人工智能开发丛书

PMML
建模标准语言基础

潘风文　潘启儒　著

Predictive Model Markup Language，预测模型标记语言，是基于XML规范的开放式挖掘模型表达语言，为不同系统提供了定义数据挖掘模型的方法，已获得IBM、SAS、NCR、FICO、NIST、Tibco等绝大多数顶级商业公司和Weka、Tanagra、RapidMiner、KNIME、Orange、GGobi、JHepWork等开源挖掘系统的支持，正在快速普及。

化学工业出版社

人工智能开发丛书

PMML
建模标准语言基础

潘风文　潘启儒　著

化学工业出版社
·北京·

本书结合实际案例介绍了PMML语言的各个组成元素，包括数据字典、挖掘模式/架构、数据转换、模型定义、输出、目标、模型解释、模型验证等元素，并介绍了表述数据挖掘模型的PMML实例文档创建流程；同时也对各种PMML元素中涉及的一些统计知识做了必要介绍。通过学习，读者可以完整地了解和掌握PMML语言，将其应用于数据挖掘建模。

本书可供从事数据挖掘（机器学习）、人工智能系统开发的软件开发者和爱好者学习使用，也可以作为高等院校大数据等相关专业的教材。

图书在版编目（CIP）数据

PMML建模标准语言基础/潘风文，潘启儒著. —北京：化学工业出版社，2019.7
（人工智能开发丛书）
ISBN 978-7-122-34258-4

Ⅰ.①P… Ⅱ.①潘…②潘… Ⅲ.①检索语言-程序设计 Ⅳ.①TP312.8

中国版本图书馆CIP数据核字（2019）第063332号

责任编辑：潘新文　　装帧设计：韩　飞
责任校对：张雨彤

出版发行：化学工业出版社（北京市东城区青年湖南街13号　邮政编码100011）
印　　装：高教社（天津）印务有限公司
787mm×1092mm　1/16　印张19　字数427千字　　2019年8月北京第1版第1次印刷

购书咨询：010-64518888　　售后服务：010-64518899
网　　址：http://www.cip.com.cn
凡购买本书，如有缺损质量问题，本社销售中心负责调换。

定　　价：89.00元

前 言

数据挖掘技术起始于20世纪下半叶，当时伴随着计算机技术和数据库在各行各业的广泛应用，业务系统产生的数据量不断膨胀，传统的统计分析工具受到巨大的挑战，这促使科学家和研究人员把当时最新的数据分析技术（例如关联规则、神经网络、决策树等）与数据库技术结合起来，从而直接导致了数据挖掘技术的诞生。进入21世纪后，各行各业对数据价值的深入探索迅速推动了数据挖掘软件的应用，各种数据挖掘系统如雨后春笋般相继出现，比较著名的开发公司有IBM、SAS、NCR、Tibco等。

数据挖掘技术目前已经应用到几乎所有的行业，并取得了巨大的成功。但是不同的系统开发厂商都是基于各自的发展规划，使用自己的技术，推出的数据挖掘系统平台各具特色，从而导致数据挖掘模型不能在不同挖掘系统间共享，给数据挖掘的进一步普及和发展造成了障碍。

为了解决上述问题，实现数据挖掘模型的共享与交换，1997年，芝加哥伊利诺伊大学的Robert Lee Grossman博士发起设计了数据挖掘模型的开放标准——PMML（Predictive Model Markup Language，预测模型标记语言）它是一种基于XML（Extensible Markup Language，可扩展标记语言）规范的开放式挖掘模型表达语言，为不同系统提供了定义数据挖掘模型的方法，可使兼容PMML规范的应用程序共享模型。采用PMML语言，用户可在一个软件系统中创建预测模型，然后将其传递到另外一个系统，并在该系统中用PMML文档中的模型预测新数据，实现预测模型的跨语言、跨平台应用，提高可移植性，充分发挥挖掘模型的应用价值。

PMML语言基于XML，XML定义了一套对电子文档进行编码的规则，以人类和计算机都能够读懂的文本格式来表现文档，可以表达任意数据结构，是万维网联盟W3C（World Wide Web Consortium）的标准语言；XML是众多应用

