

密级：_____

浙江大学

硕士学位论文



论文题目 数据智能可视化系统中图形透视表
配置的生成与推荐

作者姓名 胡凡

指导教师 胡天磊

学科(专业) 计算机科学与技术

所在学院 计算机科学与技术学院

提交日期 2017 年 1 月 4 日

A Dissertation Submitted to Zhejiang
University for the Degree of
Master of Engineering



TITLE: Generation and
Recommendation of Graphical Pivot
Table Configuration in Intelligent
Data Visualization System

Author: Fan Hu

Supervisor: Tianlei Hu

Subject: Computer Science and Technology

College: Computer Science and Technology

Submitted Date: 2017-01-04

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 浙江大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：

签字日期：

年 月 日

学位论文版权使用授权书

本学位论文作者完全了解 浙江大学 有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 浙江大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：

导师签名：

签字日期： 年 月 日

签字日期： 年 月 日

学位论文作者毕业后去向：

工作单位：

电话：

通讯地址：

邮编

摘要

大数据时代中，每一条数据都蕴含巨大的价值，但是很少有企业意识到可以通过数据可视化将这些数据转换为实际的经济价值，而这很大程度上是因为当前的许多数据可视化系统表达能力较弱，同时也缺乏一些智能性，使决策者无法及时作出正确的决策。

本文设计并实现了一个数据智能可视化系统，解决了两个最核心的技术难题：1) 具有高可扩展性的图形透视表配置的生成；2) 如何给用户推荐更有价值的图形透视表配置，而这又包括“下一步怎么看”和“应该怎么开始看”这两个重要问题。

针对图形透视表配置的生成，本文提出了新的表代数算子，并用其生成透视结构配置，然后基于改进的图形语言来生成图形设计，并归纳了图形透视表配置的显式信息与隐式信息。而在智能推荐的问题上，本文总结并设计了基于数据特征的标记类型推导规则；提出了数据特征组合的三原则，并结合先验知识设计了启发式算法解决了单字段配置问题；设计了基于优先原则的多字段图表类型优先级算法，提出了基于图形语言的图形透视表配置推荐算法。

本文设计实现的系统已经作为商业智能可视化平台网易有数的一个模块，应用在网易多款产品中，为运营决策提供了能够进行数据探索的智能可视化工具。

关键词：数据可视化，表代数，图形语言，智能推荐

Abstract

In the era of Big Data, each data contains great value, but few companies realize that they can obtain economic benefits through data visualization, which is largely due to the fact that the ability of many current data visualization systems is a little weak, but also the lack of some intelligence, so that decision makers can not make timely decisions.

This paper designs and implements an intelligent data visualization system, which solves the two most difficult technical problems: 1) how to design a high-scalability system to generate the graph pivot table configuration; 2) how to recommend valuable graphical pivot table configuration, which in turn includes the two important issues "how to see the next" and "how to see in the beginning".

In this paper, a new table algebra operator is proposed, which is used to generate the pivot configuration of a graph pivot table. Then, the graphic design can be generated based on the improved graphical language. Besides, this paper summarizes the explicit information and the implicit information of the graph pivot table configuration. On the question of intelligent recommendation, this paper summarizes and designs the derivation rules of different mark types based on data features. Then, three principles of data feature combination are proposed, and a heuristic algorithm is designed to solve the problem of single-field configuration. Finally, based on the priority principle of multi-field chart type priority problem, a graphical language based recommendation algorithm for graph pivot table configuration is proposed.

The system designed and implemented in this paper has been used as a module in Netease Youdata, which is a business intelligence visualization platform, and it is used in a variety of Netease products for decision-making to make great contribution.

Keywords : data visualization, table algebra, graphical language, intelligent recommendation

目录

摘要.....	i
Abstract.....	ii
目录.....	I
图目录.....	IV
表目录.....	VI
第 1 章 绪论.....	1
1.1 课题背景及意义.....	1
1.2 国内外发展现状.....	2
1.2.1 国内发展现状.....	2
1.2.2 国外发展现状.....	3
1.3 本文的工作和贡献.....	4
1.4 本文组织.....	5
1.5 本章小结.....	6
第 2 章 相关技术分析.....	7
2.1 高维数据可视化研究.....	7
2.1.1 散点图矩阵.....	7
2.1.2 表格透镜.....	8
2.1.3 数据透视表.....	9
2.1.4 表代数.....	10
2.2 可视化编码.....	11
2.2.1 数据特征.....	11
2.2.2 标记与视觉通道.....	11
2.3 图形符号学发展历程.....	12
2.3.1 图形符号学.....	12
2.3.2 图形语言.....	13
2.4 智能可视化进展.....	14
2.5 本章小结.....	15
第 3 章 系统架构.....	16
3.1 基本概念描述.....	16
3.2 系统工作流程.....	18
3.2.1 图形透视表配置的生成.....	18
3.2.2 单字段配置推荐.....	20

3.2.3 多字段配置推荐.....	21
3.3 系统架构设计.....	22
3.3.1 流程控制模块.....	23
3.3.2 数据获取模块.....	23
3.3.3 透视结构计算模块.....	24
3.3.4 图形设计生成模块.....	24
3.3.5 图形透视表配置解释模块.....	24
3.3.6 标记类型推导模块.....	25
3.3.7 单字段配置推荐模块.....	25
3.3.8 多字段配置推荐模块.....	25
3.4 接口设计.....	26
3.5 本章小结.....	28
第4章 图形透视表配置的生成.....	29
4.1 透视结构配置的生成.....	29
4.1.1 Table Field 与 Pane Field.....	29
4.1.2 透视结构配置生成与数据切分算法.....	31
4.2 基于图形语言的图形设计生成.....	37
4.2.1 图形语句生成算法.....	37
4.2.2 基于图形语言的图形设计转换算法.....	44
4.3 图形透视表配置的描述解释.....	45
4.3.1 显式信息的描述解释.....	45
4.3.2 隐式信息的描述解释.....	46
4.4 本章小结.....	47
第5章 图形透视表配置的推荐.....	48
5.1 标记类型的推荐.....	48
5.1.1 标记类型的使用场景.....	49
5.1.2 基于数据特征的标记类型推荐.....	51
5.2 图形透视表的单字段配置推荐.....	56
5.2.1 启发式配置原理.....	57
5.2.2 单字段配置的启发式推荐算法.....	59
5.3 图形透视表的多字段配置推荐.....	62
5.3.1 图表类型的必要条件.....	63
5.3.2 图表类型的优先级规则.....	69
5.3.3 基于图形语言的图形透视表配置推荐算法.....	72
5.4 本章小结.....	75

第 6 章 系统应用与实例展示.....	76
6.1 系统应用.....	76
6.1.1 网易有数系统架构.....	76
6.1.2 网易有数交互界面.....	78
6.2 实例展示.....	79
6.2.1 数据描述.....	79
6.2.2 实例展示.....	79
6.3 本章小结.....	92
第 7 章 总结与展望.....	93
7.1 本文的工作和贡献.....	93
7.2 未来的研究展望.....	94
参考文献.....	95
攻读硕士学位期间主要的研究成果.....	99
致谢.....	100

图目录

图 3-1 Pane-Cell 行列字段示意图	16
图 3-2 Pane-Cell 示意图	17
图 3-3 图形透视表配置生成的工作流程	19
图 3-4 单字段配置推荐的工作流程	20
图 3-5 多字段配置推荐的工作流程	21
图 3-6 模块划分示意图	23
图 4-1 Table Field 与 Pane Field 行列配置	30
图 4-2 Table Field 与 Pane Field 示意图	31
图 4-3 Query Route 对应的透视表	32
图 4-4 Pane Tree 行列配置	34
图 4-5 Pane Tree 示意图	34
图 5-1 图形透视表推荐算法通用算法流程	73
图 6-1 网易有数系统架构图	77
图 6-2 网易有数交互界面	78
图 6-3 透视结构生成实例的行列配置	80
图 6-4 X Pane Tree	81
图 6-5 Y Pane Tree	81
图 6-6 图形设计实例的行列配置	82
图 6-7 图形设计实例的标记信息配置	82
图 6-8 图形设计实例的 Cell 1	83
图 6-9 图形设计实例的 Cell 2	84
图 6-10 图形设计实例的合并 Cell 图	84
图 6-11 图形设计实例的图形语句树	85
图 6-12 图形透视表配置的描述解释示意图	86
图 6-13 标记类型推荐实例 1	87
图 6-14 标记类型推荐实例 2	87
图 6-15 单字段推荐实例的配置 1	88
图 6-16 单字段推荐实例结果图 1	88
图 6-17 单字段推荐实例的配置 2	89
图 6-18 单字段推荐实例结果图 2	89
图 6-19 多字段推荐实例的字段	90
图 6-20 多字段推荐实例结果图 1	91

图 6-21 多字段推荐实例结果图 2.....	92
--------------------------	----

表目录

表 2-1 表格示意图..... 8

表 2-2 表格透镜示意图 9

表 2-3 数据透视表示意图 10

表 5-1 无轴标记类型推荐表 52

表 5-2 单轴标记类型推荐表 54

表 5-3 双轴标记类型推荐表 56

表 5-4 X 方向和 Y 方向均不存在字段的推荐逻辑 60

表 5-5 X 方向和 Y 方向只有一个存在字段的推荐逻辑 61

表 5-6 X 方向和 Y 方向均存在字段推荐逻辑..... 62

表 5-7 图表类型必要条件 68

表 5-8 图表类型的优先级规则 71

表 5-9 视觉信息优先级 72

第1章 绪论

1.1 课题背景及意义

大数据时代已经来临，数字金矿正等待人们挖掘和使用。然而由于信息规律自身的隐蔽性，直接观察冰冷的数据并不能很好地获取到真正有价值的信息。基于此，且考虑到人类强大的视觉能力，数据可视化的课题正在被更快速地摆到荧幕前，以帮助决策者更好地决策，并产生更高的经济价值。但数据可视化领域同样也面临着许多挑战^[1]。

然而，虽然数据可视化领域正蓬勃发展，也出现了一些优秀的可视化系统拓荒者，例如 Polaris^{[2][3]}，但是数据可视化辅助决策的程度依旧还没有达到很高的水准。此问题由三方面导致：一，现有的许多数据可视化系统缺乏易用性，即决策者仍然需要借助数据分析师进行报表绘制，而无法很好地直接上手进行数据探索；二，一些数据可视化系统的可扩展性较低，即由于系统本身的局限性导致用户无法进行非常自由的数据探索；三，用户自身不知道什么样的可视化图表可以让他们获取到更重要的信息，而数据可视化系统也没有办法给出有价值、有信息量的可视化图表的推荐。基于这三个方面，当前数据可视化带给人们的价值依然处于一个较低的水平。

为了解决上面的问题，本文设计并实现了数据智能可视化系统中的图形透视表配置的生成和推荐的技术。在该技术中，图形透视表的配置可以自动化生成，也可以对图形透视表的配置给出文字性的描述解释；同时，该技术也可以给用户进行图形透视表配置的推荐，以使用户能获取到更有价值的信息。

本文的图形透视表配置的生成技术以一种自动化的方式，使用户可以不用在

意底层的实现，而把更多的时间放在数据探索上，且对图形透视表配置的文字性描述解释可以让用户对图表的内容有更充分的认识和理解。而这有利于减轻数据分析师的学习和使用成本，或是让决策者能够直接上手进行数据探索。

另一方面，本文的图形透视表配置推荐技术可以让用户在对数据了解较少的情况下，进行更智能的数据关系的挖掘，这对数据探索和价值获取来说具有非常大的价值。一旦数据可视化系统可以自动挖掘出潜在的数据特征关系，就可以辅助决策者进行更好的决策，以此获取更大的经济利益。

1.2 国内外发展现状

1.2.1 国内发展现状

当前国内的数据可视化系统还处于萌芽阶段，甚至还没有出现拥有智能推荐能力的数据可视化系统，但也不乏一些优秀的拓荒者，如 FineBI^[4]、大数据魔镜^[5]、数说立方^[6]、数加^[7]、BDP^[8]等。国内的数据可视化现状需要突破，既有机遇又是挑战^{[9][10]}。

FineBI 是帆软公司推出的数据可视化平台，其对多种数据源都有很好的支持，既可以进行数据展示，也可以进行数据分析，图表的风格也十分雅致。它有一套针对企业的管理平台，在这个平台上可以实现模板、用户和权限等需求的集中管理。不过其可视化的布局和交互较为僵硬，且操作交互逻辑不够友好，使易用性有所下降。

大数据魔镜作为一款基于 ECharts 可视化绘制引擎的数据可视化系统，其图表非常酷炫。其产品模块的规划较为完整，拥有基础企业版到 Hadoop 等共五种版本供用户选择，且定制化功能非常不错。但是其对大数据的性能问题比较突出，

无法对大数据作出快速响应，在海量数据下的绘制效率较为低下。

数说立方是数说故事公司所研发的一款 Web 版商业智能产品，其主要面向对象是数据分析师，这使它的易用性受到一定限制，可能需要较为专业的人员才能很好使用。其性能非常出色，即便对百亿级数据也能做到快速响应。除此之外，其内搭载了数据可视化、语义分析、分布式搜索这三大引擎系统的海量计算平台，与其公司的另外几款产品构成了较为完整的数据解决方案。

数加是阿里云推出的一款大数据平台，其依托阿里云强大的分布式平台，通过对数据进行深度整合、计算和挖掘，将处理得到的结果通过其可视化系统进行数据展现。同时，数加平台还拥有机器学习和数据挖掘平台，虽然目前的功能还有限，但也可以给用户提供一个很好的数据分析工具。比较遗憾的是其部分操作不够友好，有一定使用门槛，且需要捆绑阿里云账户才可以使用，这使得产品没办法很容易推广。

BDP 是一款上手容易的数据可视化平台，界面十分简洁清新。其拥有较多实用图表的绘制能力，并使用维度分析为其主要交互手段，可以完成一些简单的数据探索。不过遗憾的是，它的拓展能力较弱，无法生成较复杂的图表，也没有智能推荐的功能，因此无法更深层次地进行数据探索。

1.2.2 国外发展现状

国外的数据可视化领域发展迅速，科研与工程结合紧密，产生了 Tableau^[11] 这样优秀的的数据可视化系统。除此之外，还有 Power BI^[12]、QlikView^[13] 等后起之秀在追赶，数据可视化在企业决策过程中的地位正在逐渐提高。

Tableau 是一款优秀的的数据可视化系统，目前处于数据可视化行业的领导者地位，它既有着较高的专业性，又对新手有较好的引导性。Tableau 采用拖拽式

交互，操作简单，且支持多种数据文件和数据库，对大数据的支持也较为完好。

Power BI 是微软推出的数据可视化系统，其易用性非常高，只需要会使用 Excel 就能很好地上手绘制报表。其可扩展性也较为不错，用户甚至可以定制化地制作新的图表类型，满足了不同领域的数据可视化要求。

QlikView 是 Qlik 公司的一款数据可视化产品，用户可以很轻易地对多个数据源进行合并、搜索、可视化和分析的工作。其支持的图表类型较为丰富，交互功能也可圈可点。但是其有一定的学习成本，对报表的规范性要求也比较高。

1.3 本文的工作和贡献

本文设计并实现了一个数据智能可视化系统，主要功能是对数据库中用户感兴趣的数据生成图形透视表的配置，并对其进行自动化文字描述解释；同时在用户对数据了解较少的情况下提供标记类型和图形透视表配置的推荐功能，极大地辅助了用户进行数据探索的过程。

本文的主要工作和贡献如下：

(1) 本文提出了新的表代数算子，并基于改进的表代数表达式进行透视结构配置的生成与数据集切分的工作。

(2) 本文改进了图形语言的合并算子，并基于改进后的图形语言生成了图形透视表对应的图形语句，然后将其转换为最终的图形设计。

(3) 本文将图形透视表配置中的信息归类为显性信息和隐性信息，以对图形透视表的配置进行文字性描述解释，使用户能对图表有直观的认识。

(4) 本文基于数据特征设计并实现了图表标记类型的推荐规则，用户将可以用尽可能合适的图表标记类型来获取更直观的视觉信息。

(5) 本文提出了数据特征组合的三原则，并在此基础上设计了启发式算法，以在当前已存在的可视化局面下，对新字段的配置进行启发式推荐，解决了“下一步应该怎么看”的问题。

(6) 本文讨论并总结了图表类型的必要条件，并基于优先原则与一些先验知识设计了图表类型的优先级规则，最后设计并实现了基于图形语言的图形透视表多字段推荐算法。于是用户在给定多个感兴趣的字段的情况下，可以获知适用于展示这些字段的图表类型的优先级，并针对每种推荐的图表类型，系统将给出相应的图形透视表配置，解决了“应该怎么开始看”的问题。

(7) 本文使用 Superstore 数据集对系统中的几个关键功能进行测试和说明，验证了本文系统能够正确生成图形透视表的配置，并在需要时可以推荐最合适的图形透视表配置。

(8) 本文的图表推导和智能推荐模块已经被网易有数产品使用，并将其用于网易内部和外部若干产品的数据可视化分析需求，提高了业务效率。

1.4 本文组织

本文一共分为七个章节，各章节的主要内容如下：

第一章详细说明了本文的课题背景和研究意义，叙述了国内外数据可视化产品系统的发展现状，并阐明了本文所设计实现的数据智能可视化系统的工作和贡献。

第二章对本文涉及到的前人相关工作进行汇总和总结，本文所实现系统的工作和贡献将基于这些前人的工作成果。

第三章给出了本文所设计实现的数据智能可视化系统的架构，包括系统工作流程设计、系统架构设计、接口设计等。

第四章设计并实现了图形透视表配置的生成系统，其中提出了新的表代数算子，并用改进后的表代数表达式生成了透视结构；改进了图形语言中的合并算子，并基于改进后的图形语言生成图形语句，然后将其转换为图形设计；将图形透视表的信息归类为显性信息和隐性信息，并生成文字来对图形透视表的配置进行描述解释。

第五章设计并实现了图形透视表配置的推荐系统，包括基于数据特征的图表标记类型的推荐、基于数据特征组合原则的单字段启发式配置算法、针对多字段的图表类型的必要条件设计与优先级计算、基于图形语言的图形透视表配置的推荐。

第六章介绍了本文系统所接入的网易有数项目的架构设计和交互界面，并使用 Superstore 数据集对本文工作进行解释。

第七章总结了本文的工作和贡献，并对未来的工作和发展进行了思考和展望。

1.5 本章小结

本章对本文的课题背景和研究意义进行了说明，介绍了国内外数据可视化产品系统的发展现状，阐述了本文所设计实现的数据智能可视化系统的工作和贡献，并给出了本文章节内容的组织。

