

Git 团队协作(读书笔记)

刘涛
中科院计算所

版本:0.1

更新:June 19, 2019

引言

Git 的历史:

1. Linux 内核开发的专有版本控制系统 BitKeeper 遭到反对。
2. Linus Torvalds 花了两周时间写了一个替代 BitKeeper 的工具:Git。
3. Git: 一个用 Unix 风格编写的 shell 脚本拼接起来的小程序。

注 具有讽刺意味的是:*Git* 的分布式是使用 *rsync* 实现的,而 *rsync* 恰好就是那个将 *BitKeeper* 推下历史舞台的 *Linux* 开发者。

1 团队作战

理解并解决背后的人际关系问题,构建充满了信任、尊重和关爱的团队。

1.1 团队成员

用户, 测试, 安全、QA, 运维, 开源社区, 减少代码闲置 ==> 评审

1.2 思维策略

思维策略的三个维度:

1. 创造性思维

- 我的理解: 类似于发散性思维
- 特点: 预见未来、另辟蹊径、头脑风暴、灵光一闪、勇于质疑、保持专注

2. 理解性思维

- 分为两类: 理解信息(分析型), 理解他人(同理心)
- 分析型思维的特点: 评估现状、阐明现状、善于组织、敏锐感知、产生共鸣、善于表达

3. 决策性思维

- 弱点: 缺乏耐心
- 特点: 分清主次、善于总结、验证结论、身体力行、价值驱动、相信直觉

2 命令与控制

2.1 项目治理

了解一段代码的归属。

2.1.1 版权和贡献者协议

版权只保护作品的具体实现。

贡献者协议存在的原因：允许项目无需个人贡献者的同意就可以更改项目的分发许可。

2.1.2 分发许可

1. MIT 许可证：注明原作者，不需要为衍生软件负责。例如：jQuery、Rails。
2. Apache 许可证：将原作者的专利授权给用户，要求作者提供变更说明。例如：Apache、Subversion、NuGet。
3. GNU 许可证：以相同协议共享。例如：Linux、Git、Word Press。

2.1.3 领导力模型

- 仁慈的独裁者(BDFL)
- 共识驱动、主管批准
- 技术评审委员会或项目管理委员会

2.1.4 行为守则

Code of Conduct, 参考：Flickr 社区、Drupal 社区等。

2.2 访问模型

Git 的 MBTI 性格测试结果：INTP(内向、直觉、理性、理解)

2.2.1 适合分散贡献者仓库的模型

用 diff 创建补丁文件, 邮件评审, 用 patch 命令更新仓库。

2.2.2 适合并列贡献者仓库的模型

fork, clone, checkout, commit, push, pull request

2.2.3 共同维护的模型

内部团队（和个人团队）中最常见的权限模型。

我们组目前使用的访问模型。

3 分支策略

3.1 理解分支

每个 Git 仓库都包含一个提交库。这些提交通过元数据相互链接，每个提交包含一个指向自身父提交的引用。

Git 中的分支实际上是一个指向某个特定提交的命名指针。

你创建的任何提交对象都是本地的，并且只属于你，直到你选择主动将它们共享到一个远程仓库。

一帮来说，一种约定包含两类分支：

- 长期活跃的公开分支
- 短暂的私有分支

3.2 挑选约定

形成文档的约定让新参与者更加容易上手。

3.3 几种约定

3.3.1 主线分支开发

3.3.2 功能分支部署

3.3.3 状态分支

3.3.4 计划部署

3.4 更新分支

4 工作流

5 单人团队

6 回滚、还原、重置和变基

7 多人团队

8 准备评审

9 寻找并修复 bug

10 Github 上的开源项目

11 Bitbucket 上的私有团队工作

12 Gitlab 上自行管理的协作

13 模板设计

此模板设计的初衷是为了记录笔记，在 2013 年开始构想，初版我们设计了非常美观的定理环境，并设计了 3 套不同的颜色主题。但我们发现在实际记笔记的时候，过多的定理区块使得整个文章并不是非常美观，所以我们将 ElegantNote 更新为 ElegantBook 模板，在后面被用户熟知。而 ElegantNote 的设计自此停止。

2018 年，在被一些用户“催更”之后，ElegantBook 迎来重大更新，原先浮动的定理环境用 tcolorbox 全部改写。时至今日，ElegantBook 版本为 3.05。之后，我们便想把 ElegantNote 也彻底更新下，放弃 ElegantBook 中的定理环境设计，改用更为紧凑，更加朴素的定理环境，设计更适合笔记记录和笔记阅读的 L^AT_EX 模板。