型标记语言的基础，如化学领域的CML、数学领域的MathML以及本书介绍的PMML等。

一个完整有效的PMML实例文档包括数据字典、挖掘模式/架构、数据转换、模型定义、输出、目标、模型解释、模型验证等元素，PMML规范针对这些元素的声明和使用制定了模型创建者和模型使用者必须遵守的一致性规则，例如模型创建者通过何种方式生成何种分析模型，模型使用者通过何种方式使用何种分析模型等，这些一致性规则可以确保模型的输出在语法上是正确的，使所输出的模型符合PMML定义的语义标准，并确保模型使用者能够正确地部署和应用模型。本书主要基于以上要点讲述PMML规范以及PMML实例文档的结构和应用。

目前PMML已经发展到版本4.3，能够支持关联规则、聚类、回归、贝叶斯网络、神经网络、高斯过程等18种数据挖掘模型，涵盖了应用最广泛的常用模型。作为事实上的表达分析模型的标准，PMML已经被IBM、SAS、NCR、FICO、NIST、Tibco等绝大多数顶级商业公司所支持，也得到越来越多的开源挖掘系统如Weka、Tanagra、RapidMiner、KNIME、Orange、GGobi、JHepWork等的支持，目前其影响力越来越大。很多想学习PMML的人员苦于没有完整的学习资料，而网上的相关资料又比较零散琐碎，不成体系，为此我们结合多年来的实践和体会编写了本书，希望能在一定程度上助广大数据挖掘系统、人工智能系统开发者和使用者一臂之力，为深入学习PMML起到抛砖引玉的作用。

本书除了供数据挖掘（机器学习）、人工智能领域的软件开发人员使用外，也可以作为高等院校大数据等相关专业的教材或数据挖掘爱好者自学用书。

由于编写时间和编写精力有限，书中难免会有疏漏不当之处，敬请同行批评指正，多多提出宝贵意见和建议，共同进步。作者QQ：420165499。

编者

2019年3月

目 录

1 XML 基础

1.1 XML的发展、技术体系及应用

从广义上理解，语言是一套具有共同处理规则的用于表达思想、方法等的指令符号，它涵盖的范围较广，例如自然语言、计算机编程语言、工程图学语言、数学语言等等。XML（Extensible Markup Language，可扩展标记语言）是一种应用广泛的标记语言，它定义了一套对电子文档进行编码的规则，以人类和计算机都能够读懂的文本格式来描述文档，可以表达任意数据结构，是万维网联盟W3C（World Wide Web Consortium）的标准语言。设计XML语言的主要目标是在互联网上以简单、通用、便捷的方式交换和存储文档。XML也是众多应用标记语言的基础，如化学领域的CML、数学领域的MathML以及本书将重点介绍的PMML等。

1.1.1 标记语言和SGML

按照Wikipedia的定义，“标记语言（Markup Language）”又称为置标语言、标志语言、标识语言，是一种将文本及其他相关信息结合起来，展现文档结构和数据处理细节的计算机文字编码，通过标记文本以及相关信息（例如文本的组织结构、表现形式、呈现颜色等），实现相关内容的表达和传递。“Markup Language（标记语言）”一词引申自传统出版业中对原稿的“Markup（标记）”，即在原稿的边缘加注一些符号，指示排版格式以及打印要求，包括使用什么样的字型、字体以及字号等，然后将原稿交给排版人员进行排版。理论上讲可以有各种各样的标记语言，其中超文本标记语言HTML（HyperText Markup Language）和可扩展标记语言XML（Extensible Markup Language）被广泛应用于网络应用程序和网页中。

从XML语言的发展历史看，它是基于SGML（Standard Generalized Markup Language）发展起来的。SGML是一种通用的文档结构描述标记语言，也是定义其他标记语言的元语言，曾被用于编写牛津英语词典的电子版本。SGML的发展经历了通用编码（Generic Coding）、通用标记语言GML（Generalized Markup Language）、SGML标准化以及SGML应用四个重要阶段。

1）通用编码

大多数人把通用编码的起源归功于美国图形通信协会GCA（Graphic Communications Association）委员会主席William Tunnicliffe。1967年9月，在加拿大政府印刷局会议上，William Tunnicliffe做了题为“The Separation of the Information Content of Documents From Their Format”（文档信息内容与其格式的分离）的演讲，提出了对文本内容进行嵌入式格式化编码的思想。