第2章 相关技术分析

2.1 高维数据可视化研究

这个世界上的数据有非常多的形式，例如可以用 Story、Event Graph、Graph 作为模型来处理时序数据和异构数据^{[14][15][16]}。一般来说，人类对二维和三维图形有较好的感知和理解能力，而对更高维的图形则没有办法进行很好地理解。为了解决这个问题，常见的方法是在二维和三维图形的基础上，使用颜色、尺寸、形状等视觉属性来额外绑定高维信息。但是这样做当维度特别高时会使得可视化图形的可读性大大下降，此时必须通过另外的手段来对高维数据^[17]进行降维，使在二维或三维的场景下进行观察。下面介绍高维多元数据的两种降维方法，即散点图矩阵与表格透镜，其中前者将应用在本文图形设计启发式推荐中，而后者在改进后将作为本文高维数据降维的主要方法。

2.1.1 散点图矩阵

散点图是指将抽象的数据对象映射到二维的直角坐标系表示的空间中，这样数据对象在坐标系的位置就可以很好体现数据分布特征，由此很容易可以观察到两个属性之间的关系。

既然散点图体现的是两个属性之间的关系，那么如果有 N 个属性，就可以用散点图矩阵^[18]来挖掘每两个属性之间的关系。对这 N 个属性，画出 N 行 N 列的散点图矩阵，其中第 i 行和第 j 列所对应的散点图展示了第 i 个属性和第 j 个属性之间的关系。显然，当属性个数不太大时，这种做法可以直观地展示属性之间的关系；但是当属性个数非常多时，会使得散点图矩阵的规模过大，导致可视化的

可读性下降。

















2.1.2 表格透镜

表格来是展现高维多元数据的一种方式，其每一列表示一个数据属性，每一行是一条数据记录，于是每个单元格内就是一个数值，表示当前行的数据记录属于当前列的数据属性的具体数值，表 2-1 是一个例子。表格透镜^[19]采用了与表格相同的布局，但是其单元格内可以不是数值，而是代表该数值的一根水平横条或一个点，这样数值的大小对比信息就可以从多个单元格的横条长度对比或点的相对位置来获得，相比表格来说更加直观，可读性更好，表 2-2 是一个例子。

表 2-1 表格示意图

国家	地区	客户类别	产品类别	销售额	利润
中国	东北	企业	办公	32768	3567
中国	东北	企业	技术	46902	4873
中国	东北	消费者	办公	69825	5326
中国	东北	消费者	技术	92743	8145
中国	西南	企业	办公	10256	1476
中国	西南	企业	技术	25748	2274
中国	西南	消费者	办公	22793	2898
中国	西南	消费者	技术	54788	6073

表 2-2 表格透镜示意图

国家	地区	客户类别	产品类别	销售额	利润
中国	东北	企业	办公		
中国	东北	企业	技术		
中国	东北	消费者	办公		
中国	东北	消费者	技术		
中国	西南	企业	办公		
中国	西南	企业	技术		
中国	西南	消费者	办公		
中国	西南	消费者	技术		

2.1.3 数据透视表

由于表格的每行是数据记录，每列是数据属性，因此表格适合于单纯地展示数据，而当需要对表格中的数据进行分组统计的时候，狭义的表格就无法提供相应的支持了。此时可以对表格的概念进行扩展，让它的行与列都存放部分数据属性，并且行与列的数据属性可以各自进行嵌套，形成透视结构，统计信息则写于单元格内。表 2-3 是一个例子。

对一个普通的数据透视表来说，每个单元格都是一个数值，如果采用表格透镜的方式，就可以把这个数值用水平横条或点代替。但仍可以对此进行更进一步的扩展，也就是在每个单元格里嵌入一个复杂图形，例如散点图、柱状图，甚至是散点图矩阵，以达到更好地进行数据探索的目的。

表 2-3 数据透视表示意图

		客户类别	
国家	地区	企业	消费者
中国	东北	32768	46902
		69825	92743
	西南	10256	25748
		22793	54788

2.1.4 表代数

表代数是 Stolte 提出的用来解决数据透视表透视布局计算的形式化框架^[32]。在表代数下，数据将分为离散型和连续型，并根据表代数的三个基本算子来对透视布局进行求值计算。

对离散型数据，表代数在对其求值时，会按它的所有成员进行展开；而对连续型数据，表代数会保留其本身。假设对离散型数据 D 来说，其有三个成员 $D1$ 、 $D2$ 、 $D3$ ，那么对其求值后就会得到 $VAL(D) = \{(D1), (D2), (D3)\}$ ；而如果是连续型数据 C ，那么对其求值后仍然是其本身，即 $VAL(C) = \{(C)\}$ 。

表代数有三个算子： $+$ 、 $*$ 、 $/$ ，其中 $+$ 算子又称为连接算子，用以将算子左右两边的表代数表达式成员直接拼接在一起； $*$ 算子又称为笛卡尔积算子，用以将算子左右两边的表代数表达式成员作笛卡尔积； $/$ 算子又称为嵌算子，是在 $*$ 算子的基础上，排除掉不存在于数据表中的成员后的结果。Stolte 的论文中给出了在不同数据组合下的算子计算实例。

2.2 可视化编码

2.2.1 数据特征

在可视化中，很多讨论都会直接依赖于数据特征。各个系统可能采用不同的数据特征，例如 Polaris 将数据分为序列型 (Ordinal)、数值型 (Quantitative)，而 SAGE 将数据从多个角度进行分类^{[27][28]}，其中较重要的是数据角色 (Role)、数据阐释 (Interpretation)、数据类型 (DataType)，下面对这三个分类依据进行解释，本文系统将使用这种数据特征。

(1) 数据角色 (Role)：数据有类似自变量与因变量的角色。如果一个字段是独立的、不依赖于其他字段而存在，那么就称它的角色是维度 (Dimension)；而如果一个字段是通过其他字段经过一些操作（例如 SUM 操作等）产生的，是依赖于其他字段而存在的，那么就称这个字段的角色是度量 (Measure)，意为对其他字段的量化、衡量。

(2) 数据阐释 (Interpretation)：对一个数据字段来说，它总可以在某个角度下认为它是离散 (Discrete) 的，或是在另一个角度下认为它是连续 (Continuous) 的。例如字段中存在 1 到 10 这 10 个数，那么在对这个字段进行阐释时，既可以认为是一个离散字段，也可以认为是一个连续字段。

(3) 数据类型 (DataType)：一个数据字段的形式或属性含义称为数据类型。对一个字段来说，它的数据类型可能是整数、小数、字符串、日期、时间、地理等。

2.2.2 标记与视觉通道

在数据可视化的问题上，有一个很重要的内容就是可视化编码 (Visual

Encoding)。可视化编码的过程就是把数据映射为具体的图形元素的过程，因此在这种技术下，往往可以让数据更直观地表达在用户面前。如果说我们把数据拆分成属性（Property）和值（Value），那么可以想象，可视化编码的过程就是把属性和值分别映射为可视化信息，其中把数据属性映射为可视化元素（称为标记，即 Mark），而把数据的值映射为标记的视觉属性。在这两个映射过程下，数据信息将被很好地可视化展现出来。

可以想象，使用几何图形作为标记来对数据进行描述是一种很合理的可视化方法。而在人们的生活中，较为常见的有点、线、面、体，其中体在本文系统中不使用。对这些标记来说，可以用视觉通道来定量地定义标记的状态，例如可以用位置来表示标记的时空分布，而使用颜色、大小、形状等视觉属性来对额外的信息进行编码。一般来说，视觉通道的选择需要根据不同的需求来选择。

2.3 图形符号学发展历程

2.3.1 图形符号学

图形符号学是指用符号来描述图形，而这起源于1967年 Jacques Bertin 出版的书籍《Semiology of Graphics》^[20]（《图形符号学》）。信息的可视化编码原则就是在这本书中首次提出。Bertin 将图形系统加以区分，将其分割为内容和载体，其中前者表示图形系统所要表达的信息，而后者则表示具体的图形符号。

显然，此时可以使用各种不同的图形符号来对信息进行描述，例如常见的点、线、面、体就是图形符号。一般来说，对这些图形符号可以使用视觉变量来提供更多信息，其中视觉变量由位置（Position）变量和视网膜（Retinal）变量组

成。从名称中即可知道，位置变量指出了图形在二维或者三维平面上的具体位置，而视网膜变量则对图形的视觉属性（例如颜色、尺寸、形状等）进行了编码。

2.3.2 图形语言

1986年，Mackinlay提出了一种将数据库中的字段信息提取出并表达的可视化表达方法^{[21][22]}，其中可视化表达是指通过生成一种图形设计（Graphical Design），来表达数据与图形结构之间的关系。图形语言定义为图形语句的集合，每个图形都由满足图形语言的一个图形语句（Graphical sentence）表示。一个图形语句 s 由一系列的元组组成： $s \subset \{ \langle o, l \rangle : o \in O \wedge l \in L \}$ ，其中 O 代表一组图形对象， L 代表位置，每个元组代表一个在具体位置的图形对象。简单的图形由简单的图形语句表示，复杂的图形则由复杂的图形语句表示。

在Mackinlay的论文中，他认为图形语言的表达式需要两个性质，即可表达性和有效性。其中可表达性原则的含义为图形能够保证表达数据，而有效性的含义为图形应当尽可能更好地表达数据。

在此基础上，Mackinlay进一步设计了一套基本图形语句、以及三个组合算子。其中三个组合算子用以合并两个有着兼容信息的图形语句，三个组合算子的具体含义如下。

- （1）双轴合并：合并的图形语句具有相同的横轴和纵轴。
- （2）单轴合并：对齐图形语句中相同的横轴和纵轴。
- （3）标记合并：对齐图形语句中的标记。

Mackinlay依照这套图形语言设计了APT系统，成为了智能可视化领域中划时代的系统。在此基础上，Wilkinson提出了图形语法^[36]（Grammar of Graphics），而这应用在了GPL（Graphics Production Language）语言中，成为了SPSS使用

的语言^[37]。在这之后，改进后的图形语法应用在了 R 语言中的 ggplot2 中^{[38][39]}，也出现了很多基于图形语言的可视化应用^{[40][41][42]}。

本文 4.2 小节将在 Mackinlay 的工作基础上，改进其对于合并算子的相关操作，使其对本系统更具实用性。

2.4 智能可视化进展

智能可视化是指通过一定手段，对用户观看数据的方式作出辅助推荐的过程。这可以包括两点：1) 自动化地生成图表结构；2) 推荐用户合适的可视化方式。由于用户本身思维和信息的局限性，以及缺乏对可视化领域知识的缺乏，导致他们并不能总是使用最好的可视化效果来进行相应的可视化过程。智能可视化试图通过各种方式，帮助用户进行更好的数据可视化工作。

当前已有一些智能可视化的探索。例如 Mackinlay 使用 2.3.2 小节中介绍的图形语言实现了 APT 系统，其中就有智能的意味在。APT 中提出了两项原则表达性原则和有效性原则，其中表达性原则的含义为图形能够保证表达数据，而有效性的含义为图形应当尽可能更好地表达数据。在其论文中介绍了表达性原则和有效性原则的相关描述^{[21][22]}，根据这两个原则，APT 将 MRS 系统^[23]的逻辑编程思想作为具体实现的工具，不过 MRS 已经被淘汰，进化出的新工具中较为优秀的则是 Prolog 与 miniKanren^{[24][25][26]}。APT 构建了一个智能专家系统，由用户的数据输入来推导出合适的图形设计，是智能可视化探索道路上的一个关键点。

除了 APT 以外，还有一些智能可视化方向的探索者，例如 SAGE 使用了与 APT 系统不同的方式来推导图形，即通过数据特征来寻找更有效的图表生成方式^{[27][28]}。也有一些系统利用用户画像^[29]或者任务目的^[30]来推导合适的图表结构，而有的系统则从统计学角度来进行可视化推荐的工作^[31]。

智能可视化的道路不乏闪光者，但仍有许多空白还未填补。本文将尝试在这些研究的基础上，从更多的角度对智能可视化作进一步探讨。

2.5 本章小结

本章对前人相关技术进行总结，本文系统的设计与实现将在这些前人工作的基础上展开。2.1 小节讨论了高维数据降维的常用方法，本文 4.1 小节中的图形透视表将在此讨论上进行推广；2.2 小节介绍了数据特征、标记、可视化编码的相关知识，本文第五章的推荐系统将在此基础上展开；2.3 小节介绍了图形符号学和图形语言，本文 4.2 小节中将对图形语言进行改进，使其更适合于本系统；2.4 小节介绍了当前的智能可视化方面的进展。

第3章 系统架构

3.1 基本概念描述

(1) 图形透视表

在 2.1.3 小节中，我们介绍了数据透视表，但如果数据透视表中的内容不是单纯的数字，而是图形的话，就称这种数据透视表为图形透视表^{[32][33]} (Graphical Pivot Table)，并把图形透视表的行列的层级称为透视结构。

(2) Pane 与 Cell

图形透视表中一行和一列交叉所形成的空间称为 Pane。一张图表占用的空间称为 Cell，一个 Pane 可以包含多个 Cell^{[32][43]}。

图 3-1 中，X 轴（列）上有两个字段“客户”、“地区”，Y 轴（行）上有三个字段“类别”、“求和(数量)”、“求和(折扣)”。在这种情况下，“客户”和“类别”是真正用来划分图形透视表的字段，而“地区”、“求和(数量)”、“求和(折扣)”则为一个 Pane 中真正拥有的字段，其中 Pane 内分为两个 Cell，分别为“地区”-“求和(数量)”形成的 Cell，与“地区”-“求和(折扣)”形成的 Cell，如图 3-2 所示。

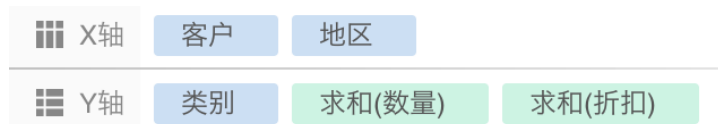


图 3-1 Pane-Cell 行列字段示意图

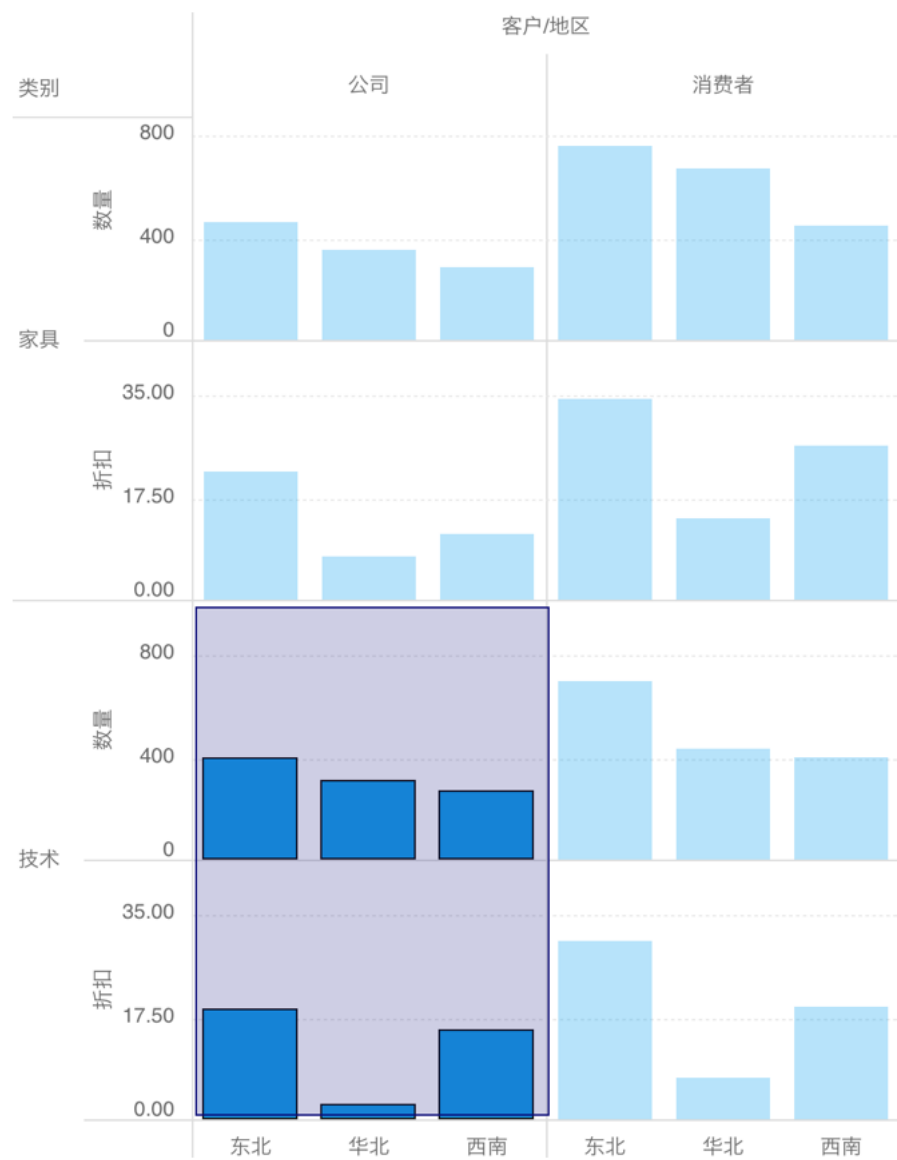


图 3-2 Pane-Cell 示意图

(3) 图形设计

图形设计 (Graphical Design) 描述了一种可视化图表的图形结构信息^[21], 包括它是由哪些图形语句根据什么组合算子组合而来, mark 集合包括什么, 哪些可视化属性编码了哪些数据等。

(4) Visual Query

对图表信息的简易描述称为可视化查询语句 (Visual Query)，其中包含了行列上字段的排列顺序、标记的视觉属性、数据成员排序依据、数据成员筛选规则等。Visual Query 是一个用户层的数据结构，因此只需要用户从宏观上提供图形透视表的基本信息即可。

3.2 系统工作流程

本文的智能可视化系统包括了三个工作场景：图形透视表配置的生成、单字段配置推荐、多字段配置推荐。其中图形透视表配置的生成是指针对确定的 Visual Query 生成图形透视表的配置；图表单字段推荐是指针对确定的 Visual Query 和一个确定的字段，将这个字段放入 Visual Query 中合适的位置，生成一个新的 Visual Query，使得新的图形透视表有尽可能好的可视化表达效果；图表多字段推荐是指对确定的若干个字段，计算适合于展示这些字段的图表类型的优先级，并对任何一种图表类型，推导出最合适展示这些字段的 Visual Query，并生成对应的图形透视表配置。