在一些朋友的建议和启发下，我们基于标准的 L^AT_EX 文类 article 重新设计了新版 ElegantNote 模板，在此特别感谢！新模板有下面几个特性：

- 添加护眼模式，颜色为绿豆沙颜色，默认为白色背景；
- 适配不同设备，包括 Pad（默认），Kindle，PC（双页），通用（A4）；
- 5 套颜色主题，分别是：green（默认），cyan，blue，sakura，black；
- 语言模式支持：中文（默认），英文；
- 支持 PDFL^AT_EX 和 XeL^AT_EX 编译；
- 更加美观的图表标题格式，列表环境，数学字体等。

13.1 护眼模式

本模板增加了护眼模式，默认为不开启，开启的方法如下：

```
\documentclass[geye]{elegantnote} % or
\documentclass[mode=geye]{elegantnote}
```

评论 此次更新只添加了护眼模式(绿豆沙色背景),如果您有希望增加其他颜色,可以在 [Github/ElegantNote](#) 创建 *issues* 或者提交 *pull request*。

13.2 设备选择

为了让笔记方便在不同设备上阅读,免去切边,缩放等操作,本模板适配不同的屏幕大小,分别为 Pad, Kindle, PC, A4。不同屏幕的选择为

```
\documentclass[device=pad]{elegantnote}
\documentclass[device=kindle]{elegantnote}
\documentclass[device=pc]{elegantnote}
\documentclass[device=normal]{elegantnote}
```

注 也可以采取直接赋值的方法选择屏幕,比如:

```
\documentclass[pad]{elegantnote}
\documentclass[kindle]{elegantnote}
\documentclass[pc]{elegantnote}
\documentclass[normal]{elegantnote}
```

注 如果想要得到普通的 A4 纸张大小的 *PDF*, 需要选择 *device=normal*, 而不是选择 *device=pc*, 因为 *device=pc* 实际上设置的是电脑双页模式。

13.3 颜色主题

本模板内置 5 套颜色主题,分别是 *green*, *cyan*, *blue*, *sakura*, *black*。其中 *green* 为默认颜色主题,如果用户不需要彩色,可以选择 *black* 主题。颜色主题的启用方法和之前一样:

```
\documentclass[green]{elegantnote}
```

```
\documentclass[color=green]{elegantnote}
...
\documentclass[black]{elegantnote}
\documentclass[color=black]{elegantnote}
```

13.4 语言模式

本模板内含两套语言环境,改变语言环境会改变图表标题的引导词(图,表),文章结构词(比如目录,参考文献等),以及定理环境中的引导词(比如定理,引理等)。不同语言模式的启用如下:

```
\documentclass[cn]{elegantnote}
\documentclass[lang=cn]{elegantnote}
\documentclass[en]{elegantnote}
\documentclass[lang=en]{elegantnote}
```

注 不管选用中文环境还是英文环境均可输入中文。另外如果在笔记中使用了抄录环境(*lstlisting*),并且其中包括了中文,请务必使用 *XeLaTeX* 编译。本说明文档也只能通过 *XeLaTeX* 编译。

13.5 编译方式

本模板支持两种编译方式,PDFLaTeX 和 XeLaTeX,选用 PDFLaTeX 编译的话,如果用到了中文,则会调用 *ctex* 宏包,而如果选用 XeLaTeX 编译的话,则会调用 *xeCJK* 宏包。模板测试环境为 Win10 + T_EX Live 2018,设定的字体为 Windows 中的宋体、楷体、黑体、微软雅黑等。如果你的电脑是 Mac/Linux 系统,而且采用 XeLaTeX 编译的话,请把 *elegantnote.cls* 中字体改为自己系统的字体。

13.6 定理类环境

此模板采用了 `amsthm` 中的定理样式, 使用了 4 类定理样式, 所包含的环境分别为

- 定理类: theorem, lemma, proposition, corollary;
- 定义类: definition, conjecture, example;
- 备注类: remark, note, case;
- 证明类: proof。