20世纪60年代后期，纽约一位名叫Stanley Rice的书籍设计师提出了一个通用参数化“编辑结构”标签的设想，这是一个非常有创意的构思设计，GCA主任Norman

Walter Scharpf敏锐地捕捉到它的价值，很快他便提出了GenCode的概念，指出可通过创建各种不同的通用代码来表达不同类型的文档，较小的文档可以作为较大文档的元素，随后他在委员会中设立了一个通用编码项目组来实现这种设计，该项目组最终演变为GenCode委员会，在SGML标准制定中发挥了重要作用。

2）通用标记语言GML

1969年，IBM的Charles Goldfarb与Edward Mosher、Raymond Lorie共同推出了通用标记语言GML（Generalized Markup Language），GML基于Tunnicliffe和Rice的通用编码思想，但没有采用简单标记方案，而是引入了具有显式嵌套元素结构的文档定义类型概念。Goldfarb对文档的结构进行了深入的研究，提出了很多新概念，例如简短引用、链接过程、并发文档类型等，这些概念后来逐步成为SGML的一部分。

3）SGML标准化

1978年，美国国家标准协会ANSI（American National Standards Institute）信息处理委员会设立了计算机语言处理文本委员会，Goldfarb加入了该委员会，组织开发基于GML的文本描述语言标准项目，GCA的GenCode委员会也为这个项目做出了很大贡献。

SGML标准的第一份草案于1980年推出；1983年，GCA推出了SGML标准的第六份草案，并被作为行业标准（GCA101-1983），1986年此标准成为国际标准ISO 8879:1986 Information processing - Text and office systems - Standard Generalized Markup Language（SGML）。

4）SGML应用

SGML是一个具有较高稳定性和完整性的国际标准语言，其规范制定得相当细致严密，可满足不同应用领域使用者的需求，具有较好的可移植性（可携性），SGML文件可以跨平台使用；支持SGML格式的应用软件比较多，相关的数据转换技术也比较丰富；与SGML搭配使用的很多语言（如HyTime、DSSSL等）也都是国际标准语言。

早期的SGML多被应用于行业和企业组织内部的项目，如美国出版商协会AAP（the Association of American Publishers）的电子手稿项目、美国国防部计算机辅助采集和后勤保障计划CALS（the Computer-aided Acquisition and Logistic Support）的文档组件项目等，都采用了SGML。

不过SGML的使用比较复杂，例如美国出版商协会AAP的电子手稿项目，其技术工作由Aspen Systems公司承担，参与信息处理工作的组织超过了30个，包括IEEE、图书馆资源委员会、美国索引协会、美国国会图书馆、美国化学学会、美国物理学会、生物学编辑理事会和美国数学学会等。由于本身过于复杂，SGML最终没有被广泛普及，但是其设计理念非常先进，因此它成为各种标记语言的始祖，现在流行的各种标记语言全都是基于SGML派生的。

XML摒弃了SGML的复杂性，提高了易用性和开放性，因此很快得到普及，与其相关的应用有很多，例如XHTML、RSS、XML-RPC和SOAP等等；随着XML语言的

发展，在其基础上又衍生出一系列应用标准语言，如XHTML、SVG、SMIL、XBRL以及PMML（见图1-1），因此可以说XML是一种元标记语言，可以用来创建满足特定需求的专用标记语言。

