

# 机器学习算法举例

## 1. Supervised learning

### 1.1. Generalized linear model

#### 1.1.1. Lasso

#### 1.1.2. Logistic regression

### 1.2. Linear and quadratic discriminant analysis

#### 1.2.1. Dimensionality reduction using Linear Discriminant Analysis

#### 1.2.2. Dimensionality reduction using quadratic Discriminant Analysis

### 1.3. Kernel ridge regression (核岭回归)

### 1.4. Support vector machines

#### 1.4.1. Classification

#### 1.4.2. Regression

### 1.5. Stochastic Gradient Descent (随机梯度下降)

### 1.6. Nearest Neighbors

### 1.7. Naive Bayes (文本分析)

### 1.8. Decision Trees

#### 1.8.1. Classification

#### 1.8.2. Regression

### 1.9. Random forests

#### 1.9.1. Classification

#### 1.9.2. Regression

### 1.10. Neural network models

## 2. Unsupervised learning

### 2.1. Gaussian mixture models (高斯混合矩阵)

### 2.2. K-means

### 2.3. Principal component analysis

### 2.4. Kernel Density Estimation (核密度估计)

# 标准机器学习算法实现规范

## 一、命名规范

### 1、命名包

- (1) 包的命名都使用小写的英文字母组成，每个包名称之间用点号分隔开。
- (2) 全局包的名字这里约定为 `com.shu`

### 2、类、编译单元命名

#### (1) 命名类

约定使用完全的英文描述符，所有单词的第一个字符大写，并且单词中大小写混合。类名应是单数形式。示例：Kmeans， RandomForest。

#### (2) 命名编译单元

编译单元在这个情况下是一个源码文件，应被赋予文件内定义的主要的类的名字。用与类相同的名字命名文件，扩展名.java 作为后缀名。示例：RandomForest.java。

### 3、成员函数命名标准

#### (1) 命名成员函数

成员函数的命名应采用完整的英文描述符，大小写混合使用：所有中间单词的首字母大写。有四个函数在类中一定要有，分别是：用于初始化的构造函数(构造函数名与类名相同)，训练模型函数 fit， 预测函数 predict， 评分函数 score。其他成员函数的名字请自行起名，但第一个单词应使用一个有强烈动作色彩的动词。

#### (2) 成员函数的可见性

良好的程序设计应尽可能减小类与类之间的耦合，所遵循的法则是：尽量限制成员函数的可见性，如没必要公有（public），就定义为保护（protected）；没必要保护，就定义为私有（private）。

### 4、字段、属性命名标准

应使用完整的英文描述符来命名字段，以便使字段所表达的意思一目了然。像数组或者是矢量这样是集合的字段，命名时应使用复数来表达它们为多值。示例：orderItems。

## 5、局部变量命名标准

命名局部变量遵循与命名字段一样的规范

## 6、类及成员函数参数命名标准

- (1) 常数参数使用希腊字母罗马音进行规范 (alpha, beta, gamma, delta) .
- (2) 若参数需要用两个及以上单词命名, 单词间用下划线”\_”分割, 单词首字母全小写 (如 fit\_intercept), 若第二次单词只有一个字母, 则大写 (如 copy\_X) .
- (3) 对于长单词 (10 字母以上) 请用缩写代替
- (4) 所有参数需要有默认值
- (5) 参数个数限制在 10 个之下

## 二、注释规范

### 1、注释约定

Java 有 3 种注释语句风格:

- 以/\*\*开始, \*/结尾的文档注释;
- 以/\*开始, \*/结尾的 c 语言风格注释;
- 以//开始, 以代码行末尾结束的单行注释。

下面是对注释语句的用法的一个概括, 以及几个例子。

注释语句类型	用法	示例
文档注释	在接口、类、成员函数和字段声明之前紧靠它们的位置用文档注释进行说明。文档注释由 javadoc 处理, 为一个类生成外部注释文档, 如下所示。	<pre>/**  * Customer (顾客). 顾客是指作为我们的  * 服务及产品的销售对象的任何个人或组织。  *  * @author S.W. Ambler  */</pre>
C 语言风格注释	采用 C 语言风格的注释语句将无用的代码注释掉。保留这些代码是因为用户可能改变想法, 或者只是想在调试中暂时不执行这些代码。	<pre>/*  * 这部分代码已被它前面的代码替代, 所以于  * 1999 年 6 月 4 日被 B. Gustafsson  * 注释掉。如果两年之后仍未用这些代码, 将</pre>

		其删除。  ...（源代码）  */
单行注释	在成员函数内部采用单行注释语句对业务逻辑、代码片段和临时变量声明进行说明。	// 因为让利活动  // 从 1995 年 2 月开始，  // 所以给所有超过 \$1000 的  // 发货单 5% 的折扣。