本文系统已作为一个子模块接入商业数据智能可视化平台网易有数中，关于网易有数的系统架构见 6.1 小节。本文系统将着重处理图形透视表配置的生成与推荐，即上面介绍的三个工作场景。

3.2.1 图形透视表配置的生成

图 3-3 展示了以 Visual Query 作为输入的情况下，输出对应的图形透视表的配置与解释信息的工作流程，其中箭头的方向代表数据的流向。

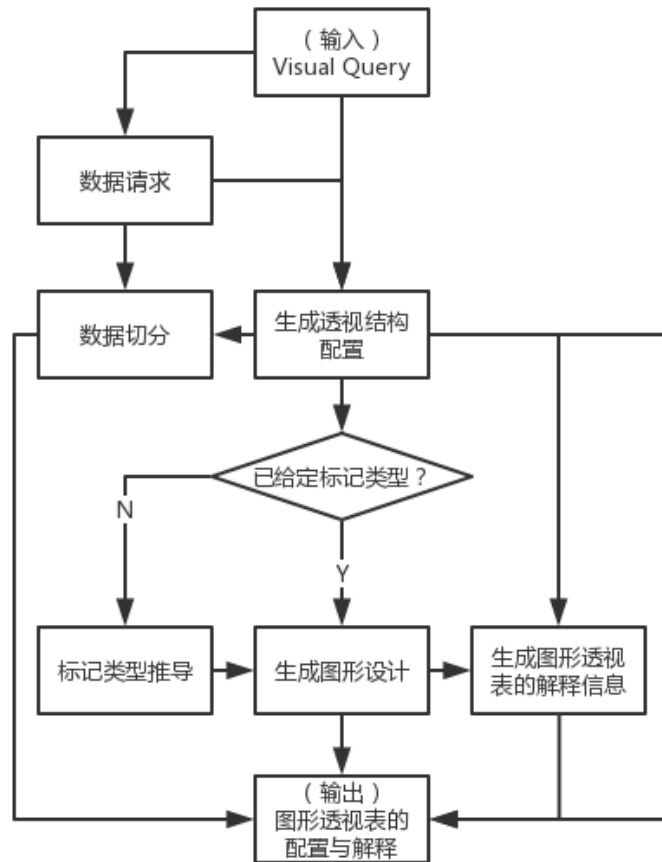


图 3-3 图形透视表配置生成的工作流程

该流程以输入一个 Visual Query 开始，首先针对 Visual Query 中出现的字段，向数据库进行数据请求，然后通过请求到的大数据集(Big Dataset)与 Visual Query 生成图表的透视结构配置 (Pivot Table Config)，然后根据该透视结构配置把数据表进行数据切分，得到每个单元格中所拥有的小数据集(Small Datasets of Cells)。与此同时开始生成每个 Pane 内部的图形设计 (Pane Graphical Design)。如果用户没有事先指定标记类型 (Mark Type)，那么先推导出最适合的标记类型，然后进行图形设计的生成；否则，如果用户已经指定了标记类型，

那么就直接生成图形设计。之后，根据透视结构配置与图形设计，生成图形透视表的解释信息。最后输出图形透视表的配置（包含透视结构配置、图形设计、切分数据集）与解释信息。

3.2.2 单字段配置推荐

图 3-4 展示了单字段配置推荐的工作流程。

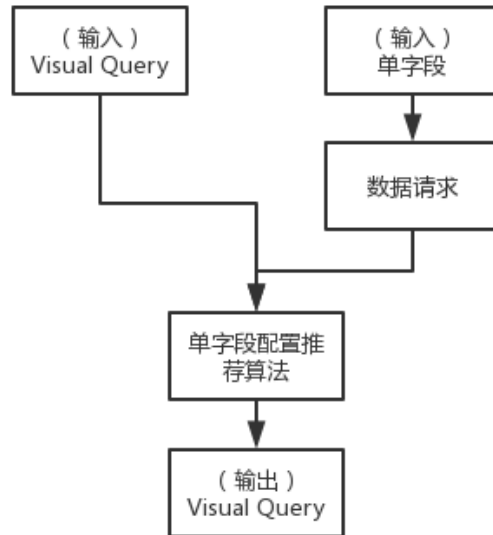


图 3-4 单字段配置推荐的工作流程

当用户已经有了一个确定的 Visual Query，此时对某个新字段感兴趣时，可以将原有的 Visual Query 与该新字段作为输入，在进行数据请求后执行单字段配置推荐算法，推导出在当前 Visual Query 基础上最适合展现新字段的方式（例如把新字段作为行或列的一部分，或是用颜色、尺寸、形状等视觉属性来展现它），并输出新的 Visual Query。用户可以使用这个新的 Visual Query 进行图形透视表配置的生成，或者继续往里添加新字段。

3.2.3 多字段配置推荐

多字段配置推荐有两个步骤：1) 根据用户给定的若干个字段，计算所有适合于展示这些字段的图表类型的优先级；2) 对这些字段和用户指定的某个图表类型，推导出在该图表类型下最适合展示这些字段的图形透视表的配置。

图 3-5 展示了计算图表类型优先级、以及在指定图表类型的情况下生成图形透视表推荐配置的工作流程。

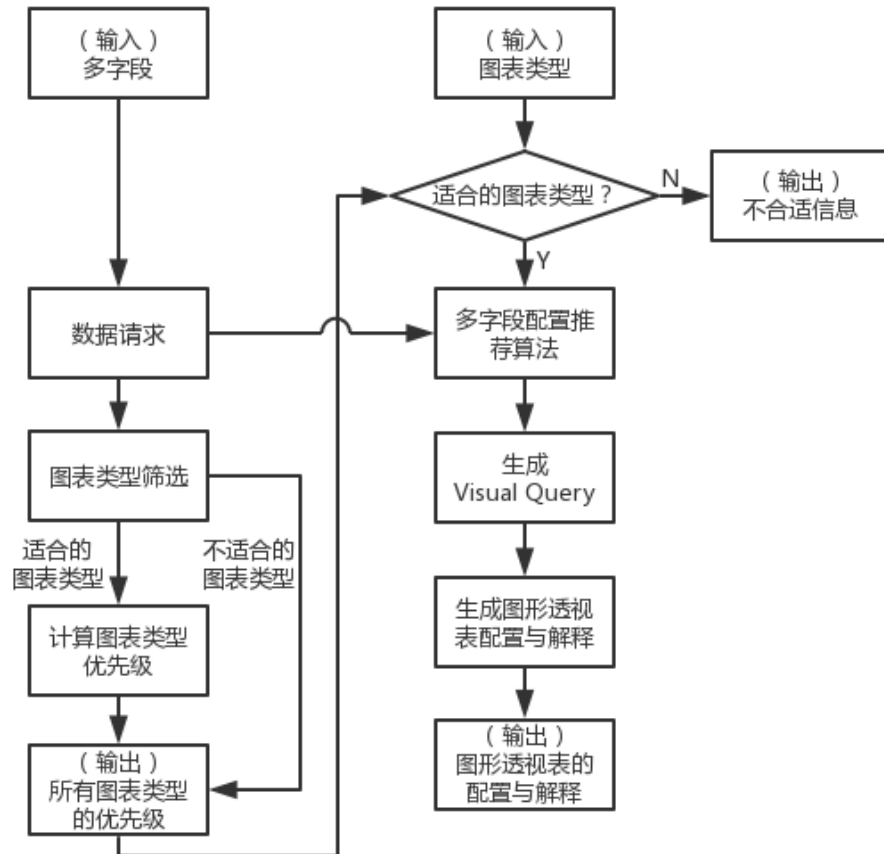


图 3-5 多字段配置推荐的工作流程

该任务将用户输入的若干个字段的输入，在进行数据请求后，筛选出适合于展示这些字段的图表类型，计算这些图表类型的优先级，同时把不适合展示这些字段的图表类型的优先级设为某个特殊值。之后，当用户针对这些字段选择了一种想展示的图表类型时，先判断该图表类型是否适合用于展示这些数据，如果不适合，则返回相应信息；否则，进行多字段配置推荐算法，初步推导出在该图表类型下适合于展示这些字段的方式，并转换为 Visual Query，再执行 3.2.1 小节中图形透视表配置的生成流程，最后得到图形透视表的配置与解释信息。

3.3 系统架构设计

下面将工作流程模块化，以降低耦合性，并方便编码。根据 3.2 小节给出的工作流程，可以将系统分为以下模块：流程控制模块、数据获取模块、透视结构计算模块、图形设计生成模块、图形透视表配置解释模块、标记类型推导模块、单字段配置推荐模块、多字段配置推荐模块。这些模块之间的调用流程如图 3-6 所示，其中每个模块的工作见各小节。

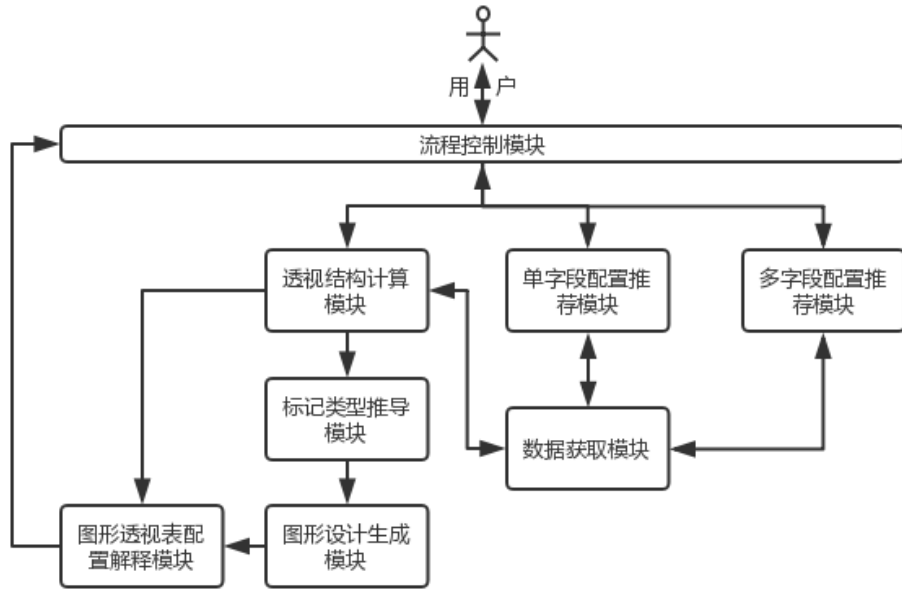


图 3-6 模块划分示意图

3.3.1 流程控制模块

流程控制模块用来保证整个系统的稳定运作，控制模块间的数据传输过程，同时保证输入数据的接收与输出结果的返回。由于本文系统共有三个分立的工作流程，因此需要在此模块中对用户的请求进行解析，判断其属于图形透视表配置生成请求、单字段推荐请求、多字段推荐请求的哪一个，并调用正确的模块。

3.3.2 数据获取模块

针对 Visual Query 中涉及到的字段，或是推荐模块中用户给出的单字段与多字段，都需要向数据库进行请求。在获取到数据集后，还需要根据一定需求，对数据集进行一系列数据变换，例如排序、过滤、聚合等。此模块中对数据的数

据变换操作由专门的数据处理模块完成，不属于本文工作。

3.3.3 透视结构计算模块

对图形透视表来说，需要得到其透视结构配置，因此此模块以 Visual Query 作为输入，以透视结构配置作为输出。本文提出了新的表代数算子，并使用改进后的表代数表达式对透视结构配置进行计算求解。

3.3.4 图形设计生成模块

在得到透视结构配置后，我们需要知道每个 Pane 中的图形设计，因此此模块以单元格配置（即经过透视结构计算模块处理过的 Visual Query）为输入，先合成出对应的图形语句（Graphical Sentence），然后将图形语句转换为图形设计进行输出。为了便于对比可视化结果，由图形透视结构划分出的各个 Pane 内的图形设计是相同的，而这同时也提高了生成图形设计的效率，因为如果每个 Pane 都有自己的图形设计的话，当图形透视表比较庞大时，就会消耗大量的时间来生成每个 Pane 各自的图形设计，这会对效率产生极大影响。

3.3.5 图形透视表配置解释模块

为了使图形透视表的配置有较好的可读性，此模块通过解析透视结构配置和图形设计，生成用来描述图形透视表配置的一段文字。这段文字将可以对图形透视表的透视结构和每个 Pane 中的图形设计进行说明，以让不是数据分析师的用户也能很好地读懂图形透视表。

3.3.6 标记类型推导模块

有时用户可能对感兴趣的数据不太了解，于是对生成的图形透视表的标记类型不是很有把握，那么就可以把这个工作交给这个模块来做。标记类型推导模块将单元格配置作为输入，推导出最适合展示这个图形透视表的标记类型。这将有助于用户对数据特性进行更好的理解。

3.3.7 单字段配置推荐模块

如果用户已经有了一个初步的 Visual Query，但是又对新字段产生了兴趣，那么就需要把这个字段放入 Visual Query 中。此模块以 Visual Query 和新字段作为输入，以融合了新字段的 Visual Query 作为输出。通过这种方式的辅助，新字段将可以在原有的图形透视表上尽可能发挥更大的作用，也就解决了“下一步怎么看”的问题。

3.3.8 多字段配置推荐模块

如果用户有若干个想要看的字段，但是又不知道什么图表类型有利于这些字段展示出更有价值的信息，那么此模块将会先计算所有图表类型针对这些字段的优先级，并将优先级最高的图表类型作为最推荐的图表类型，此时用户可以选择这个最推荐的图表类型，让此模块推导出在此图表类型下最适合展示这些字段的图形透视表的配置。而如果用户想要尝试其他图表类型，也可以获得相应的图形透视表的配置。此模块辅助用户在数据探索的初期进行高质量的图形透视表的绘制，解决了“应该怎么开始看”的问题。

3.4 接口设计

对一个字段来说，其数据结构中包含了数据特征的信息，本文系统将其称为 Field, 定义如下：

```
"Field": {  
  "$type": "DimensionField" | "MeasureField"  
  "role": "Dimension" | "Measure"  
  "interpretation": "Discrete" | "Continuous"  
  "dataType": "Integer" | "Decimal" | "String" | "Date" | "Time" | "Geo"  
  "geoRole": "None" | "City" | "Province" | "Country" | "Latitude" |  
  "Longitude"  
}
```

其中 role 代表数据角色，interpretation 代表数据阐释，dataType 代表数据类型，具体含义在 2.2.1 小节中已经说明。geoRole 代表字段的地理属性，如果不是地理属性则为 None，否则为对应的地理属性。

下面是 Visual Query 数据结构：

```
"VisualQuery": {  
  "row": [Field]  
  "column": [Field]  
  "mark": markObject  
  "markMatrix": [markObject]  
  "sort": sortInfo
```

```
"filter": [filterInfo]

"measure": [MeasureField]

}
```

其中 row 和 column 是行列上放置的字段信息，sort 是排序的信息，filter 是对字段中数据成员进行筛选的信息，measure 是数据角色为 Measure 的字段的集合，mark 是图形透视表的整体标记信息 markObject，而如果一个 Pane 里有多 个 Cell，那么 markMatrix 就是各个 Cell 的标记信息。markObject 的具体定义 如下所示：

```
"markObject": {

    "markType": String

    "color": Field

    "size": Field

    "label": Field

    "shape": Field

    "labels": [Field]

    "details": [Field]

}
```

其中 markType 表示标记类型，它可能是 Bar、Line、Area、Scatter、Text、Pie、FilledMap、GanttBar 等基本图形元素；color、size、angle、shape、labels、details 是标记的视觉属性。

3.5 本章小结

本章对系统架构设计作出了详细的说明。3.1 小节对本文的重要基本概念进行了初步介绍；3.2 小节叙述了系统中三个主要功能的工作流程，即图形透视表配置的生成工作、单字段配置推荐工作、多字段配置推荐工作；3.3 小节根据工作流程中涉及到的内容将系统模块化，使系统具有低耦合性和复用性，并对各模块的内容和作用进行了简要的说明；3.4 小节对系统的输入数据结构及接口设计进行了说明和解释。

本文系统作为一个子模块接入商业数据智能可视化平台网易有数中，关于网易有数的整体系统架构与交互设计见 6.1 小节。

第4章 图形透视表配置的生成

本文把透视结构配置、图形设计、切分后的数据合称为图形透视表的配置，本章主要讨论图形透视表配置的生成。关于此问题已有较为初步的解决方案^[44]，但其功能有限，扩展性较弱，无法对复杂的组合图表进行描述。本文改进了图形透视表配置的生成算法，并将整个生成过程划分为三步：1) 得到图形透视表的透视结构配置；2) 在此基础上切分总数据集，获取每个单元格内的数据；3) 对 Pane 内部的各个 Cell，生成相应的图形设计。本章使用增加了新的表代数算子的表达式来生成透视结构配置，然后基于改进后的图形语言来生成图形设计。此外，为了使图形透视表的配置更容易理解，本章根据透视结构和图形设计生成了文字性的解释，以对图形透视表的配置进行描述。

4.1 透视结构配置的生成

4.1.1 Table Field 与 Pane Field

对一个图形透视表来说，可以把其中的字段分为两部分，其中一部分用来真正地划分透视结构，而另一部分则用来生成 Pane 中的图形设计。我们把真正用来划分透视结构的字段称为 Table Field，而把剩下的字段称为 Pane Field。

Table Field 是真正用来划分透视结构的字段，它只存在于行列上。对行（即 Y 方向）或者列（即 X 方向）来说，如果只存在离散字段，那么除了最内层的离散字段外，其他的外层离散字段全部都是 Table Field；而如果存在连续字段，那么所有的离散字段都是 Table Field。除了 Table Field 以外的所有字段都是

Pane Field（包括行列上剩余的字段和标记的视觉属性所使用的字段）。

图 4-1 中，X 方向上的字段为“客户”和“地区”，均为离散字段；而 Y 方向上的字段为“类别”、“求和(销售额)”和“求和(利润)”，其中“类别”为离散字段，而“求和(销售额)”与“求和(利润)”为连续字段。按照上面所述的逻辑，对 X 方向上的字段来说，所有字段都是离散字段，那么除了最内层的字段“地区”以外，其他所有字段都是用来划分透视结构的字段，也就是说“客户”是 Table Field，而“地区”是 Pane Field；而对 Y 方向来说存在连续字段，于是所有离散字段都是 Table Field，所有连续字段都是 Pane Field，因此“类别”是 Table Field，而“求和(销售额)”和“求和(利润)”是 Pane Field。于是我们便可以知道，这个图形透视表就是一个由“客户”和“类别”这两个字段形成的透视结构，而“地区”、“求和(销售额)”与“求和(利润)”则是一个 Pane 内部所拥有的字段。