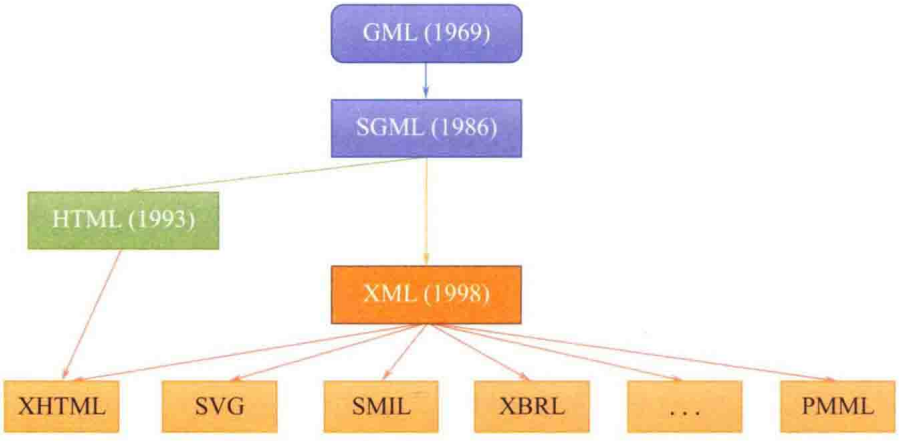

图1-1 标记语言的发展历史

1.1.2 XML的特点和应用

XML是由XML工作组（最初称为SGML编辑审查委员会）于1996年在万维网联盟W3C组织下开发出来的，最初XML工作组由Sun Microsystems的Jon Bosak主持，XML特殊兴趣小组（以前称为SGML工作组，由W3C组织）也积极参与了开发。

XML的设计目标是：

- XML可以直接在Internet上使用；
- XML应支持各种应用程序；
- XML应与SGML兼容；
- XML文档处理器的编写不需要很高深的技术；
- XML中的可选功能的数量应尽可能少，甚至为零；
- XML文档应易于理解并且相当清晰；
- XML应容易上手，使用快速便捷；
- XML设计应该正规而且简单；
- XML文档应易于创建；
- XML标记的简洁性不作为重点考虑因素。

经过多年的发展，XML语言已经非常成熟，它具有以下优点。

1）开放的标准

XML的开放性体现在它既与平台无关，又与技术提供厂商无关。W3C的XML工作组致力于维护XML的开放性，为开发人员在不同系统之间进行数据处理提供技术支持，

不断推进XML标准的发展。

2）文档内容和展示分离

XML把标记与展示分开，开发者可以在结构化数据中嵌入程序化的描述，以指明如何展示数据。

3）可自定义标记

XML不仅仅是一种标记语言，它还可以用来创建各种自描述性的标记——只要这种标记在相关领域得到认可。

4）良好的可读性和可维护性

XML文档包含文档类型声明，用来指定文档的结构、包含的元素及其意义，这样可使XML文档结构显得清晰，便于阅读和维护，并可以验证标记的定义和使用是否符合语法规则。

5）XML是各种技术的集成者

XML集数据验证、展示表达、文件转换、文档对象链接、组件选择等多种数据处理技术于一体，是各种技术的集成者。

XML主要应用领域如下。

◆ 数据交换　不同的应用系统可以按照基于XML的同一标准共享和解析数据，实现不同平台和系统间的无缝数据交换。基于Web服务的应用系统广泛使用XML文档进行数据传输。

◆ 内容管理　XML文档的内容和展示是分离的，其内容（数据）通过元素及其属性来描述，可通过扩展样式表语言XSL（Extensible Stylesheet Language，XSL文档也是一种XML文档，遵循XML的所有规范）转换成各种格式的文件，如HTML、PDF、CSV等，以进行展示。

◆ 系统配置　系统配置管理是每个应用系统必备的功能。XML文档的结构化、易用性优点使它被很多系统用来进行系统配置，各种Web服务器（如Tomcat、JBoss等）都采用XML文件作为系统参数配置文件。

◆ 创建新的标记语言　XML可以用来创建标记语言，目前有很多标记语言是基于XML创建的，例如MusicML、MathML、CML、SVG、WML、SMIL和PMML等。

实际上XML技术的应用远远不止这些，随着各种相关技术的日益成熟，XML在各个行业都开始得到广泛应用。

1.1.3 XML技术体系

XML目前最新版本为第5版，XML的官方网址为：https://www.w3.org/TR/xml/，可

以通过官方网站了解XML的基本语法规范以及用XML设计各种应用标准语言的方法和规则等。

图1-2所示是XML家族技术体系，其底层是XML的核心，包括XSD（XML Schema Definition，也称XML Schema）、Namespace、DTD（XML Document Type Definition）。XML Schema用于定义和描述XML文档结构、内容模式、元素之间的关系以及元素和属性的数据类型，为XML文档的处理提供基础，XML Schema于2001年5月成为W3C的正式标准，官方网址：https://www.w3.org/XML/Schema。XML Namespace提供了对XML文档中的元素和属性进行统一命名的机制，以避免不同标记词汇表的元素和属性的命名冲突。1999年1月14日XML Namespace成为W3C的推荐规范。官方网址：https://www.w3.org/TR/REC-xml-names/。DTD源于SGML，采用了非XML的语法规则，仅支持少量的数据类型，扩展性比较差，已经逐步被XML Schema所代替，因此本书不对DTD做详细介绍。中间一层是所支持的相关规范和工具，最上层是针对某一具体行业或领域的XML应用。

