# 浅谈如何从程序开发的角度入门机器学习

广州中医药大学医学信息工程学院　罗晓牧

【摘要】机器学习是最近十分热门的研究领域。本文讨论了如何从程序开发的角度入门机器学习，包括设计系统流程、使用最新的工具、针对性的练习、记录成果等。采用自上而下的学习方法，十分适合大学本科学生、程序员的自学，有效的解决端到端的问题，激发学习热情，循序渐进的全面掌握机器学习领域。

【关键词】机器学习；程序开发；系统流程
DOI:10.19353/j.cnki.dzsj.2016.08.018

机器学习最近十分火热，尤其在Alphago 与李世石的围棋人机大战之后[1]，更掀起了一股机器学习的热潮。如何更好的学习这一领域的知识，是不少同学的疑惑。机器学习算法的传统学习路径是从统计学、概率论、线性代数、微积分等多种数学知识开始的。但作为计算机专业的学生、专业程序员、机器学习爱好者，这种自下而上的方法停留在算法层面，没有考虑到项目的开发和交付；同时，书本上的知识通常过于数学化，公式化，理论化，令人感到枯燥。虽然网上也已经有不少视频公开课，如Coursera [2]上面的机器学习公开课， 但是内容与教科书、博客差不多，没有将相关的知识融会贯通到一起。有没有更适合初学者的方法呢？在本文中，将探讨如何学习这一领域。

## 1　传统的机器学习途径

打开任意一本与机器学习相关的书籍，一般都会从问题的定义开始，然后到概念和算法的数学描述，复杂性不断增加。概念的定义与数学描述是十分清晰和简洁的，然而要对它们充分的掌握，则需要相关的数学背景去理解[3]。这也是为什么机器学习课程通常在研究生阶段才开设的原因。因为要教会某一特定科目的前提是需要足够的先修课程。例如，通常需要对以下科目具有良好的基础：统计、概率、线性代数、多元统计学，微积分等等。这种自下而上并且以算法为导向的学习方法，在机器学习领域相当主流。不少的网络论坛上，对于“如何开始机器学习”的问题，通常也是会得到类似的答案[4]。各种在线课程和公开课，也都是模仿学校的方法去进行教学。如果你已经有相当的学术背景，或已经具有硕士或博士学位，这种方法是不错的选择。然而，对于一般的程序开发者，这样的方法并不可取。这会让脚踏实地的软件工程师觉得应该重回学校，获得一个硕士或博士学位，然后才有资格去“做”机器学习。

这种自下而上的学习方法是错误的。为什么呢？假设你是一个年轻的软件开发人员，已经学会了某种编程语言，并开发出一些单机版的软件。当你告诉你的朋友和家人，想以每天编程作为职业，那么他们通常会告诉你，你需要在计算机科学方面取得一个学位，才有可以得到一份程序员的工作。然后，你就开始注册和修读计算机学位。几个学期之后，你已经接触到越来越多复杂的代数、微积分和离散数学。但你正在使用过时的编程语言，对编程和开发软件的热情也慢慢发生动摇了。

看到问题所在了吗？如果一个程序开发者想“做”机器学习，他真的需要花几年时间和数万元去获得数学知识和更高的学位吗？答案当然不是，还有一个更好的方法。

## 2　更好的入门方法

我们不能仅仅将模式翻转为自上而下，却使用相同的教学材料。正如计算机科学的课程从来没有把软件开发和软件交付等实际问题涵盖到课程内容当中，机器学习课程和教科书也远远不够，它们一般都停留在算法层面上。你需要一个自上而下的方法来学习机器学习。一种专注于实际效果的方法是：使用最新和最好的工具和平台，处理实际的端到端的机器学习问题，这才是正确的途径[5]。这种方法主要包括三个方面：设计系统流程、选择最新的工具、针对性的练习。

### 2.1 设计系统流程

一旦学会了使用某种工具和算法，很多同学可能马上会认为解决某一问题相当容易，并认为这个问题已经“解决”了。然而，这是远远不够的。怎么确定问题已经解决了呢？怎么确定结果是好的呢？怎么确定这些结果在实际的数据集上是可靠的呢？在解决机器学习问题上，需要一个系统化的流程。这类似于一个软件工程项目，一个好的系统流程才可以得到一个高质量的，可重复性的结果。系统流程包括以下几个方面：问题的定义、数据准备、检查算法、改进结果，如图1所示。