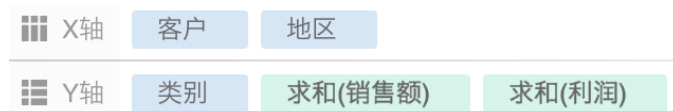


图 4-1 Table Field 与 Pane Field 行列配置

假设“客户”字段有两个成员“公司”和“消费者”，而“类别”有两个成员“家具”和“技术”，那么这个图形透视表就是一个 2*2 的透视结构，“公司”-“家具”、“公司”-“技术”、“消费者”-“家具”、“消费者”-“技术”共形成四个 Pane。如图 4-2，在每个 Pane 中有两个 Cell，即两个图表，其信息分别为“地区”-“求和(销售额)”与“地区”-“求和(利润)”，且这两个 Cell 的图表都是柱形图，此处“地区”有三个成员“东北”、“华北”、“西南”。

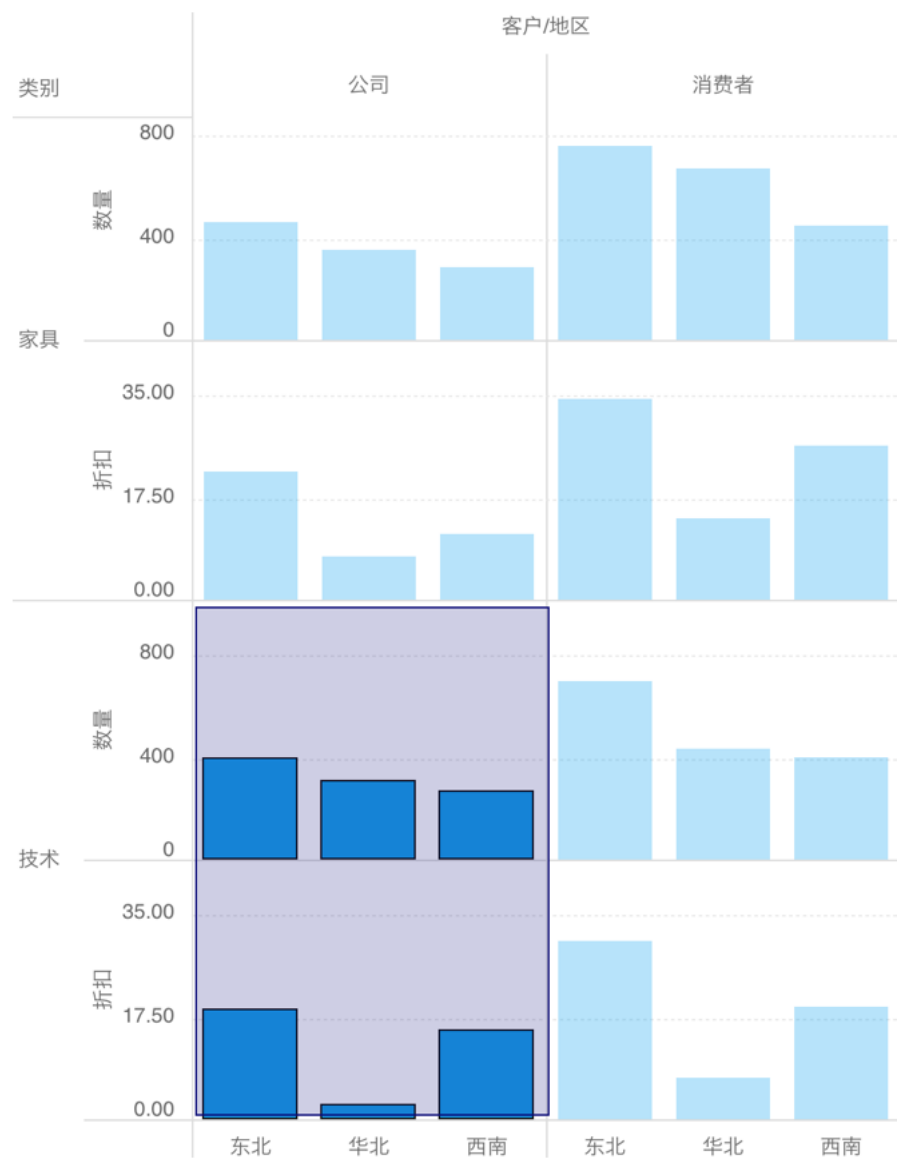


图 4-2 Table Field 与 Pane Field 示意图

4. 1. 2 透视结构配置生成与数据切分算法

4.1.2.1 Query Route 与 Query Sentence

透视结构由行列上的字段及其成员共同切分而成。

例如当列上放置“客户”、“地区”和“类别”三个字段、Y 轴上放置“求和(销

售额)”时, Table Field 是“客户”与“地区”, 而“类别”和“求和(销售额)”是 Pane Field。如果“客户”有两个成员“公司”与“消费者”, 而“地区”有三个成员“东北”、“华北”、“西南”, 那么具体的图形透视表将如图 4-3 所示。可以看到, 整个图形透视表先通过“客户”的两个成员“公司”和“消费者”分为两部分, 然后对着两部分分别通过“地区”的三个成员“东北”、“华北”、“西南”再次进行划分, 最后对生成的每个单元格都绘制一个“类别”-“求和(销售额)”的柱形图。

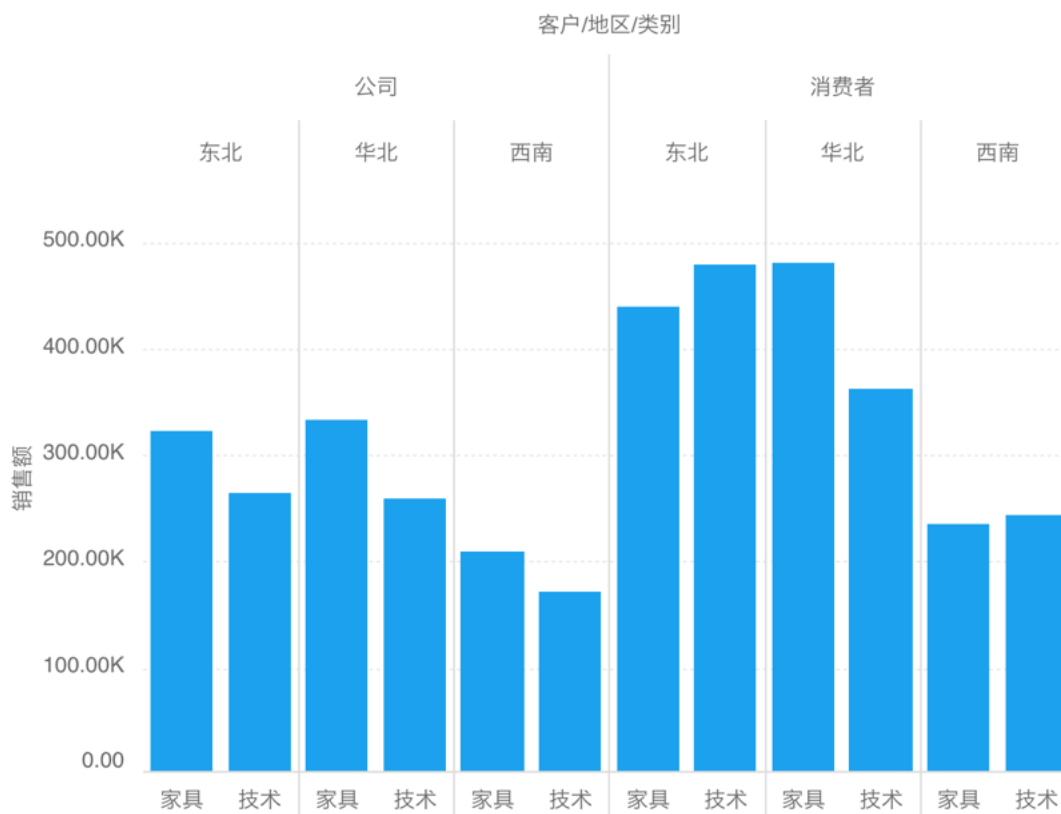


图 4-3 Query Route 对应的透视表

为了最终生成与成员相关的透视结构配置, 同时将大数据集切分为以 Cell 为单位的小数据集, 需要根据 Visual Query 中的行列字段计算出 Cell 的“路径”,

也就是从透视结构的最外层到达 Cell 的路径（可以通过“行路径”和“列路径”定位一类 Cell，这类 Cell 拥有相同的行字段和列字段）。考虑到可以根据这个路径对数据库请求路径上字段的成员，并以此生成与成员相关的透视结构配置，我们将这种路径称为 Query Route。

例如，假设列上的字段为 $[d_1, d_2, c_1, c_2, c_3]$ ，其中 d_1 和 d_2 是离散字段， c_1 、 c_2 、 c_3 是连续字段，那么显然 d_1 和 d_2 是 Table Field，而 c_1 、 c_2 、 c_3 是 Pane Field，于是列上的 Query Route 就是 $[[d_1, d_2, c_1], [d_1, d_2, c_2], [d_1, d_2, c_3]]$ 。

一般地，如果行（或列）上的字段为 $[d_1, d_2, \dots, d_m, c_1, c_2, \dots, c_n]$ ，那么行（或列）的 Query Route 就是 $[[d_1, d_2, \dots, d_m, c_1], [d_1, d_2, \dots, d_m, c_2], \dots, [d_1, d_2, \dots, d_m, c_n]]$ 。

可以通过行列的 Query Route 来定位出每一类 Cell（即拥有相同行字段和列字段的 Cell），而定位某类 Cell 的 Query Route 称为 Cell Query Route。如果将 Cell 的标记视觉属性相关的字段信息与 Query Route 结合起来，就可以生成属于 Cell 的图形设计。我们把 Cell Query Route 与标记属性字段信息合称为 Query Sentence。通过 Query Sentence 可以由 4.2.2 小节的算法推导出 Cell 的图形设计，进而组合出 Pane 的图形设计。

4.1.2.2 Pane Tree

为了生成与成员相关的透视结构配置，一个较为节省空间的做法就是形成树形结构。根据 4.1.1 小节对字段的划分，我们把 Table Field 作为这棵树的非叶子节点，并对其不同成员进行树枝的分岔，而把 Pane Field 作为树的叶节点进行存储。这样就形成了一个与成员相关的透视树，我们把这棵树称为 Pane Tree。

例如对 X 方向上字段如图 4-4 配置时，“客户”和“地区”是 Table Field，

而“求和(折扣)”和“求和(利润)”是 Pane Field。假设客户有两个成员“公司”和“消费者”，“地区”有三个成员“东北”、“华北”、“西南”，那么在 X 方向上就会形成如图 4-5 的 Pane Tree。

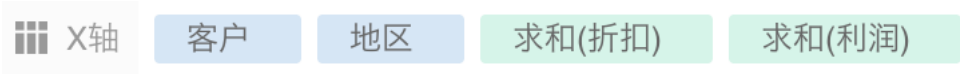


图 4-4 Pane Tree 行列配置

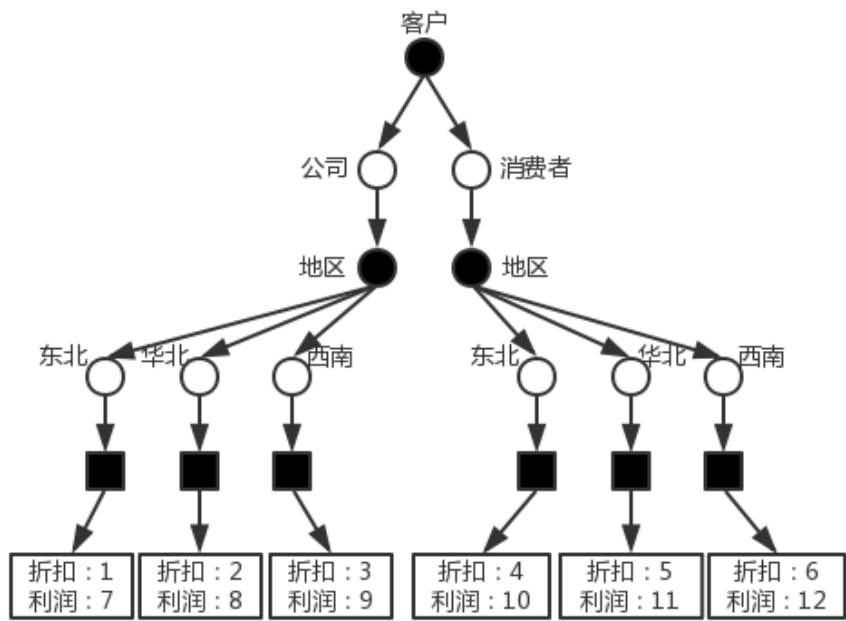


图 4-5 Pane Tree 示意图

其中黑色结点用来标记是 Table Field 还是 Pane Field，黑色圆形为 Table Field，而黑色正方形为 Pane Field。可以看到在每一层黑色圆形下都会根据这个字段的不同成员产生不同的分岔，而最后一层黑色正方向之下的“折扣”与“利润”就是属于这个 Pane 内部字段的信息。

4.1.2.3 基于扩展表代数的 Pane Tree 生成算法

表代数有三种算子，即加算子 (+)、乘算子 (*)、除算子 (/)。本文在此基础上设计了一种新的算子，即半加算子 (符号为 &)，其作为加算子使用，但在其右边的部分将作为整体参与加算子。例如 $\{(左), (右)\} + \{(上), (下)\} = \{(左), (右), (上), (下)\}$ ，而 $\{(左), (右)\} \& \{(上), (下)\} = \{(左), (右), ((上), (下))\}$ 。

在新增了半加算子后，上图对应的表代数表达式为“(客户*地区)&(折扣+利润)”，其中“客户”有两个成员“公司”、“消费者”，“地区”有三个成员“东北”、“华北”、“西南”。

先进行简单表代数表达式的求值：

VAL(客户) = {(公司), (消费者)};

VAL(地区) = {(东北), (华北), (西南)};

VAL(折扣) = {(折扣)};

VAL(利润) = {(利润)};

之后对复合表代数表达式进行求值：

VAL(客户*地区) = {(公司, 东北), (公司, 华北), (公司, 西南), (消费者, 东北), (消费者, 华北), (消费者, 西南)};

VAL(折扣+利润) = {(折扣, 利润)};

VAL((客户*地区)&(折扣+利润)) = {(公司, 东北, (折扣, 利润)), (公司, 华北, (折扣, 利润)), (公司, 西南, (折扣, 利润)), (消费者, 东北, (折扣, 利润)), (消费者, 华北, (折扣, 利润)), (消费者, 西南, (折扣, 利润))}。

针对本文 Pane Tree 生成的问题，可以注意到，只要对 Table Field 做乘算子操作，对 Pane Field 做加算子操作，然后将两者的结果分别作为半加算子的

左右部即可，最后的结果稍加转换格式就可以得到 Pane Tree。

4.1.2.4 基于 Query Route 与 Pane Tree 的数据切分算法

在生成 Pane Tree 后，透视结构配置便已经得到，接下来讨论图形透视表的数据。由于从数据库中请求到的是整个图形透视表的数据全集，因此需要依据透视结构配置对数据以 Cell 为单位进行切分。

考虑到通过 Query Route 可以定位到每一类 Cell，而 Pane Tree 以树形结构存储了透视结构信息与数据信息，因此以 Cell 为单位进行数据切分的工作实际上等价于在 Pane Tree 上进行 Cell 的 Query Route 路径遍历。

数据切分的算法步骤如下：

Step 1: 对某一条 Query Route，从前往后遍历字段，同时 Pane Tree 也从根节点开始往下进行递归；

Step 2: 对遍历到的当前字段，枚举 Pane Tree 当前结点的所有孩子，即该字段的所有成员；

Step 3: 沿着 Pane Tree 收集路径上成员的数据，直到 Query Route 中的字段全都遍历完毕。

以 4.1.2.2 中的例子为例，可以得到其 Query Route 为[[客户, 地区, 求和(折扣)], [客户, 地区, 求和(利润)]]。对[客户, 地区, 求和(折扣)]来说，从 Pane Tree 中得到“客户”字段的成员“公司”与“消费者”，并对这两者分别递归。针对“客户”的成员“公司”和“消费者”，分别枚举下一个字段“地区”的成员“东北”、“华北”、“西南”，并继续递归，此时下一个字段“求和(折扣)”属于叶节点，从叶节点中可以获得对应的数值。于是可以获得隶属于[客户, 地区, 求和(折扣)]的 6 个 Cell 的数据[[公司, 东北, 1], [公司, 华北, 2], [公司, 西南, 3], [消费者, 东北, 4], [消费者, 华北, 5], [消费者, 西南, 6]]。同理可以得到隶属于[客户,

地区, 求和(利润)]的6个Cell的数据[[公司, 东北, 7], [公司, 华北, 8], [公司, 西南, 9], [消费者, 东北, 10], [消费者, 华北, 11], [消费者, 西南, 12]]。

4.2 基于图形语言的图形设计生成

在得到图形透视表的透视结构后, 下一步的任务是生成每个 Pane 内的图形设计。为了更好地对图形设计进行描述, 本文使用图形语言来描述图形结构, 然后再将其转换为最终的图形设计。

4.2.1 图形语句生成算法

可视化图表被看成一种形式化语言, 用来描述可视化图表的语言被称为图形语言 (Graphical Language)。每一个具体的图表都是这种语言中的一句话, 称为图形语句 (Graphical Sentence)。有若干种基本的图形语句, 即基本的图表; 符合一定条件的多个图形语句可以组合成复杂的图形语句, 即复杂的图表。

本文在 Mackinlay 图形语言的基础上, 对部分概念进行了修改与合并, 使其更具有实用性, 并将其应用在本文设计的系统当中。

4.2.1.1 图形语句的组成

一条图形语句分为三个部分: 标记 (Mark)、标尺 (Scale)、视觉 (Retinal)。

(1) 标记 (Mark)

