymbol{x}^{(i)}$ 表示第 i 个样本, 并设总共有 m 个样本, 即有

$$\boldsymbol{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \cdots, x_n^{(i)})^T, i = 1, 2, \cdots, m$$

采用这样的记号, 能够清楚的理解概念, 但推导时可能略显麻烦, 因此出现第 2 种方式: 仍用粗体的 $\boldsymbol{x}$ 表示样本, 但同时用带下标的 $\boldsymbol{x}_i$ 表示第 i 个样本, 注意 $\boldsymbol{x}_i$ 实际上有 n 个分量, 但为了推导方便并不写出, 直接理解为带 n 个分量的向量即可, 比如可以理解为双下标 $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{in})^T$, 或者带上标 $\boldsymbol{x}_i = (x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(n)})^T$(这样就和第 1 种方式类似, 只不过上下标相反), 但注意一定要加粗, 有些文档中甚至不加粗, 这很不好 (虽然读者能从上下文中看出, 但不加粗就是不好, 当然有些文章由于网站等显示原因不能加粗也无可厚非), 然后设样本个数为 N. 这也是一种常见的记号, 适用范围也更广, 我以后会经常采用这种方式.

以上两种方式, 我均用小写的 n 表示分量的个数 (维数), 而样本的个数用 m 或者 N 来表示, 也有的文献把分量个数或者维数用 D 来表示.

当然, 最简洁的一种表示是把所有的样本记为大写粗体的 $\boldsymbol{X}$(或者 $\mathbf{X}$), 也有文献中不加粗, 毕竟已经大写了, 但牵涉到多个大写字母时, 加粗还是很有必要的. 有时候可能只是笼统的表示 $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(m)}\}$ 或者 $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$, 有时候可能是把所有的样本按行并起来形成的矩阵 (维数按记号不同, 可能为 $m \times n, N \times n, N \times D$). 当然, 在统计学习中, 常见的是用 n 表示样本数, 用 p 表示变量个数, 样本矩阵维数为 $n \times p$.

最后, 补充一点关于样本的理解, 有时候模型中会出现 y(或粗体的 $\boldsymbol{y}$), 比如分类问题中的类标签 y 或者回归问题中的预报值 y, 本质上 $\{\boldsymbol{x}, y\}$ 合在一起是一组样本, 应该当做一个粗体的 $\boldsymbol{x}$(相当于把 y 当成 $\boldsymbol{x}$ 的第 $n+1$ 个分量), 只不过有时候为了讨论方便分开写. 此外, 有时为了讨论方便, 会引入 $x_0 = 1$.

(3) 未知参数:

未知参数可能有多个 (比如设有 n 个), 一般把它们写成一个列向量, 所以应用粗体表示, 我一般用粗体的 $\boldsymbol{\theta}$ 或者 $\boldsymbol{w}$ 来表示, 即有

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_n)^T 或 \boldsymbol{w} = (w_1, w_2, \cdots, w_n)^T$$

注意, 有时为了讨论的方便, 会引入 θ_0 或者 w_0, 当然也有用参数 b 来表示的.

当然, 未知参数的形式可能很复杂, 比如多元高斯分布 $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, 整个均值向量和协方差矩阵可能都是待估参数. 再比如混合高斯模型, 有 K 个高斯分布 $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), k = 1, 2, \cdots, K$, 此处每个 $\boldsymbol{\mu}_k$ 都是一个均值向量 (比如说有 n 个分量), 其分量形式还可以借助双下标或者上标来表示 (参考样本), 而每个 $\boldsymbol{\Sigma}_k$ 都是一个矩阵, 完整的写出分量形式实在复杂, 大多数时候也没有必要, 因此只要理解其含义就可以了.

(4) 隐变量:

隐变量 (潜变量) 用粗体的 $\boldsymbol{z}$ 来表示, 与粗体的 $\boldsymbol{x}$ 相对应, 有时也笼统的把所有的隐变量用大写粗体的 $\boldsymbol{Z}$ 来表示.