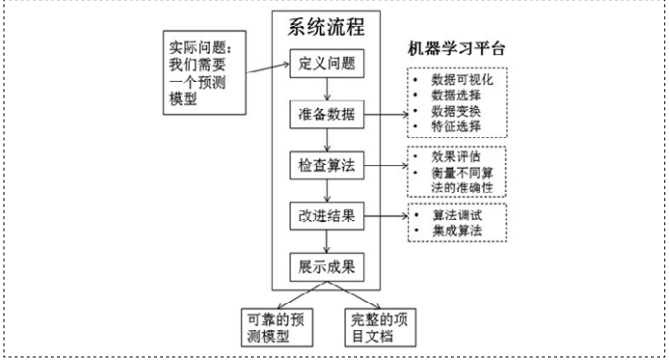


图1 系统流程

### 2.2 使用最新的工具

机器学习工具和库总在不断更新中，但是无论在任何时候都应该选择使用最新最适合的工具，而不是过时的算法或库文件，这样才可能自动化处理系统流程中的大部分环节，并得到可靠的高质量结果。可以通过咨询有经验的人，他们正在使用的最新工具，然后自己根据实际情况做出选择。针对不同类型的工作，可以使用不同的平台和工具。例如：

1) 一次性的预测模型： 对于“通过去年的销售数据预测今年的销售数据”等简单任务，可以使用Weka 平台[6]。因为通过Weka可以加载一个CSV文件，设计一个实验并马上得到最佳的模型，而无需写任何一行代码。

2) 嵌入式的预测模型：在现有的软件工程项目中增加预测功能模块，可以选择Python [7]中的scikit-learn库[8]，因为那样就可以在开发模型和部署项目时使用相同的语言。Ipython[9]是一种向更大的团队展示工作流程和预测结果的方式。

3) 深入研究的模型：对于需要深入理解并不断优化的预测模型，可以选择R平台[10]以及Caret扩展包[11]，因为那样就可以快速并自动地尝试很多最新的模型，并使用整个R平台实现更精准的特征

基金项目：国家自然科学基金（编号：61301294）。

选择以及算法优化实验。

事实上，这三种工具将根据具体的实际问题混合使用，需要不断学习并充分利用它们；但同时眼观六路、耳听八方，当更新更好的工具可以出现时，马上使用最新的工具，并将它们嵌入到你的系统流程中。

### 2.3 针对性的练习

通过大量的练习可以有效的学习软件开发，同样的方法也适用于机器学习。在每个项目中练习的环节越多，对机器学习则掌握得越好（最理想的是端到端的实际问题）。选择练习的数据集应该是真实的而不是人为的。现在已经有成百上千的公开数据集可以免费获取，并且数据集的复杂性不断增加：

1) 建议从规模较小的可以装进内存的数据集开始，如UCI机器学习库[12]。它们是公开的并且相对干净的数据集，可以作为建立系统流程和学习新工具的一个良好开端。

2) 在此基础上，可以研究规模更大一点的却仍然可以装进内存的数据集，比如来自Kaggle[13]和KDD cup[14]的比赛数据集。它们本身含有一定的噪声，需要更灵活的处理并使用不同的技巧。

### 2.4 记录成果

为每一个机器学习项目建立和维护一个半正式的工作成果：将所学所做详细记录下来，生成一个单独的文件夹，以便在日后的项目中参考并再次使用。这类似于软件开发者为每一个项目保留一个文件夹，用于日后参考和代码重用。这种做法将大大加速机器学习的进程。保留所有的脚本、代码以及生成的图片很重要，但更重要的是写下学习的感悟和发现，把它们想象为代码的注释。一个单独的记录文件可以是一个简单的PPT或文本文件形式，更加详细的记录文件可以是会议中的演讲稿或者是生成视频上传到Youku中[15]。

将每一个项目保存到公共的版本控制库（如GitHub）中[16]，这样其他的初学者可以借鉴你的项目并拓展你的工作。将你项目的链接放在博客、微博、微信、Linkedln或其他任何地方，通过网络平台展示你不断提升的技术和能力。GitHub中的项目文件正是公司在招聘过程中，在简历上真正关心的技能和已经取得的成果。

## 3 专为软件开发者提供的方法

使用上面所说的方法，可以作为软件开发者学习机器学习的开始，并能不断取得进步。开始使用这个方法，通常可能会产生一些疑虑，然而这些疑虑是不必要的，例如：

1) 不需要去写代码: 与web开发类似，学习机器学习不需要写大量的代码。例如像Weka这样的工具，可以在不需要任何编程的前提下，使得设计机器学习算法和建立预测模型的过程变得十分简单。编写代码可以让你掌握更多的不同工具，但这不是必须的。