标记是用来展现数据表的图形, 一张数据表对应一个标记集合 (Mark Set), 表中的每条记录 (record) 都由标记集合中的一个实例表示, 称为数据点。标记一般为一些基本的几何图形, 例如 Line、Scatter、Bar、Pie 等。

(2) 标尺 (Scale)

标尺描述了数据点的空间位置属性与数据的关系, 这意味着数据点在空间中

是确定位置的，通过数据点的标尺可以找到每一个数据点。

(3) 视觉 (Retinal)

视觉描述了数据点的视觉属性和数据的关系。不同的标记会有不同的视觉属性集合，常见的视觉属性有颜色 (Color)、尺寸 (Size)、形状 (Shape)、角度 (Angle)、标签 (Label) 等。

由此可以得到图形语句的数据结构,如下所示。本文系统使用四种图形语句,即 HorzSentence、VertSentence、RetinalSentence、CompositionSentence,具体含义在后文介绍。scales 存储了这个图形语句中所有的标尺信息,markSets 是所有标记集合的信息,encodings 是数据集与字段的映射关系信息。

```
"GraphicalSentence": {  
  "type": "HorzSentence" / "VertSentence" / "RetinalSentence"  
    / "CompositionSentence"  
  "encodings": [relationDescription]  
  "scales": [scaleDescription]  
  "markSets": [markSetDescription]  
}
```

映射关系信息的数据结构 (relationDescription) 如下,其代表了从数据集 (DataSetDescription) 到具体字段 (FieldDescription) 的映射。

```
"relationDescription": {  
  "type": "FunctionRelation"  
  "independent": DataSetDescription  
  "dependent": FieldDescription  
}
```

标尺信息的数据结构（即 `scaleDescription`）如下，其中 `encoding` 是该标尺所编码字段的信息。

```
"scaleDescription": {  
  "type": "HorzScale" / "VertScale"  
  "encoding": FieldDescription  
}
```

标记集合的数据结构（即 `markSetDescription`）如下，其中 `kind` 是标记的类型，`axes` 是轴信息，`retinals` 是视觉属性的信息，`encoding` 是数据集信息。

```
"markSetDescription": {  
  "type": "MarkSet"  
  "kind": "Bar" / "Scatter" / "Line" / "Area" / "Gantt" ...  
  "axes": [axisDescription]  
  "retinals": [retinalDescription]  
  "encoding": DataSetDescription  
}
```

轴信息的数据结构（即 `axisDescription`）如下，其中 `placed` 表示这条轴使用的标尺信息，而 `encoding` 则为这条轴将哪个字段应用到标尺上（标尺可能给出了较大的映射集，但对具体的轴来说可以只使用其中的一个子集）。

```
"axisDescription": {  
  "type": "HorzPos" / "VertPos"  
  "placed": scaleDescription  
  "encoding": FieldDescription  
}
```

视觉属性信息的数据结构(即 `retinalDescription`)如下,其中 `retinalType` 表示该视觉属性的类型,而 `encoding` 则为该视觉属性所编码字段的信息。

```
"retinalDescription": {  
  "type": "RetinalProperty"  
  "retinalType": "Color" / "Size" / "Shape" / "Angle" / "Label" ...  
  "encoding": FieldDescription  
}
```

4.2.1.2 两类基本的图形语句

如果一条图形语句不需要由其它图形语句合成而来,其中对图形的描述是最原始(Primitive)的,那么就称这条图形语句是基本的图形语句。本文系统将基本图形语句分为两类:标尺语句(Scale Sentence)、视觉属性语句(Retinal Sentence),下面对这两类基本图形语句分别介绍。

(1) 标尺语句(Scale Sentence)

标尺语句用来描述数据点的空间位置属性,在直角坐标系中可以认为是轴语句。本文系统支持直角坐标系中的水平轴语句(Horizontal Sentence)与垂直轴语句(Vertical Sentence)。

下面的数据结构描述了一条标尺语句,其中 `markSets` 虽然是数组,但里面只有一个元素,该唯一元素 `AxisMarkSetDescription` 表示只含有 `Axes` 字段的 `markSetDescription`; `scale` 表示该标尺语句的标尺信息, `encodings` 是数据集与字段的映射关系信息。

```
"scaleSentence": {  
  "type": "HorzSentence" / "VertSentence"  
  "markSets": [AxisMarkSetDescription]  
  "scale": scaleDescription  
  "encodings": [relationDescription]  
}
```

(2) 视觉语句 (Retinal Sentence)

视觉语句描述了数据点的单个视觉属性，包括颜色语句 (Color Sentence)、尺寸语句 (Size Sentence)、形状语句 (Shape Sentence)、角度语句 (Angle Sentence)、标签语句 (Label Sentence) 等。

下面的数据结构描述了一条视觉语句，其中 markSets 虽然是数组，但里面只有一个元素，该唯一元素 RetinalMarkSetDescription 表示只含有 Retinals 字段的 markSetDescription，而 encodings 是数据集与字段的映射关系信息。

```
"retinalSentence": {  
  "type": "RetinalSentence"  
  "markSets": [RetinalMarkSetDescription]  
  "encodings": [relationDescription]  
}
```

4.2.1.3 两种组合算子

人的视觉感知能力是一种资源，组合操作的意图是尽量合并两个图表兼容的部分，产生出能让人类更高效地视觉感知的图表。一条图形语句要么是基本图形

语句，要么是经过组合操作得来的复合图形语句。

在 Mackinlay 的系统中使用了三种组合算子：双轴合并算子（Double-Axis Composition）、单轴合并算子（Single-Axis Composition）、标记合并算子（Mark Composition）^{[21][22]}。对双轴合并来说，是当出现了两根相同的轴时进行合并；而对单轴合并来说，是当出现了一根相同的轴时进行合并。在本系统中，出于实用性考虑，且考虑到组合操作的意图是尽量合并两个图表兼容的部分，因此尽可能多地合并相同的东西才是效率最高的做法。本文对双轴合并算子与单轴合并算子的概念进行结合，使用轴合并算子（Axis Composition）作为双轴合并算子与单轴合并算子的代替。对轴合并算子来说，只要合并的两个子图有至少一根轴相同，那么就把所有相同的轴全部合并，以尽可能提高视觉感知效率。

下面对本文系统使用的两种组合算子进行说明：

（1）标尺合并算子（Scale Composition）

标尺合并算子就是上面提到的 Axis Composition，只要用于合并的两个子图中拥有至少一个相同的 Scale，就把相同的 Scale 进行合并，生成的复合图形语句中相同的 Scale 只会存在一个。

对两个图形语句应用标尺合并算子的算法步骤如下：

Step 1: 分别取出两个图形语句的标尺 Scale1 与 Scale2;

Step 2: 对 Scale1 中的每一个标尺，寻找 Scale2 中是否存在着相同编码信息的标尺：

1) 若能找到，则把重复的标尺去除，相同信息只保留一份；

2) 若对 Scale1 中的任何一个标尺都无法在 Scale2 中找到有着相同编码信息的标尺，则说明两个图形语句无法应用标尺合并算子，返回失败信息。

(2) 标记合并算子 (Mark Composition)

标记合并算子当两个子图的标记类型相同、标记上已被使用的轴、视觉属性兼容时可以使用，它将两个子图的轴与标记进行合并，生成新的复合图形语句。

对两个图形语句应用标记合并算子的算法步骤如下：

Step 1: 分别取出两个图形语句的标尺 Scale1 与 Scale2;

Step 2: 分别取出两个图形语句的标记集合 MarkSet1 与 MarkSet2;

Step 3: 判断标尺 Scale1 与 Scale2 是否兼容，即对 Scale1 中的 Horz Scale (或 Vert Scale)，在 Scale2 中要么不存在，要么编码了相同的信息。如果不满足，则说明两个图形语句无法应用标记合并算子，返回失败信息；

Step 4: 判断标记集合 MarkSet1 与 MarkSet2 是否兼容，而这需要满足下面三个条件，如果不满足，则说明两个图形语句无法应用标记合并算子，返回失败信息。

1) MarkSet1 与 MarkSet2 的标记类型必须相同；

2) MarkSet1 与 MarkSet2 所编码的数据集必须相同；

3) MarkSet1 与 MarkSet2 中的视觉属性必须兼容，即对 MarkSet1 中的每一个 Retinal 属性，在 MarkSet2 中要么不存在，要么编码了不互斥的信息。

Step 5: 将 Scale1 与 Scale2 中相同的部分合并，将 MarkSet1 与 MarkSet2 中不互斥的部分合并。

对上面的两种合并算子，最后生成的图形语句格式如下。其中 encodings 表示数据集与字段的映射关系信息，scales 表示标尺信息，markSets 表示标记集合的信息，left 和 right 分别表示形成这个复合图形语句的两个子图形语句。无论是轴合并还是标记合并的结果都属于 Composition 语句。

```
"CompositionSentence": {  
  "type": "CompositionSentence"  
  "encodings": [relationDescription]  
  "scales": [scaleDescription]  
  "markSets": [markSetDescription]  
  "left": GraphicalSentence  
  "right": GraphicalSentence  
}
```

4.2.2 基于图形语言的图形设计转换算法

在生成图形语句后，下一步就是将图形语句转换为图形设计。这是由于图形语句本身是递归性质的，它保存了合并过程中的所有路径，这就使得它有许多重复的信息；其递归的形式也不利于其方便地转换成绘制引擎所需要的格式。因此这一步的主要任务就是提取出图形语句中描述了最终图形的部分。

转换主要处理两个内容：Scale 与 MarkSet。对我们最终的图形设计，我们所关心的就是标尺和标记集合，将图形语句转换为图形设计的算法步骤如下：

```
Step 1: 根据最终图形语句的类型，获取 Scale 信息；  
  1) 若图形语句类型为 RetinalSentence，则不存在 Scale 信息，直接返回空 Scale 数组；  
  2) 若图形语句类型为 HorzSentence、VertSentence、CompositionSentence 中的一个，则将图形语句顶层的 Scale 取出作为图形设计的 Scale。  
  
Step 2: 取出图形语句的 MarkSet，根据数据集与字段的映射关系，生成视觉属性字段的具体成员，以方便绘制引擎进行可视化编码。
```


4.3 图形透视表配置的描述解释

在获得图形透视表的配置后，我们已经可以把结果交给绘制引擎进行绘制了，但是如果可以把图形透视表配置以文字的方式展现给读者，同时把一些在 Visual Query 中没有显式出现的内容一并给出，那么这对于理解生成的图形透视表配置有着很好的帮助。

考虑到用户对系统的输入是 Visual Query，因此我们把能从 Visual Query 中直接获取到的信息称为显式信息，而把 Visual Query 中获取不到的信息称为隐式信息。显然，能从 Visual Query 中获取的信息一定能从图形透视表配置中获取到，因此我们可以直接从图形透视表配置中获取需要的信息。下面我们考虑对显式信息和隐式信息分别讨论。

4.3.1 显式信息的描述解释

可以直接从 Visual Query 中获取到的信息主要有：

(1) 每个字段的数据特征

根据 3.4 小节中的叙述，字段的数据特征包括数据角色 (Dimension 或 Measure)、数据阐释 (Discrete 或 Continuous)、数据类型，这些信息都可以直接获取到，在透视结构 Pane Tree 的节点中也有存储。

(2) 真正用来切分透视结构的字段

透视结构的划分规则是：如果全为离散字段，则最后一个离散字段之前的所有字段都用来切分透视结构；如果存在连续字段，那么所有离散字段都用来切分透视结构。这个信息可以从 Visual Query 中得，但是在 Pane Tree 中只要获取到所有非叶子节点即可。

(3) Pane 的结构

同样根据透视结构划分规则，行列上所有的 Pane Field 都将用来进行 Pane 的划分。假设行上有 2 个连续字段，而列上全部是离散字段，那么 Pane 的结构就是 2 行 1 列的 Cell 集合；假设行上有 3 个连续字段，而列上有 2 个连续字段，那么 Pane 的结构就是 3 行 2 列的 Cell 集合。这个信息也可以从 Visual Query 中获得，而在 Pane Tree 中可以直接根据叶节点中相关信息来获得。

(4) Cell 中标记类型、标记的视觉属性

对每个 Cell 的标记类型及视觉属性，是在 Visual Query 中直接给出的，而在图形设计中也会显式地给出。

4.3.2 隐式信息的描述解释

有些信息是无法直接从 Visual Query 中获取到的，将它们给出可以对图形透视表有更好的理解。

(1) 标记的总个数

在本文系统中，标记是与数据记录一一对应的，一条数据记录就对应一个标记，因此获取到标记的总个数能对图形透视表能展示出的信息总量作出一个概述，而这是无法从 Visual Query 中直接获取到的。

(2) Cell 的个数

由于透视划分是由字段和字段成员共同决定的，因此最终图形透视表中 Cell 的个数无法直接从 Visual Query 中得到，而这可以在 Pane Tree 中通过叶节点的个数与叶节点中相关信息计算获得。

(3) 离散字段的成员

在描述透视信息与标记视觉属性信息时，如果能对离散字段的成员个数或具

体成员进行展示，可以对图形透视表的透视结构与视觉信息有更好的把握。例如如果标记的颜色属性使用了有非常多成员的离散字段，那么就应该知道这个离散字段不适合用颜色来展示信息。

(4) 连续字段的统计信息

与离散字段类似，对连续字段来说，较为重要的就是其统计信息，包括但不限于最值、平均值、四分位等。这些信息有利于对图形透视表的进一步探索，因此提供给用户是非常重要的。

4.4 本章小结

本章对图形透视表配置的生成原理进行了详细的描述。4.1 小节给出了划分图形透视表透视结构的规则，设计了用来代表透视结构的 Pane Tree，然后使用扩展了半加算子的表代数表达式来生成 Pane Tree，并以此进行数据切分；4.2 小节优化了图形语言的组合算子，并用改进后的图形语言生成图形语句，然后将图形语句转换为图形设计；4.3 小节将图形透视表配置中的信息归类为显式信息与隐式信息，以此对图形透视表的配置进行描述解释。

第5章 图形透视表配置的推荐

对用户来说，可能没办法在数据探索的初期就绘制出一个很好的图形透视表，这时如果系统能够给出较好的图形透视表配置的推荐，就可以对用户进行数据探索的过程起到很好的辅助推进作用。为了达到这个目的，本文将对图形透视表配置的推荐做出探讨。

本章将分三小节分别解决图形透视表配置推荐的三个问题^[33]：

(1) 如何推荐一个合适的标记类型。

(2) 如何在给定 Visual Query 的情况下，把一个新的感兴趣的字段放入 Visual Query 中，使其有最好的可视化效果，即“下一步怎么看”的问题。

(3) 如何针对给定的若干个感兴趣的字段推荐最合适的图表类型，并对任何一种用户选定的图表类型，生成在该图表类型下最合适的 Visual Query，并导出对应的图形透视表配置，即“应该怎么开始看”的问题。

5.1 标记类型的推荐

对一个除了标记类型以外都确定了图形透视表，选择不同的标记类型可能会给最后可视化的效果带来很大的影响，因此有必要辅助用户进行标记类型的确定。本节先对本系统中标记类型的使用场景进行分析，然后根据结合字段的数据特征进行标记类型的推荐。

5.1.1 标记类型的使用场景

本文系统使用的标记类型有：Bar（柱形）、Line（线）、Area（面积）、Scatter（散点）、Text（文本）、Pie（饼）、GanttBar（甘特柱）、FilledMap（填充地图）。对这些标记类型来说其经验使用场景总结如下：

（1） Bar（柱形）

柱形标记是非常常用的标记，人类对柱子的高度信息非常敏感，因此柱形图一般有两种常见用法：对比有限根柱子的高度，以获取对比；或是用密集的柱子来表现某种连续的趋势，以达到线图的效果。

对第一种用法，其实就是 X 方向是离散字段、Y 方向是连续字段的效果。在这种情况下，可以获得离散字段的各个成员在连续字段上值的对比。

第二种用法的目的是获取趋势，在某些情况下密集的柱子可以获得较强的视觉冲击。但由于其毕竟仍然是有限根柱子产生的结果，因此其一般不如面积图来得直接和准确。

（2） Line（线）

线是将若干个点连接产生的几何图形，人类对线的斜率高低有较为直观的接受和理解的能力。因此最常用于制作线图的场合是展示随着时间的流逝，某个连续字段值的起伏变化。

（3） Area（面积）

面积图是线图的变种。只需要把线往零轴方向填充阴影面积，就可以得到面积图。面积图的出现主要是出于人类对色块面积大小对比的敏感，因此常用来对不同的面积进行对比的情况。

（4） Scatter（散点）

散点图常用来展示两个连续属性的字段之间的关系。例如把出生率和死亡率

分别作为散点图的 X 轴和 Y 轴,然后让不同的散点代表某个国家不同年份的数据,那么就可以通过散点的位置分布得到出生率和死亡率之间的规律。

(5) Text (文本)

当标记本身就是文本时,人们在意的往往只是文字本身,因此此时的展示作用是最主要的。所以文本标记最常用于表格当中,在其他场合则使用不多。

(6) Pie (扇形)

饼图实际上算是极坐标中的柱形图,但在直角坐标系中把扇形单独认为是一个标记也是可行的。饼图的主要作用是用来展示扇形区域被切分的情况,以对比子集占全集的百分比。考虑到对直角坐标系中的单个饼图来说实际上不需要 X 轴和 Y 轴,并且扇形对各种视觉属性都有着很好的相性,因此当轴上不存在字段的时候饼图是很好的选择。

(7) GanttBar (甘特柱)

甘特柱是一种特殊的柱形,它是一种横向的柱子,其中一个用途是展示不同的项目在时间中的起始与结尾。因此它较常用于对离散字段的不同成员在连续时间下的水平柱长的对比。

(8) FilledMap (填充地图)

顾名思义,填充地图适合用于地图中。我们知道,当 X 轴和 Y 轴上是经纬度时就可以画出地图,但是填充地图作为标记类型,它本身实际上是地图上的不同区域,因此如果想要使用填充地图,那么最好有地理属性的字段出现在标记属性中(例如用颜色来区分不同的省)。

5.1.2 基于数据特征的标记类型推荐

在已有的系统中, Tableau 对标记类型的推荐只考虑轴上字段的数据特征^[33], 且只有少量规则, 其智能性较低。而本文系统在此基础上, 根据有字段的轴的个数进行分类讨论, 分情况对标记类型进行推荐, 并且在推荐逻辑中增加考虑了视觉属性对标记类型的影响, 使得智能型大大提高。