使用单行注释方式来说明业务逻辑，使用 c 语言风格的注释屏蔽旧的代码。

## 2、包的注释标准

对于每个包，应说明：

### （1）包的基本原理

（2）包中的类。在包中要包含一个类的列表，每个类用简短的一行文字来说，以便了解这个包中包含什么。

## 3、类的注释标准

以下的信息应写在文档注释中紧靠类的定义的前面（采用 javadoc 形式）：

- （1）类的目的和作用。
- （2）已知的问题，还应注明为什么不解决问题的原因
- （3）类的开发/维护历史。通常要包含一个历史记录表，列出日期、类的作者和修改概要。
- （4）版权信息。此处标注个人信息即可

## 4、成员函数注释标准

### （1）成员函数的函数头

每一个成员函数都应包含某种称之为“成员函数文档”的函数头。这些函数头在源代码前面，用来记录所有重要的有助于理解函数的信息：

- 成员函数做什么以及它为什么做这个。
- 哪些参数必须传递给一个成员函数。
- 成员函数返回什么。
- 已知的问题。

示例：

```
/**
 * 执行 SELECT 函数，可以分页。按照输入的页码、每页的行数，返回当前页的结果
 * 集
 *
 * 又多行记录：结果放入 VECTOR，VECTOR 里面的每一项代表一行记录。
 *
 * VECTOR 里面包含的 String[]就是真正的结果
 *
 * @param pageNo 返回 SELECT 结果中的第几页数据
 *
 * @param pageCount 每页的最大行数
 *
 * @throws java.lang.Exception
 *
 * modified by fjy 2003-07-03 增加了 pageCount 参数的检查
 */
public void ExecuteSelect(int pageNo,int pageCount) throws Exception {
}
```

## （2）内部注释

在成员函数内部还需要加上注释语句来说明你的工作。内部注释应采用两种方式：对于业务逻辑采用单行注释；采用 c 语言风格注释去掉无用的代码。

## 5、局部变量注释标准

声明和注释局部变量有几种约定：

- a) 一行代码只声明一个变量。
- b) 用一个行内注释语句说明局部变量。（行内注释是一种紧接在同一行的命令代码后，用符号//标注出来的单行注释风格）。

## 6、成员函数参数注释标准

成员函数的参数在采用 javadoc @param 标识的头文件中注释。应说明：

- c) 参数用来做什么。
- d) 任何约束或提前条件。比如参数的值域、正负。
- e) 示例。若应传递什么样的参数不明显，那么应该在注释种给出例子。

### 三、一般规定

约定目标	约定
存取成员函数	<p>考虑对数据库中的字段使用滞后初始化</p> <p>使用存取函数获得和修改所有字段</p> <p>对常量采用存取函数</p> <p>对于集合，加入成员函数来插入和删除项</p> <p>一旦可能，将存取函数置为被保护类型，不是公共类型</p>
字段	<p>字段永远为私有类型</p> <p>不要直接访问字段，应使用存取成员函数</p> <p>不要使用静态常量字段（常量），应使用存取成员函数</p> <p>不要隐藏名字</p> <p>一定要初始化静态字段</p>
类	<p>最小化公共和保护接口</p> <p>在开始写代码之前定义一个类的公共接口</p> <p>按以下顺序声明一个类的字段和成员函数：</p> <ul style="list-style-type: none"> <li>● 构造函数</li> <li>● 公共成员函数</li> <li>● 被保护成员函数</li> <li>● 私有成员函数</li> <li>● 私有字段</li> </ul>
局部变量	<p>不要隐藏名字</p> <p>一行代码只声明一个局部变量</p> <p>用一个行内注释说明局部变量</p> <p>在使用局部变量之前声明它</p>

	仅将局部变量用于一件事
成员函数	<p>给代码加上注释</p> <p>给代码分段</p> <p>使用空白，控制结构之前用一个空行，成员函数之前用两个空行</p> <p>一个成员函数应能在 30 秒内让人理解</p> <p>写短小单独的命令行</p> <p>尽量限制成员函数的可见性</p> <p>说明操作的顺序</p>

## 四、API

每个算法需提供参数列表与方法列表（下面例子的列表显示在下页文档示例中）。

例：K-means 算法

```

Class KMeans (n_clusters=8, n_init=10, max_iter=300, ...) {
    //初始化
    _init_( n_clusters=8, n_init=10, max_iter=300, ...) {
        //具体实现
    }
    //训练模型
    def fit (X, y) {
        //具体实现
    }
    //预测
    def predict (X) {
        //具体实现
    }
    //评分
    def score () {
        //具体实现
    }
}

```