评论 在选用 `lang=cn` 时, 定理类环境的引导词全部会改为中文。

14 写作示例

我们将通过三个步骤定义可测函数的积分。首先定义非负简单函数的积分。以下设 E 是 \mathcal{R}^n 中的可测集。

定义 14.1 (可积性) 设 $f(x) = \sum_{i=1}^k a_i \chi_{A_i}(x)$ 是 E 上的非负简单函数, 其中 $\{A_1, A_2, \dots, A_k\}$ 是 E 上的一个可测分割, a_1, a_2, \dots, a_k 是非负实数。定义 f 在 E 上的积分为

$$\int_E f dx = \sum_{i=1}^k a_i m(A_i). \quad (1)$$

一般情况下 $0 \leq \int_E f dx \leq \infty$ 。若 $\int_E f dx < \infty$, 则称 f 在 E 上可积。

一个自然的问题是, Lebesgue 积分与我们所熟悉的 Riemann 积分有什么联系和区别? 之后我们将详细讨论 Riemann 积分与 Lebesgue 积分的关系。这里只看一个简单的例子。设 $D(x)$ 是区间 $[0, 1]$ 上的 Dirichlet 函数。即 $D(x) = \chi_{Q_0}(x)$, 其中 Q_0 表示 $[0, 1]$ 中的有理数的全体。根据非负简单函数积分的定义, $D(x)$ 在 $[0, 1]$ 上的 Lebesgue 积分为

$$\int_0^1 D(x) dx = \int_0^1 \chi_{Q_0}(x) dx = m(Q_0) = 0 \quad (2)$$

表 1: 燃油效率与汽车价格

	(1)	(2)
燃油效率	-238.90*** (53.08)	-49.51 (86.16)
汽车重量		1.75*** (0.641)
常数项	11,253*** (1,171)	1,946 (3,597)
观测数	74	74
R^2	0.220	0.293

括号内为标准误

*** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$

即 $D(x)$ 在 $[0, 1]$ 上是 Lebesgue 可积的并且积分值为零。但 $D(x)$ 在 $[0, 1]$ 上不是 Riemann 可积的。

定理 14.1 (Fubini 定理) 若 $f(x, y)$ 是 $\mathcal{R}^p \times \mathcal{R}^q$ 上的非负可测函数, 则对几乎处处的 $x \in \mathcal{R}^p$, $f(x, y)$ 作为 y 的函数是 \mathcal{R}^q 上的非负可测函数, $g(x) = \int_{\mathcal{R}^q} f(x, y) dy$ 是 \mathcal{R}^p 上的非负可测函数。并且

$$\int_{\mathcal{R}^p \times \mathcal{R}^q} f(x, y) dx dy = \int_{\mathcal{R}^p} \left(\int_{\mathcal{R}^q} f(x, y) dy \right) dx. \quad (3)$$

证明. Let z be some element of $xH \cap yH$. Then $z = xa$ for some $a \in H$, and $z = yb$ for some $b \in H$. If h is any element of H then $ah \in H$ and $a^{-1}h \in H$, since H is a subgroup of G . But $zh = x(ah)$ and $xh = z(a^{-1}h)$ for all $h \in H$. Therefore $zH \subset xH$ and $xH \subset zH$, and thus $xH = zH$. Similarly $yH = zH$, and thus $xH = yH$, as required.

回归分析 (regression analysis) 是确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法。根据定理 14.1, 其运用十分广泛, 回归分析按照涉及的变量的多少, 分为一元回归和多元回归分析; 按照因变量的多少, 可分为简单回归分

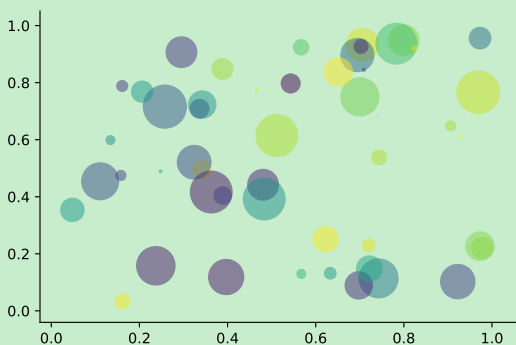


图 1: Matplotlib: Scatter Plot Example

析和多重回归分析;按照自变量和因变量之间的关系类型,可分为线性回归分析和非线性回归分析。

但是由于绝对值不易作解析运算,因此,进一步用残差平方和函数来度量总偏差。偏差的平方和最小可以保证每个偏差都不会很大。于是问题归结为确定拟合函数中的常数和使残差平方和函数最小。

- Routing and resource discovery;
 - Language Models
 - Vector Space Models
- Resilient and scalable computer networks;
- Distributed storage and search.

15 示例

```
\documentclass[gey,green,pad,cn]{elegantnote}  
  
\title{Note Example}  
  
\author{ddswhu}
```

```
\institute{Elegant \LaTeX{} Program}  
\version{1.00}  
\date{\today}  
  
\begin{document}  
  
\maketitle  
  
\section{Introduction}  
The content of Introduction.  
  
\end{document}
```