5.1.2.1 无轴

当 X 轴和 Y 轴上都没有字段时, 属于无轴情形。此时对标记类型的推荐只能根据标记视觉属性来进行。

依照 5.1.1 中的分析, 在无轴时, 使用除了饼图以外的标记类型均无法提供较为有效的信息来发挥各种标记类型的优势, 而饼图却能通过各种视觉属性来展示更多信息量。例如颜色属性可以区分不同的维度成员, 尺寸属性可以控制组成饼图的扇形半径, 而角度属性则可以控制扇形的角度。因此对无轴的大部分情形下, 都可以使用扇形来作为标记类型的推荐。

但是这并不是说所有情况下都使用扇形, 例如如果使用某个字段来应用颜色属性, 而尺寸、角度没有应用其他字段的话, 只能绘制出一个等分的饼图, 这并没有给出什么额外的信息量, 因此如果此时标签这一项视觉属性上应用了字段的话, 应该使用文本作为首选的标记类型。但另一方面, 当没有字段应用颜色字段时, 角度无法从视觉上区分, 此时如果存在字段应用了标签视觉属性的话, 则也应当使用文本作为标记类型。

另外, 对饼图来说, 使用尺寸和角度来应用离散字段是没有意义的, 因此当尺寸或角度上应用了离散字段时, 不应当使用饼图作为标记类型, 而应当使用文本作为标记类型。

由上面的讨论可以得到无轴情形下标记类型的推荐规则, 见表 5-1, 其中 Yes

表示使用了某字段编码了该视觉属性，No 表示不使用任何字段编码该视觉属性，Any 表示有或者没有字段编码该视觉属性都无所谓，而如果对字段有具体要求则会给出对应说明。

表 5-1 无轴标记类型推荐表

颜色	尺寸	角度	形状	标签	标记类型
Any	Any	Any	离散	Any	Scatter
Yes	No	No	No	Yes	Text
No	Yes	No	No	Yes	Text
No	No	Yes	No	Yes	Text
Any	离散	Any	No	Yes	Text
Any	Any	离散	No	Yes	Text
其他情况					Pie

5.1.2.2 单轴

X 方向和 Y 方向上只有一个方向存在字段的情况称为单轴。此时由单轴上最后一个字段的数据特征与标记视觉属性共同决定最合适的标记类型。

当用户让连续字段来编码角度属性时，我们认为 Pie 是最适合的标记类型；而使用离散字段来编码形状属性时，我们认为 Scatter 是最适合的标记类型。这是因为除了这两种标记类型外，其他类型均不适合用来编码角度和形状属性。

单轴上最后一个字段的数据特征将很大程度影响标记类型。

当这个字段是一个连续度量字段时，使用 Bar 作为标记类型是最合适的，因为此时柱子的高度刚好可以用来展示该连续度量字段，而颜色、尺寸等视觉属性

则能充分发挥柱子的颜色、宽窄对人的视觉刺激，这一点是其他标记类型所做不到的。

而如果这个字段是一个连续维度时，其意味着它可能存在着很多个成员，因此使用 Bar 作为标记类型是不合适的。但是 GanttBar 的起点可以刚好由该连续维度字段所确定，其本身可切分的特性使得它可以代替 Bar 作为最合适的标记类型。

最后，当单轴的最后一个字段是离散字段时，假如此时没有提供任何额外的视觉属性信息，那么使用任何以图形展现信息的标记类型就失去了类型，此时最能使用文本作为最合适的标记类型。需要注意的是，一旦有除了标签以外的视觉属性出现时，就意味着我们可以发挥其他标记类型的优势，例如当维度字段使用颜色进行编码时就意味着可以切分出饼图的扇形，此时如果尺寸属性有连续字段时就能使饼图提供更多信息，因此可以选用 Pie 作为最合适的标记类型；除此之外的情况，不妨使用 Bar 的颜色或者宽窄等视觉特征来展示更多信息。

由上面的讨论可以得到单轴情形下标记类型的推荐规则，见表 5-2，其中 Yes 表示使用了某字段编码了该视觉属性，No 表示不使用任何字段编码该视觉属性，Any 表示有或者没有字段编码该视觉属性都无所谓，而如果对字段有具体要求则会给出对应说明。

表 5-2 单轴标记类型推荐表

轴字段	颜色	尺寸	角度	形状	标签	标记类型
Yes	Any	Any	连续	Any	Any	Pie
Yes	Any	Any	Any	离散	Any	Scatter
连续度量	Any	Any	No	No	Any	Bar
连续维度	Any	Any	No	No	Any	GanttBar
离散	No	No	No	No	Any	Text
	维度	连续	Any	No	Any	Pie
	其他情况					Bar

5.1.2.3 双轴

X 方向和 Y 方向上都存在字段的情况称为双轴。此时由两根轴上最后一个字段的数据特征与标记视觉属性共同决定最合适的标记类型。

与单轴情况类似，当用户让连续字段来编码角度属性时，我们认为 Pie 是最适合的标记类型；而使用离散字段来编码形状属性时，我们认为 Scatter 是最适合的标记类型。

当 X 方向和 Y 方向的最后一个字段同为连续度量或是同为连续维度时，用户最可能需要的信息是这两个连续字段之间的关系，因此使用 Scatter 作为标记类型是最合适的。

当 X 方向和 Y 方向的最后一个字段一个为连续度量字段、另一个为连续维度时字段，考虑到连续维度的连续特征保证了连续精确性，而维度特征给出了这个连续属性上的关键点，因此当另一个字段刚好是连续度量时，使用 Line 作为标记类型可以很好地观察到连续维度字段上面的成员的在连续度量字段上的某种

变化趋势。类似的，当连续维度字段使用时间代替时，用户最有价值的信息是随着时间的变化，连续度量字段的值的某种变化趋势，此时也应当使用 Line 作为标记类型。

当 X 方向和 Y 方向的最后一个字段都是离散字段时，如果没有其他视觉属性被编码、或是只有颜色上编码了度量字段，那么此时意味着没有一种标记类型可以很好地展示几何图形的优势，此时使用 Text 作为标记类型较为合适。而如果在颜色上编码了维度字段，且在尺寸上编码了连续字段，那么饼图就可以展现出更好的信息，此时不妨用 Pie 作为标记类型。对其他情况来说，可以使用 Bar 作为标记类型。

当 X 方向和 Y 方向的最后一个字段一个为离散字段、另一个为连续度量字段时，使用柱子的高度可以最直观地展示两根轴的信息，而这种做法也最贴近人类直觉，因此使用 Bar 作为推荐的标记类型。

当 X 方向和 Y 方向的最后一个字段一个为离散字段、另一个为连续维度字段时，与单轴的情况类似，使用 GanttBar 作为标记类型可以尽可能发挥连续维度字段的信息量，因此使用 GanttBar 作为推荐的标记类型。

最后需要提的是，如果两根轴的最后是地理经度和地理纬度时，应当认为此时用户最需要的是一个地图，因此将 FilledMap 作为推荐的标记类型。

由上面的讨论可以得到双轴情形下标记类型的推荐规则，见表 5-3，其中 Yes 表示使用了某字段编码了该视觉属性，No 表示不使用任何字段编码该视觉属性，Any 表示有或者没有字段编码该视觉属性都无所谓，而如果对字段有具体要求则会给出对应说明。

表 5-3 双轴标记类型推荐表

轴字段 1	轴字段 2	颜色	尺寸	角度	形状	标签	标记类型
Yes	Yes	Any	Any	Yes	Any	Any	Pie
Yes	Yes	Any	Any	Any	Yes	Any	Scatter
连续度量	连续度量	Any	Any	No	Any	Any	Scatter
连续维度	连续维度	Any	Any	No	Any	Any	Scatter
连续度量	连续维度	Any	Any	No	No	Any	Line
时间	连续度量	Any	Any	No	No	Any	Line
离散	离散	No	No	No	No	Any	Text
		度量	Any	No	No	Any	Text
		维度	连续	Any	No	Any	Pie
		其他情况					Bar
离散	连续度量	Any	Any	No	No	Any	Bar
	连续维度	Any	Any	No	No	Any	GanttBar
地理经度	地理纬度	Any	Any	No	No	Any	FilledMap

5.2 图形透视表的单字段配置推荐

本小节主要探讨“下一步怎么看”的问题：当我们已经有了一个可视化局面时，如果有一个新的感兴趣的字段，那么应当把这个字段放在哪个位置可以使得可视化结果尽可能有价值。对这个问题，本文提出了数据特征组合的三原则，并结合一些先验知识设计了单字段配置推荐算法。

5.2.1 启发式配置原理

5.2.1.1 数据特征组合三原则

(1) 吸附性

对一个已有的可视化局面，如果需要将新字段放在 X 方向或者 Y 方向上，那么新字段应当放置在与该字段的数据特征尽可能相似的字段旁边。

这个做法的目的是希望在原有图表的基础上，让新字段辅助展示更多信息。例如当 X 方向放置了离散字段“地区”、Y 方向放置了连续字段“求和(销售额)”，需要把一个连续字段“求和(利润)”放置到合适的位置，那么应当考虑放在 Y 方向上，这样就可以形成“地区”-“求和(销售额)”、“地区”-“求和(利润)”两个结构相同的 Cell，有利于进行对比；而如果需要把一个离散字段“省份”放置到合适的位置，那么应当考虑放在 X 方向上，这样在 X 方向上就有两个字段“地区”、“省份”，可以形成有层级的透视结构，以进行更进一步的数据探索。

(2) 有效性

对一个已有的可视化局面，如果需要将新字段作为视觉属性进行编码，那么使用的视觉属性应当能尽可能符合人类的视觉习惯。

例如当已有的可视化局面是一个散点图时，如果新字段是离散字段，那么应当优先考虑使用形状来编码该字段，这对散点图来说是视觉效率最高的；而如果新字段是连续字段，则应当优先考虑使用尺寸来编码该字段，因为形状不能编码连续字段，而渐变颜色的视觉有效性没有尺寸来得直观。

(3) 多样性

对不同的可视化局面，新字段的添加应当使可视化结果尽可能多样，即尽量不要出现多种可视化局面导向同一个结果的情况。

多样性主要是考虑到一些有经验的系统使用者在脑海中可能已经有了最终

的图形透视表的模样，使用这个功能纯粹是为了快速生成想要的 Visual Query，那么此时应当在保证前两点的前提下，尽可能多地支持多样的图表。也就是说，对用户想要的图形透视表，尽可能存在一种字段添加顺序，使得能够生成用户想要的图形透视表对应的 Visual Query。

例如对以下两种情况：1) X 方向上有一个离散字段“地区”、Y 方向没有字段；2) X 方向上没有字段、Y 方向上有一个连续字段“求和(利润)”。如果第一种情况下添加连续字段“求和(利润)”时放到了 Y 方向上，而第二种情况下添加离散字段“地区”时放到了 X 方向上，那么这两种情况就会导向同一种结果，即 X 方向上是“地区”、Y 方向上是“求和(利润)”。为了尽可能多地支持多样的添加结果，可以考虑把其中其中之一的添加结果换一种仍然非常有效的添加方式。

5.2.1.2 关联配置

如果新字段与 X 方向或者 Y 方向的某个字段本身就具有层级关系，那么字段添加时应当优先考虑与那个字段形成层级关系^[33]。

这种做法是为了让字段添加结果能在原来图形透视表结构的基础上，更深入地进行数据探索。例如如果 X 方向上放置了离散字段“类别”，Y 方向上放置了离散字段“地区”，需要添加的字段是离散字段“子类别”，那么显然把该字段放在 X 方向而非 Y 方向能得到更好的结果。

5.2.1.3 散点图矩阵

在 2.1.1 小节中已经指出，散点图矩阵是观察若干个连续度量字段之间两两关系的好方法，因此在有可能形成散点图矩阵的情况下，优先考虑形成散点图矩阵。

在本文系统中，散点图矩阵的生成只需要在 X 方向和 Y 方向的末尾放置若干

个相同的连续度量字段即可,因此当 X 方向和 Y 方向的末尾都是连续度量字段时,如果新添加的字段仍然是连续度量字段,那么就把 X 方向和 Y 方向所有的连续度量字段、加上新字段一起形成散点图矩阵。

例如 X 方向有连续度量字段“求和(销售额)”,Y 方向有连续度量字段“求和(利润)”,需要添加的字段是连续度量字段“求和(折扣)”,那么就利用这三个字段形成散点图矩阵,也就是让 X 方向和 Y 方向各自都放置“求和(销售额)”、“求和(利润)”、“求和(折扣)”这三个字段,形成一个 Pane 里 3*3 个 Cell 的配置。

5.2.2 单字段配置的启发式推荐算法

由上面的原则及各类讨论,我们已经可以得到从可视化先验知识角度出发的启发式推荐算法。下面针对 X 方向和 Y 方向是否存在字段进行分类,对单字段推荐算法的主要逻辑进行总结。

(1) X 方向和 Y 方向均不存在字段

如果两个方向上均不存在字段,那么根据新字段的数据特征,将该字段配置到两个方向之一,或是配置到视觉属性上。此部分由于当前可视化局面信息太少,因此主要逻辑在于推测用户最可能想要绘制的图表类型,然后把这个字段放到合适的位置上。例如当新字段是离散维度字段时,推测用户可能想要绘制表格,因此把该字段放在 Y 方向上会更合适,因为之后只需要再让一个新的字段使用标签属性,就能绘制出一个表格。

此部分的处理逻辑总结见表 5-4。

表 5-4 X 方向和 Y 方向均不存在字段的推荐逻辑

条件	新字段	推荐配置
	时间	X 方向
颜色属性不存在字段	离散度量	颜色
颜色属性存在字段	离散度量	标签
	离散维度	Y 方向
	连续度量	Y 方向
	连续维度	X 方向

(2) X 方向和 Y 方向只有一个方向存在字段，另一个方向不存在字段

如果两个方向上有一个方向有字段存在，此时这个方向的最后一个字段的数据特征将作为很重要的分类参考。这种情况可能会作为最终图表形成的一个中间阶段，因此引导用户绘制出可视化结果较好的最终图表成为此部分的重点。例如当 X 方向最后一个字段是连续度量字段、新字段也是连续度量字段时，应当考虑把新字段放置在 Y 方向上形成散点图，这是观察两个连续度量字段关系的最好的手段。

此部分的处理逻辑总结见表 5-5：

表 5-5 X 方向和 Y 方向只有一个存在字段的推荐逻辑

条件 1	条件 2	新字段	推荐配置
		时间	X 方向
	颜色属性不存在字段	离散度量	颜色
	颜色属性存在字段		标签
X (Y) 方向存在字段, 且 最后一个是离散字段		离散维度	Y (X) 方向
		连续度量	标签
		连续维度	X 方向
X (Y) 方向存在字段, 且 最后一个是连续字段		离散维度	Y (X) 方向
		连续度量	Y (X) 方向
	X (Y) 方向最后是度量字段	连续维度	Y (X) 方向
	X (Y) 方向最后是维度字段		X 方向

(3) X 方向和 Y 方向都存在字段

由于两个方向上都存在字段, 因此此时新字段的配置方式将更容易做到多种多样。可以考虑结合各种配置思路, 例如吸附性、散点图矩阵等, 以使配置结果有较好的可视化价值。

此部分的处理逻辑见表 5-6:

表 5-6 X 方向和 Y 方向均存在字段推荐逻辑

条件 1	条件 2	新字段	推荐配置
		时间	X 方向
	颜色属性不存在字段	离散度量	颜色
	颜色属性存在字段		标签
X 方向最后是离散字段 Y 方向最后是离散字段		离散维度	两个方向上 字段个数较 少的方向
		连续度量	标签
		连续维度	X 方向
X (Y) 方向最后离散 Y (X) 方向最后连续		离散维度	X (Y) 方向
		连续度量	Y (X) 方向
		连续维度	X (Y) 方向
X 方向最后是连续字段 Y 方向最后是连续字段		离散维度	形状
		连续度量	散点图矩阵
	颜色属性不存在字段	连续维度	颜色
	颜色属性存在字段 尺寸属性不存在字段		尺寸
	颜色、尺寸属性存在字段		标签

5.3 图形透视表的多字段配置推荐

本节解决“应该怎么开始看”的问题：对若干个感兴趣的字段，如何推荐出

最合适展现这些字段的图表类型，以及在该图表类型下图形透视表的配置。5.3.1 小节探讨了能尽可能发挥各种图表类型优势的必要条件；5.3.2 小节给出了所有适合这些字段展现的图表类型的优先级规则，只需要选择优先级最高的图表类型作为推荐的图表类型；5.3.3 小节对任意适合的图表类型，给出在确定的图表类型下的图形透视表的配置。

5.3.1 图表类型的必要条件

每种图表类型都在一些条件下能够尽可能展现该图表的信息优势，我们把这些条件称作图表类型的必要条件。Tableau 中对必要条件进行了简单讨论^[33]，本文将在此基础上进行扩展，以标记进行分类，对每种标记下形成的图表类型的必要条件进行更进一步的探讨。

5.3.1.1 Bar

本小节主要讨论标记为 Bar 的图表，包括堆叠柱形图、堆叠条形图、并列柱形图、并列条形图、百分比柱形图。通过下面的讨论可以得到结论，对本文系统以 Bar 为标记的图表来说，它们的必要条件都是相同的。

(1) 堆叠柱形图/堆叠条形图

堆叠柱形图是指在柱形图的基础上，使用颜色对柱子进行切分，使得能在已有的柱子上通过“子柱”的高度来直观获取到用颜色切分的字段成员的某个度量字段的数值大小差异。由此可以知道，对堆叠柱形图来说，至少需要一个维度字段用来编码颜色属性，至少需要一个度量字段用来作为柱子的高度。

堆叠条形图是把堆叠柱形图交换 X 方向和 Y 方向字段的结果，目的是把柱子变为水平条，以横向宽度来让人感觉到数值的差异。显然堆叠条形图的必要条件与堆叠柱形图相同。

（2） 并列柱形图/并列条形图

并列柱形图是指在柱形图的基础上，使用不同颜色的并列柱子，通过对比柱子的高度来直观获取到用颜色切分的字段成员的某个度量字段的数值大小差异。显然，并列柱形图和堆叠柱形图是非常类似的，所以其必要条件也是相同的。

并列条形图是把并列条形图交换 X 方向和 Y 方向字段的结果，目的同样是为了把柱子变成水平条，以横向宽度来让人感觉到数值的差异，因此并列条形图的必要条件与并列柱形图相同。