注意, 有时候为了讨论的方便, 会把未知参数和隐变量通记为 $\boldsymbol{Z}$, 因为它们都是未知的东西 (一个是待估计的未知参数, 一个是不可观测的变量).

(5) 迭代:

求解优化问题我们一般用迭代法, 表示第 k 步时, 可灵活应用上标或者下标.

(6) 双下标:

我个人是比较反感双下标的, 因为这通常意味着双 $\sum$ 或双 $\prod$ 的出现, 但有时又不得不引入双下标, 这时注意理解清楚变量所代表的含义即可.

3 关于统计与机器学习

目前比较火的是深度学习, 之前是统计机器学习. 那么统计和机器学习之间的关系和区别到底是什么?

我这里只说一下自己的个人理解. 机器学习中用到了很多统计学的思想, 但是关注点是不一样的. 之前在概率论与数理统计的课本中曾提到, 说假设检验是衡量一个人是否真正掌握统计方法的试金石. 确实, 假设检验是统计学的亮点. 可是机器学习呢? 我们貌似很少看到过假设检验, 因为机器学习只关注的是学习器的表现效果, 而且机器学习有自己的一套方法去衡量它, 也就是模型的泛化能力.

再说一个方法上的不同. 在机器学习中, 一般我们设样本有 n 个特征 (变量), 即 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$, 假设有 m 个样本, 即 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m$, 这里我们采用双下标 (当然, 在统计中, 一般用 n 表示样本个数, 用 p 表示变量个数, 即维数), 即有

$$\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{in})^T, i = 1, 2, \cdots, m$$

把样本按行并起来, 每一列表示一个特征 (变量), 则所有的样本组成了一个维数为 $m \times n$ 样本矩阵, 如下

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_m^T \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

其中的 x_{ij} 便表示第 i 个样本在第 j 个特征 (变量) 上的取值, 或者说第 j 个特征 (变量) 在第 i 个样本上的取值.

而在统计学中呢? 回顾一下数据分析方法, 我们是假设了一个 n 维总体 $(X_1, X_2, \cdots, X_n)^T$, 每个维度是一个特征 (变量), 然后从总体中抽样得到了 m 个样本, 其中 $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{in})^T, i = 1, 2, \cdots, m$, 同样的, 我们也是把样本按行并起来得到了与上面一样的样本矩阵 (当然, 对于有的算法, 有时为了推导和编程的方便, 是把样本直接按列并起来行成样本矩阵, 即 $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m)$, 相当于对原来的样本矩阵取了个转置, 维数为 $n \times m$, 这些在推导中都可以借助线性代数中的矩阵分块来理解).

不过, 在统计学中, 我们是假设这个总体服从一定的分布, 比如高维正态分布等, 分布的参数可以由样本估计得到, 然后在此基础上进行的各种讨论.

说到这里, 其实应该就可以把机器学习和之前学过的数据分析方法相统一起来了. 事实上, 很多机器学习的算法也是依靠的统计学总体的假设, 尤其是贝叶斯统计学, 具体可参见 Bishop 的 PRML 一书.

4 推荐参考资料

网上的好资源很多, 这里总结几个, 也是本系列的主要参考来源.

(1) Andrew Ng 的机器学习课程及其周边资料.

讲的确实不错, 符号也比较一致, 本系列的第一套符号来自于此. 在 Coursera 上的公开课是 https://class.coursera.org/ml-005/lecture, 里面有视频和课件可供下载, 其中课件不错. 这门课的原型是斯坦福大学的 CS229 课程, 可见 http://cs229.stanford.edu/, 上面也有一些不错的资料 (比如 materials: http://cs229.stanford.edu/materials.html), 此外, 还有对应的一个 openclassroom, 可见 http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=MachineLearning, 上面主要有习题和数据, 还有部分 Matlab 编程.

(2) pluskid 的博客

写的真的很好, 比较清楚, 可见 http://blog.pluskid.org/(一些早期博客) 和 http://freemind.pluskid.org/(近期博客).

5 总结

附录