2) 你不需要精通数学：如同软件开发者一样，不需要懂得计算复杂度或者O(n)再去编写代码。同理，可以在没有统计、概率论、线性代数的背景下解决端到端的机器学习问题。需要强调的是，我们的方法没有从理论开始学习，并不意味着忽视理论。当需要研究理论的时候，可以从开发中分离出来深入研究。事实上，因为解决机器学习问题是令人着迷的过程，理论的学习是在不知不觉中的。为了追求更好的结果和更精确的预测，你将使用任何可以找到的资源，提取其中的精华并运用到你的问题上。以软件开发的流程来学习机器学习，对初学者来说这个方法则是相当有意义的。

3) 不需要一个更高的学位：机器学习方面的知识都是公开的，可以今天马上进行自学，并不需要通过大量的时间和金钱去获得一个学位再开始。开始解决机器学习问题并完成了一个小型的项目后，再去考虑获得学位的事情。到那时候，你将对整个领域和自己感兴趣的部分有更深入的理解。

4) 不需要一个大数据集：机器学习算法在小数据集上设计和理解更佳。数据足够小则可以全部加载到内存中，并使用普通的台式计算机进行处理。大数据并不等于机器学习。

5) 不需要一台超级计算机：虽然某些最先进的算法，如深度学习确实需要非常强悍的多核GPU资源，但同时它们也是可以解决小问题的算法，使用你的台式计算机CPU就可以了。你可以马上开始机器学习算法，不必等到拥有一台超级的计算机才开始。在决定去买一台超级计算机或者租用昂贵的EC2[17]之前，最好花时间用于理解这些算法如何用于更小但更加容易理解的数据集之上，从中获得最大的收益。

6) 不需要大量的时间：如前所述，处理机器学习问题会让人上瘾。如果在机器学习竞赛中被超越了，你会很乐意牺牲一个月晚上看电视的时间将算法的准确度提高几个百分点。其实，如果一开始就有一个清晰的系统流程和最新的工具，你就可以在一两天之内处理一个从端到端的问题。将问题分解为多个小问题，然后一个一个的解决。

综上所述，通过正确的流程和方法，可以在最大程度上激发初学者的学习热情，迅速入门并不断提高。无论对于程序开发者还是自学的大学生，通过这种自上而下的系统流程可以全面的掌握机器学习算法，并随时运用最新的算法，有效的解决端到端的实际问题。

### 参考文献

[1]新浪科技,"http://tech.sina.com.cn/d/AlphaGo/."
[2]Coursera.https://www.coursera.org/learn/machine-learning.
[3]周志华,机器学习.清华大学出版社,2016.
[4]A.Smola,"https://www.quora.com/session/Alex-Smola-1/1."
[5]J.Brownlee,"http://machinelearningmastery.com/machine-learning-for-programmers/."
[6]WEKA.http://www.cs.waikato.ac.nz/ml/index.html.
[7]Python,"https://www.python.org/."
[8]Scikit-Learn,"http://scikit-learn.org/."
[9]IPython,"https://ipython.org/."
[10]R语言,"https://cran.r-project.org/."
[11]Caret,"http://topepo.github.io/caret/index.html."
[12]UCI数据库,"https://archive.ics.uci.edu/ml/index.html."
[13]Kaggle,"https://www.kaggle.com/."
[14]KDD比赛,"http://www.kdnuggets.com/datasets/kddcup.html."
[15]Youku优酷,"http://www.youku.com/."
[16]GitHub,"https://github.com/."
[17]EC2,https://aws.amazon.com/cn/.

作者简介：
罗晓牧（1980－），男，广东广州人，讲师，工科博士研究生毕业，研究方向：机器学习，无线传感器网络，生物信息获取。

（上接第31页）

学生的计算思维，符合中职学生发展的需求。教师可以通过明确教学目标，构建信息技术学习模型以及丰富教学内容，激发学生课堂学习兴趣等方式，关注学生思维的培养，为中职学生带来全新的信息技术课堂学习体验。

### 参考文献

[1]郭守超,周睿,邓常梅,等.基于AppInventor和计算思维的信息技术课堂教学研究[J].中国电化教育,2014,03(22):91-96.
[2]王旭卿.从计算思维到计算参与:美国中小学程序设计教学的社会化转向与启示[J].中国电化教育,2014,03(16):97-100.
[3]梁展锋,魏晓彤.基于微信公众号构建微课平台的探索与实践——以教师信息技术培训为例[J].中国现代教育装备,2016,02(14):59-61.
[4]张昭玉,任建平,吴勇,等.以计算思维为导向的《大学计算机基础》教学改革研究[J].现代计算机(专业版),2016,01(25):16-19,27.
[5]王明辉,龚彬.从实践中来到实践中去——浅谈苏科版信息技术教材2015年版的修订与特点[J].科学大众(科学教育),2015,09(27):179,41.