（3） 百分比柱形图

百分比柱形图是一类特殊的柱形图，它在堆叠柱形图的基础上，对每一个切分柱子的成员计算该成员的“子柱”占整根柱子高度的百分比，然后使用标签将百分比显示出来。在这种情况下，事实上并不需要在堆叠柱形图的基础上提供更多信息，因此其必要条件与堆叠柱形图相同。

5.3.1.2 Line/Area

本小节主要讨论标记为 Line 或 Area 的图表，之所以把这两种标记放在一起是因为 Area 的图表实际上就是在 Line 的基础上填充上面积而已，其背后的逻辑其实是一致的。本小节的图表包括线图、面积图。

（1） 线图

线图是指将数据点通过线段进行连接，以展现出某种趋势的图表。对线图来说最适合的是展现随着时间的流逝，某个连续度量字段的值的变化趋势。因此对线图来说，其必要条件是至少一个时间字段、至少一个连续度量字段。

（2） 面积图

面积图是在线图的基础上，将线与零轴之间的区域使用有色面积进行填充的

结果。因此面积图与线图的必要条件一致。

5.3.1.3 Scatter

本小节主要讨论标记为 Scatter 的图表，包括散点图、分组散点图。

(1) 散点图

散点图是以两个连续度量字段分别作为 X 方向和 Y 方向上产生的图表，其目的是为了展现两个连续度量字段之间的数值关系。更进一步的用法是用离散字段的成员在散点图上产生大量散点，通过这些散点的分布来寻找两个连续度量字段之间的关系。由此可以得到散点图的必要条件，即至少两个连续度量字段、零或多个离散字段。

(2) 分组散点图

分组散点图的作用与散点图不同，它更倾向于将连续度量字段放在同一根轴上进行比较，而在这种情况下，图中需要有一些散点才有信息量。因此分组散点图的必要条件是至少一个离散字段、至少一个连续度量字段。

5.3.1.4 Pie

本小节主要讨论标记为 Pie 的图表，包括饼图、环图。

(1) 饼图

饼图是指在一个实心圆中使用不同圆心角的扇形来切分这个圆，以不同扇形的面积差异来展现信息的图形。因此，饼图需要有一个离散字段来切分出扇形，同时需要一个连续度量字段来编码尺寸或者角度属性。因此饼图的必要条件是至少一个离散字段、至少一个连续度量字段。

(2) 环图

环图是饼图在图形上的一个变形，是把饼图进行空心化后的图形，因此它的

必要条件与饼图相同。

5.3.1.5 Text

本小节主要讨论标记为 Text 的图表，此处只包括文本透视表一种。

(1) 文本透视表

文本透视表是指每个单元格内直接展示文字或数值的图形透视表。显然它可以出现透视划分，也可以没有透视划分。由于其最主要的作用是展示数据本身，因此必须有一个字段是编码了标签属性的。因此文本透视表的必要条件是最弱的，即只需要存在一个字段就可以绘制文本透视表，而不管这个字段是离散的还是连续的。

5.3.1.6 GanttBar

本小节主要讨论标记为 GanttBar 的图表，其中只包括甘特图。

(1) 甘特图

甘特图是使用水平矩形来提供信息的图表，其每个水平矩形的起点都可以通过维度字段提供。经验上一般使用甘特图完成对时间上不同项目的进度控制，因此甘特图需要至少一个时间字段、至少一个维度字段。

5.3.1.7 Map

本小节主要讨论地图相关的图表，包括标记地图和填充地图。

(1) 标记地图

标记地图是指在地图上用散点来表示在地图每个区域的信息。由于地图本身是通过实际地理信息形成的图形，因此通过散点的颜色、大小可以很容易获取到有着实际地址的信息。标记地图需要能提供地理信息的字段存在，且希望能存在

0 到 2 个连续度量字段。

（2） 填充地图

填充地图是指在地图上用色块填充每一个区域。对填充地图来说，需要有一个字段来绑定颜色信息，而这个字段既可以是提供地理信息的字段，也可以是连续度量字段。

5.3.1.8 Composition

本小节主要讨论通过合并 Cell 可以得到的图表，包括双线图和柱线图。

（1） 双线图

双线图是指由两个标记类型均为 Line 的 Cell 合并而成的图表。在双线图中，一个 Cell 内有一个 X 轴与两根 Y 轴。双线图希望通过这种方式提高对两个连续度量字段的对比。与线图类似，双线图最合适在时间字段上比较两个连续度量字段的值，因此双线图的必要条件是至少一个时间字段、至少两个连续度量字段。

（2） 柱线图

柱线图是指由两个标记类型分别为 Bar 与 Line 的 Cell 合并而成的图表。柱线图的一个 Cell 中同样有一个 X 轴与两根 Y 轴。可以注意到，柱线图只是双线图换了种展现方式，其本质是一致的。因此柱线图的必要条件与双线图相同。

5.3.1.9 图表类型必要条件汇总

由上面的讨论，我们可以总结出本文系统所支持的图表类型的必要条件，见表 5-7。

表 5-7 图表类型必要条件

标记类型	图表类型	必要条件
Bar	堆叠柱形图/堆叠条形图	至少 1 个维度字段 至少 1 个度量字段
	并列柱形图/并列条形图	
	百分比柱形图	
Line/Area	线图	至少 1 个时间字段
	面积图	至少 1 个连续度量字段
Scatter	散点图	至少 2 个连续度量字段
	分组散点图	至少 1 个离散字段 至少 1 个连续度量字段
Pie	饼图	至少 1 个离散字段
	环图	至少 1 个连续度量字段
Text	文本透视表	至少 1 个字段
Gantt	甘特图	至少 1 个时间字段 至少 1 个维度字段
Map	标记地图	至少 1 个地理维度字段 0 到 2 个连续度量字段
	填充地图	至少 1 个地理维度字段 0 到 1 个连续度量字段
Composition	双线图	至少 1 个时间字段
	柱线图	至少 2 个连续度量字段

5.3.2 图表类型的优先级规则

本节讨论不同图表类型的优先级。由于从客观上来说，合适的图表类型与用户本身的意图有关，且不同的图表类型在不同场景下可能有不一样的作用，因此很难在不了解用户意图的情况下计算图表类型的优先级。但考虑到大部分用户都在绘制图表时重复着“历史”，也就是说或许可以通过研究人类经验，来推测用户最有可能需要的图表类型、以及每种图表类型在不同场景下产生的信息量的区别，以此来对用户进行推荐可能是最合适的图表类型。

5.3.2.1 优先原则

本小节总结了图表类型优先级的四个优先原则，以对图表类型的优先级确定提供理论经验基础。

(1) 专有优先

专有优先是指，专有的场合应当使用专有的图表类型，而不应该使用较通用的图表类型。此时专有图表类型的优先级应当较高。

一个经典的例子是填充地图，当用户提供了地理信息的字段时，我们将有很大把握用户希望绘制地图，此时填充地图的优先级应当是最高的。当然有可能用户虽然提供了地理信息但并不想要利用它的这一信息，但是从统计学来说绘制地图确实是用户更希望看到的结果。

(2) 稀有优先

稀有优先是指，图表必要条件要求越高的，优先级越高。也就是所谓的“物以稀为贵”。

例如对文本透视表来说，它只能简单地显示数据的内容，不能以图形的方式提供更多信息。正因为如此，绘制它所需要的必要条件就非常弱，只需要字段存

在就可以绘制文本透视表。对这种必要条件要求比较低的图表类型，就应当设置较低的优先级。

（3）经典优先

经典优先是指，对某一类起到类似展现效果的图表类型，应当把最常用、最经典的图表类型的优先级设置为最高。

例如以 Bar 为标记类型的图表有很多种，但是实际上它们展现信息的方式是非常类似的，即堆叠柱形图和并列柱形图都用颜色来区分不同的维度成员，不一样的只是堆叠柱形图将柱子叠加、而并列柱形图将柱子并列而已。这种情况下无法明确地知道用户的意图，因此应当选用最常用的堆叠柱形图作为推荐。

（4）有效优先

有效优先是指，如果某种图表类型在一些场合下能够恰好发挥其特点，或是能使用恰当的视觉属性更有效地展现信息，那么在那些场合下它的优先级应该提高。

例如对散点图来说，当离散字段个数恰好为 1 个或 2 个时，是最适合发挥散点图的特点的，因为只要让离散字段编码颜色属性或是形状属性，就可以很清晰地展现出连续度量字段之间的关系。此时散点图的优先级应当比其他场合要高。

5.3.2.2 优先级规则

依据 5.3.2.1 小节归纳的几个优先原则，本文对 5.3.1 小节中介绍的 17 种图表类型进行优先级设置。考虑到在不同的场合下满足必要条件的图表类型是不同的，为了使优先级的设置总是有效，本文采取固定优先级的方式，即对每种图表类型在各个场合下设置固定的优先级，当某种图表类型满足必要条件时才考虑它的正数优先级，否则将其优先级设为-1。

表 5-8 给出了本文系统采用的 17 种图表类型的优先级。

表 5-8 图表类型的优先级规则

图表类型	条件	优先级
标记地图		17
填充地图		16
散点图	$1 \leq \text{离散字段} \leq 2$	15 (否则 1)
分组散点图	$1 \leq \text{离散字段} \leq 2$	14 (否则 1)
线图		13
甘特图		12
双线图		11
面积图		10
柱线图		9
百分比柱形图	离散字段 == 2 连续度量字段 == 1	8 (否则 1)
堆叠柱形图	离散字段 ≥ 3	7 (否则 5)
并列柱形图	离散字段 ≥ 3	6 (否则 4)
堆叠条形图	离散字段 ≥ 3	5 (否则 3)
并列条形图	离散字段 ≥ 3	4 (否则 2)
饼图		3
环图		2
文本透视表		1

5.3.3 基于图形语言的图形透视表配置推荐算法

当用户给定了若干个字段、并指定了某个图表类型时，需要在这种图表类型下推荐最合适的图形透视表的配置。显然我们只需要先推荐最合适的 Visual Query，然后将 Visual Query 送入第四章的生成系统中即可得到图形透视表的配置。

5.3.3.1 视觉信息优先级

对人类视觉来说，不同的视觉信息对人类的刺激程度是不同的，因此视觉信息的使用也会存在优先级。在 Mackinlay 的论文中针对数值型 (Quantitative)、无序类目型 (Ordinal)、有序类目型 (Nominal) 的各种视觉信息的优先级进行了讨论，本文则针对本文系统中采用的数据特征设计了视觉信息优先级，见表 5-9。

表 5-9 视觉信息优先级

数据特征	优先级
离散维度	Horz > Vert > Shape > Color > Size > Label > Angle
连续度量	Vert > Horz > Angle > Size > Color > Label > Shape
离散度量	Vert > Horz > Size > Angle > Shape > Color > Label
连续维度	Horz > Vert > Color > Size > Angle > Shape > Label

对一个字段来说，它应当依次考虑从高到低优先级的视觉信息，如果已经有其他字段占用了高优先级的信息，那么就往较低优先级的视觉信息中去考虑。但是同样要注意的是，优先级只是作为一种参考，事实上对不同的图表类型来说，同样的视觉属性也可能有不一样的优先级。例如对饼图来说，它的颜色和角度是最重要的；而对散点图来说，它的颜色和形状则是最重要的。本文系统针对不同

的图表类型做了优先级的优化，但其规则较为繁琐和复杂，不在此处描述。

5.3.3.2 图形透视表配置推荐算法

接下来考虑具体的配置过程。

对一个图形透视表配置推荐算法来说，它必须能保证生成的图形透视表配置是合法的，因此需要在生成 Visual Query 的过程中就要考虑到最终生成的图形语句的合法性。为了这一目的，本文在生成 Visual Query 的过程中同时对基本图形语句进行生成，如果基本图形语句能正常合并，则说明可以使用生成的 Visual Query，否则应当重新生成。

基于此，本文设计了通用的图形透视表配置推荐算法，其算法流程如图 5-1 所示。

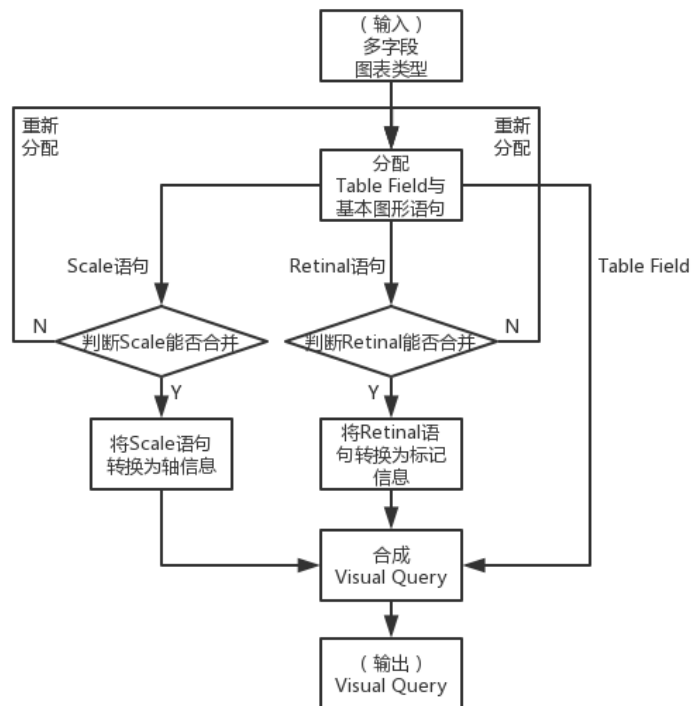


图 5-1 图形透视表推荐算法通用算法流程

根据该算法流程，在输入多字段与图表类型后，分配 Table Field 与基本图形语句，其中基本图形语句包括 Scale 语句与 Retinal 语句。分别判断 Scale 语句与 Retinal 语句能否合并：只要出现合并失败的情况，就重新分配 Table Field 和图形语句，直到合并操作合法为止。之后，将 Scale 语句转换为轴信息，将 Retinal 语句转换为标记信息，与 Table Field 一同合成 Visual Query，并将其返回，此时返回的 Visual Query 一定是合法的，可以用来生成图形透视表的配置。

此处以散点图为例来执行上面的算法流程。散点图的分配规则如下：

- (1) 选择一个连续度量字段分配 X 轴。
- (2) 选择一个连续度量字段分配 Y 轴。
- (3) 将剩余的 0-2 个连续度量字段依次分配到颜色属性和尺寸属性。
- (4) 将所有的离散维度字段依次分配到形状属性、颜色属性、尺寸属性、标签属性。
- (5) 将剩余的离散维度字段依次分配为 Table Field。
- (6) 将所有其他数据特征的字段都分配到标签属性。

根据上面的分配规则，当有 3 个连续度量字段、2 个离散维度字段时，会把其中 2 个连续度量字段分配为 X 轴和 Y 轴的字段，剩余的 1 个连续度量字段分配到颜色属性；然后把 2 个离散维度字段依次分配到形状属性和颜色属性。此时在判断 Retinal 语句合并时，会发现颜色属性同时分配了两个字段，造成了不兼容，说明这种分配方式不可行。在重新分配时，将 2 个离散维度字段依次分配到形状属性与尺寸属性时，可以正确合成所有 Scale 语句与 Retinal 语句。然后把 Scale 语句转换为轴信息、把 Retinal 语句转换为标记信息，然后与 Table Field 一起

生成 Visual Query。

由于不同图表类型的分配规则不同，但配置推荐的整体算法流程是统一的，因此其他图表类型只需要使用相同的流程、不同的分配规则进行分配即可，此处不再赘述。

5.4 本章小结

本章讨论智能可视化中的三个问题：1) 如何推荐合适的标记类型；2) 对给定的可视化局面，把新的感兴趣字段放在哪个位置可以使可视化效果更好；3) 对给定的若干个感兴趣字段，给出适合展示这些字段的图表类型的优先级，并在任意给定的图表类型下对图形透视表的配置进行推荐。5.1 小节讨论了各种常见的标记类型及相关图表类型的特点，并以此设计了图表类型的必要条件；5.2 小节设计了四条优先原则，并结合先验知识设计了图表类型的优先级算法；5.3 小节针对本文使用的数据特征给出了视觉信息优先级，并对给定的图表类型设计了基于图形语言的图形透视表配置的推荐算法。

第6章 系统应用与实例展示

6.1 系统应用

本文的智能可视化系统已作为一个模块接入到商业数据智能可视化平台网易有数^[34]中，其中使用了 NEV^[35]作为可视化图表绘制引擎。

6.1.1 网易有数系统架构

网易有数是一个商业数据智能可视化平台，它可以对接许多关系型数据源，包括 MySQL、Oracle、Hive、Monderain 等。它能对用户感兴趣的字段的组合进行可视化展现，并用美观的图形透视表展现给用户。本文系统作为 Insight 模块接入了网易有数项目，其功能为图形透视表配置的生成与推荐。

图 6-1 是网易有数的项目架构，其分层信息如下：

（1） 界面层

界面层是具体展现给用户信息与相关处理逻辑的层次，包括前端、合成模块、NEV，其中合成模块将 Insight 模块生成的图形透视表配置转换成 NEV 的输入接口，然后前端再调用 NEV 对图形透视表进行绘制。

（2） 应用逻辑层

应用逻辑层是后端处理逻辑的层次，包括 Controller 和 Insight 模块。其中 Controller 负责响应来界面层的所有请求，并将处理结果以合适的形式反馈给界面层，而 Insight 则为本文系统，负责通过 Controller 传过来的 Visual Query 生成图形透视表的配置，或是进行图形透视表配置推荐的工作。

（3） 数据访问层

数据访问层是与数据打交道的层次，有 Data Connector 模块，主要用来对各种数据源进行请求，并对请求的数据经过处理，得到能够直接让 Insight 处理的形式。数据访问层的存在是为了屏蔽底层异构的数据源，而使应用逻辑层能更专注地进行图形透视表相关的工作。其参考了 Tableau 中的做法设计了一套 DSL^{[45][46]}，采用类似 OLAP (Online Analytical Processing) 数据库的方式，扩展了关系数据库，并在此基础上设计了类似 MDX^[47] (MultiDimensional expressions) 与 LINQ^{[48][49]} (Language Integrated Query) 的查询语言。