图1-2 XML家族技术体系

下面先简要介绍其中的几个主要部分。

1）XML Schema

为了便于说明XML Schema，下面先看一个XML DTD文档：

```
1.  <!DOCTYPE CATALOG [
2.
3.  <!ENTITY AUTHOR "John Doe">
4.  <!ENTITY COMPANY "JD Power Tools, Inc.">
5.  <!ENTITY EMAIL "jd@jd-tools.com">
6.
7.  <!ELEMENT CATALOG (PRODUCT+)>
```

```
8.
9.  <!ELEMENT PRODUCT
10. (SPECIFICATIONS+,OPTIONS?,PRICE+,NOTES?)>
11. <!ATTLIST PRODUCT
12. NAME CDATA #IMPLIED
13. CATEGORY (HandTool|Table|Shop-Professional) "HandTool"
14. PARTNUM CDATA #IMPLIED
15. PLANT (Pittsburgh|Milwaukee|Chicago) "Chicago"
16. INVENTORY (InStock|Backordered|Discontinued) "InStock">
17.
18. <!ELEMENT SPECIFICATIONS (#PCDATA)>
19. <!ATTLIST SPECIFICATIONS
20. WEIGHT CDATA #IMPLIED
21. POWER CDATA #IMPLIED>
22.
23. <!ELEMENT OPTIONS (#PCDATA)>
24. <!ATTLIST OPTIONS
25. FINISH (Metal|Polished|Matte) "Matte"
26. ADAPTER (Included|Optional|NotApplicable) "Included"
27. CASE (HardShell|Soft|NotApplicable) "HardShell">
28.
29. <!ELEMENT PRICE (#PCDATA)>
30. <!ATTLIST PRICE
31. MSRP CDATA #IMPLIED
32. WHOLESALE CDATA #IMPLIED
33. STREET CDATA #IMPLIED
34. SHIPPING CDATA #IMPLIED>
35.
36. <!ELEMENT NOTES (#PCDATA)>
37.
38. ]>
```

这个DTD文档摘自网站http://www.vervet.com/，它定义了一个产品目录，可以看出，这个DTD文档由不同的标签组成，这些标签用来规划一个XML文档的结构。由于DTD文档不是一个XML文档，可扩展性差，并且不支持元素的数据类型，对属性的类型定义也有限，因此DTD最终被更规范、更开放的XML Schema取代。XML Schema支持命名空间（Namespace）机制，支持整体验证和局部验证，而这都是DTD所没有的。下面是一个简单的XML Schema文档：

```
1. <?xml version="1.0" encoding="UTF-8" ?>
2. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3.
4. <xs:element name="shiporder">
5.   <xs:complexType>
6.     <xs:sequence>
7.       <xs:element name="orderperson" type="xs:string"/>
8.       <xs:element name="shipto">
9.         <xs:complexType>
10.          <xs:sequence>
11.            <xs:element name="name" type="xs:string"/>
12.            <xs:element name="address" type="xs:string"/>
13.            <xs:element name="city" type="xs:string"/>
14.            <xs:element name="country" type="xs:string"/>
15.          </xs:sequence>
16.        </xs:complexType>
17.      </xs:element>
18.      <xs:element name="item" maxOccurs="unbounded">
19.        <xs:complexType>
20.          <xs:sequence>
21.            <xs:element name="title" type="xs:string"/>
22.            <xs:element name="note" type="xs:string" minOccurs="0"/>
23.            <xs:element name="quantity" type="xs:positiveInteger"/>
24.            <xs:element name="price" type="xs:decimal"/>
25.          </xs:sequence>
26.        </xs:complexType>
27.      </xs:element>
28.    </xs:sequence>
```