(4) 数据存储层

数据存储层是真正存储数据的层次，也就是各种数据源，可以是 mySQL、Oracle、Hive、Monderain 等各种异构的数据源。

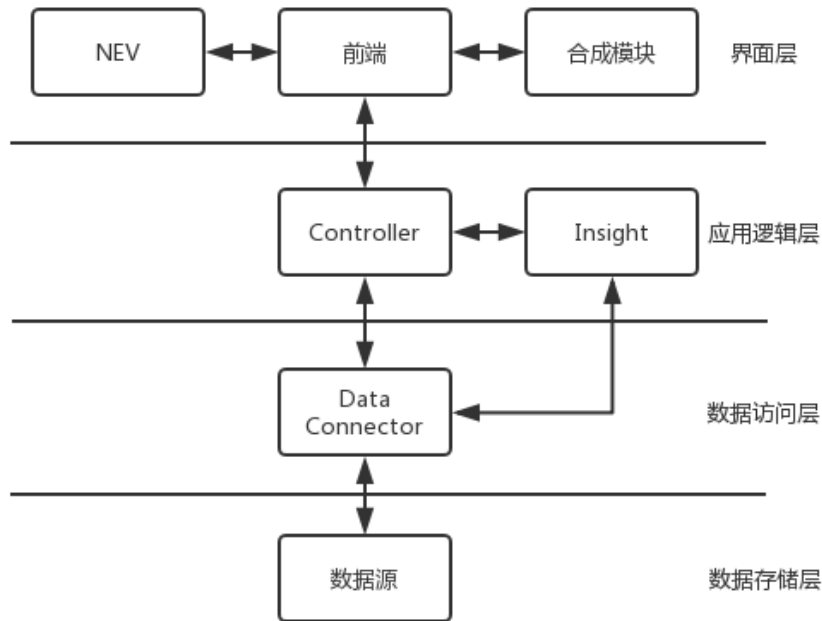


图 6-1 网易有数系统架构图

6.1.2 网易有数交互界面

网易有数的交互界面如图 6-2 所示。其中黄色区域为数据字段区域，表示数据源中的数据字段；蓝色区域为行列字段区域，也就是 X 轴和 Y 轴的区域；绿色区域为标记属性区域，用来控制标记的类型与视觉属性；红色区域为图形绘制区域，这是一个图形透视表绘制完成后用来展示的区域；紫色区域为多字段推荐区域，用来针对多个感兴趣的字段推荐各种图表类型供用户选择。



图 6-2 网易有数交互界面

网易有数的交互与 Polaris 类似^{[50][51]}，采用拖拽的方式，用户只需要在数据字段区域将感兴趣的字段拖到行列字段区域或者是标记属性区域，然后点击运行按钮，就可以在图形绘制区域绘制出对应的图形；如果在一个已有的可视化局面下想要将单个字段加到界面上，只需要在数据字段区域双击那个字段即可；如果

是对多个字段想要让系统推荐合适的图表类型，只需要把这些感兴趣的字段拖到行列字段区域或是标记属性区域，在多字段推荐区域会自动把不适合展现这些字段的图表类型置灰，并把优先级最高的图表类型用蓝色框高亮，而当用户想要让系统推荐具体某个图表类型下的图形透视表的配置的话，点击多字段推荐区域内对应的图表类型即可，系统会自动推导出图形透视表的配置并在图形绘制区域绘制出结果。

6.2 实例展示

6.2.1 数据描述

本节将使用经典的 Superstore 数据集作为测试数据集。主要使用了其中的离散维度字段“客户”、“地区”、“类别”、“省”、“发货日期”、“产品子类别”，以及连续度量字段“求和(销售额)”、“求和(折扣)”、“求和(利润)”。

为了最终的展现效果，本文在部分小节中对有较多成员的字段的成员进行了筛选，筛选后“客户”字段有两个成员“公司”、“消费者”，“地区”字段有三个成员“东北”、“华北”、“西南”，“类别”字段有两个成员“家具”、“技术”。

6.2.2 实例展示

6.2.2.1 透视结构的生成

如图 6-3，假设 X 轴上使用了两个字段“客户”和“地区”，而 Y 轴使用三个字段“类别”、“求和(折扣)”、“求和(利润)”，其中“客户”、“地区”、“类别”均为离散维度字段，而“求和(折扣)”和“求和(利润)”为连续度量字段。

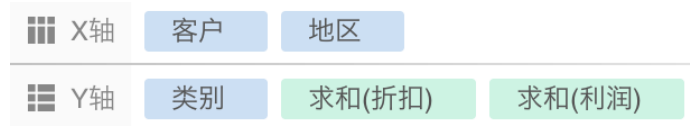


图 6-3 透视结构生成实例的行列配置

根据 Table Field 与 Pane Field 的划分规则，在 X 方向上“客户”是 Table Field，而“地区”是 Pane Field；Y 方向上“类别”是 Table Field，“求和(折扣)”和“求和(利润)”是 Pane Field。

下面我们讨论 X 方向和 Y 方向的表代数表达式。为了叙述简洁，以下用“折扣”代替“求和(折扣)”、用“利润”代替“求和(利润)”。

根据 Table Field 和 Pane Field 的划分，可以很容易知道 X 方向对应的表代数表达式为“客户&地区”，而 Y 方向对应的表代数表达式为“类别&(折扣+利润)”。

先对简单的表代数表达式进行求值：

VAL(客户) = {(公司), (消费者)}；

VAL(地区) = {(东北), (华北), (西南)}；

VAL(类别) = {(家具), (技术)}；

VAL(折扣) = {(折扣)}；

VAL(利润) = {(利润)}；

然后对复杂的表代数表达式进行求值：

VAL(客户&地区) = {(公司, ((东北), (华北), (西南))), (消费者, ((东北), (华北), (西南)))}；

VAL(折扣+利润) = {(折扣), (利润)}；

VAL(类别&(折扣+利润)) = {(家具, ((折扣), (利润))), (技术, ((折扣), (利

润))));

于是可以得到 X 方向和 Y 方向的 Pane Tree，如图 6-4 和图 6-5 所示：

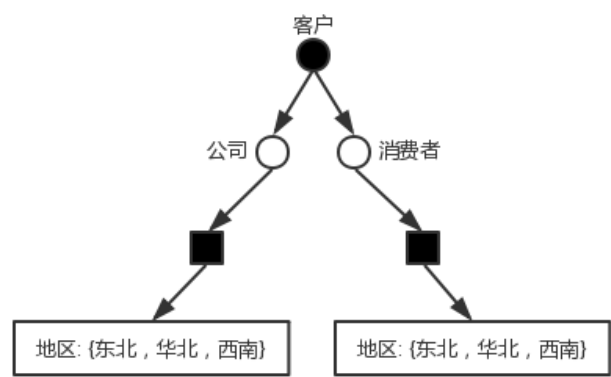


图 6-4 X Pane Tree

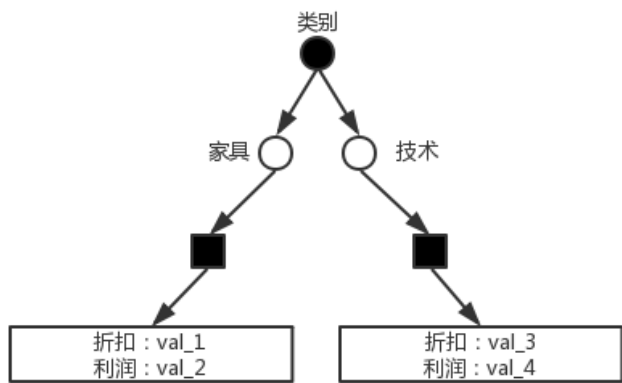


图 6-5 Y Pane Tree

这两棵 Pane Tree 构成了图形透视表的透视结构，共生成了 2*2 个 Pane，即 X 方向“客户”的成员“公司”、“消费者”与 Y 方向“类别”的成员“家具”、“技术”形成的笛卡尔积。

6.2.2.2 图形设计

对一个如图 6-6 所示的行列配置,在这个图形透视表的 Pane 中有两个 Cell,其中一个为“年(发货日期)”-“求和(折扣)”,另一个为“年(发货日期)”-“求和(利润)”。

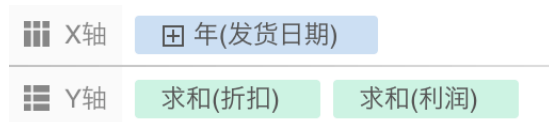


图 6-6 图形设计实例的行列配置

我们对“年(发货日期)”-“求和(折扣)”这个 Cell 的标记进行设置,将其类型设置为 Line,并使用“求和(销售额)”编码尺寸属性、使用“求和(折扣)”编码标签属性;再对“年(发货日期)”-“求和(利润)”这个 Cell 的标记进行设置,将其类型设置为 Bar,并使用“类别”编码颜色属性,如图 6-7 所示。

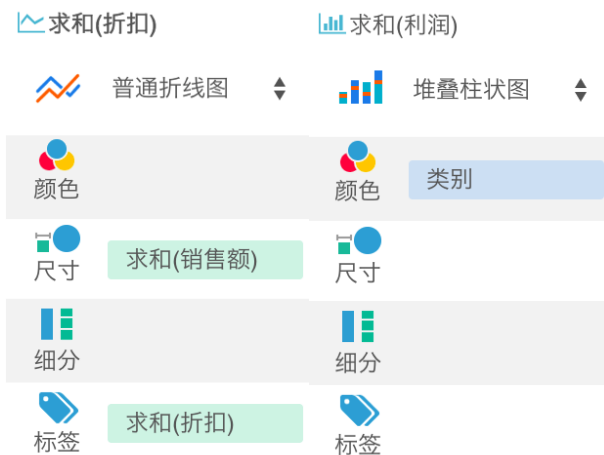


图 6-7 图形设计实例的标记信息配置

此时如果分别绘制这两个 Cell,就会得到图 6-8 和图 6-9。

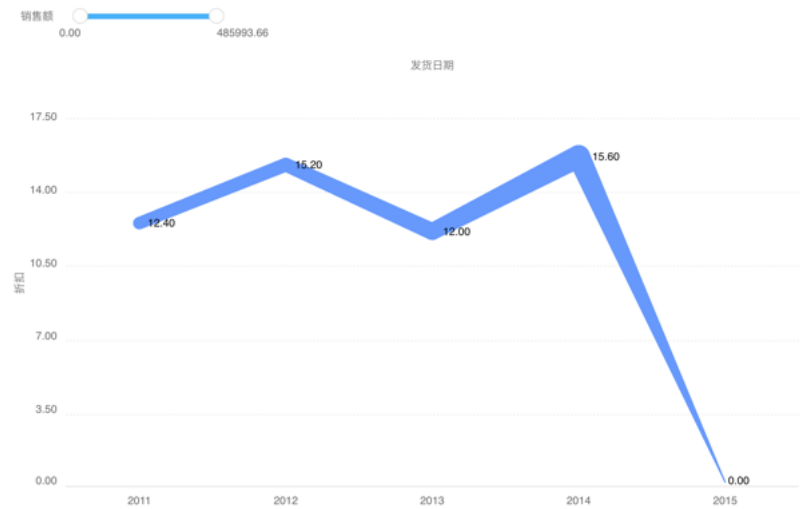


图 6-8 图形设计实例的 Cell 1

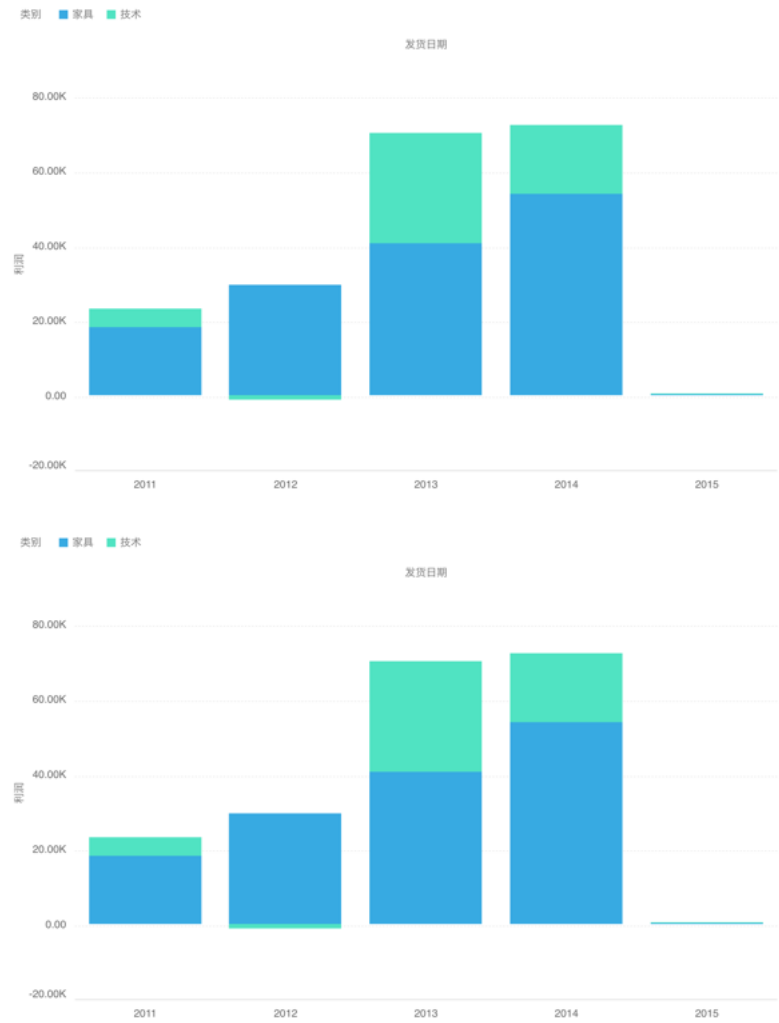


图 6-9 图形设计实例的 Cell 2

而如果把这两个 Cell 进行合并，也就是在一个 Cell 同时显示“年(发货日期)”、“求和(折扣)”、“求和(利润)”，那么就会出现图 6-10。

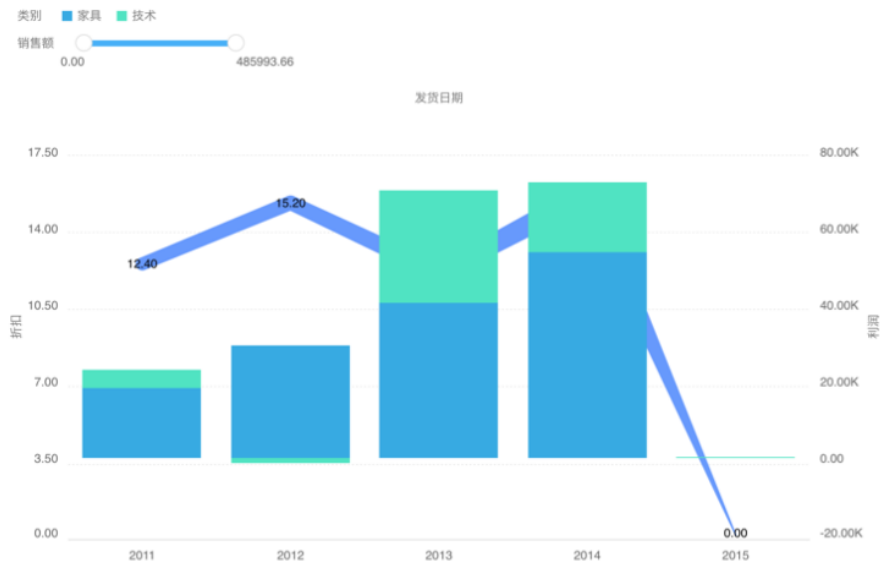


图 6-10 图形设计实例的合并 Cell 图

图 6-11 的树形结构表示了生成的最终的图形语句。

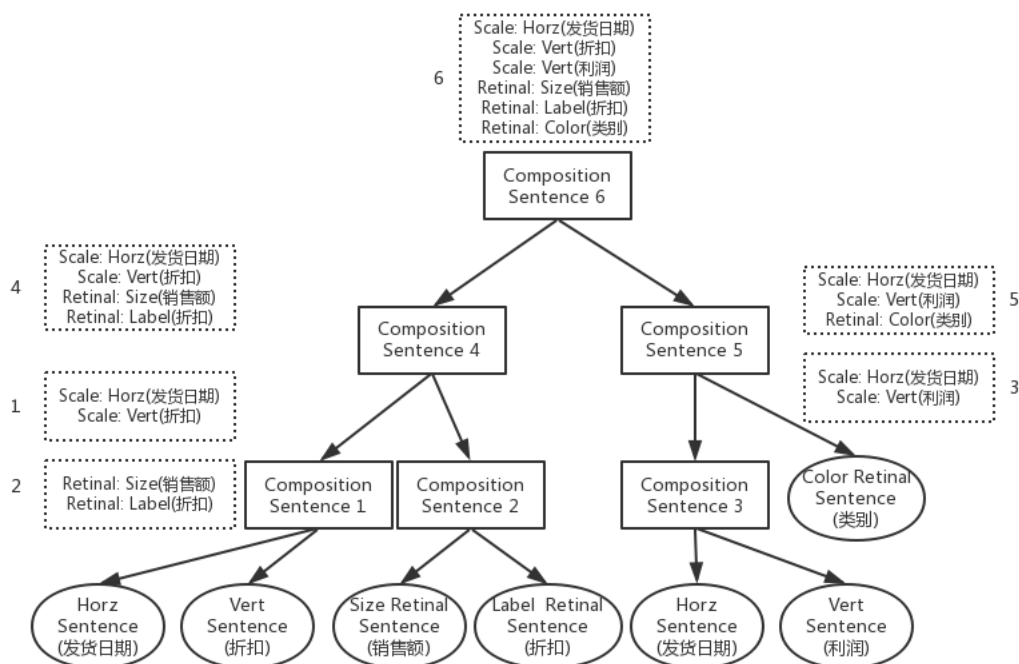


图 6-11 图形设计实例的图形语句树

在这棵图形语句树中，叶子节点表示基本图形语句，非叶子节点表示复合图形语句，而 Sentence 6 代表了最终图形的图形语句，其由两个子图形 Sentence 4 与 Sentence 5 合成而来，而这两个子图形就是上面提到的两个 Cell 对应的图形语句，可以注意到由于它们有着相同的 Scale，即 Horz 轴都为“发货日期”，因此在合并时将这两者进行了去重；而两个子语句的 Retinal 部分都是兼容、不冲突的，因此可以直接合并。

对某一个子语句 Sentence 4 来说，它由 Sentence 1 与 Sentence 2 合成而来，其中 Sentence 1 代表了两个基本图形语句 Scale Sentence 的合成结果，而 Sentence 2 代表了两个基本图形语句 Retinal Sentence 的合成结果。

6.2.2.3 图形透视表配置的描述解释

当 X 方向、Y 方向、标记属性的设置如图 6-12 所示时，把鼠标悬停置图形绘制区域的左上角时，会显示对该图形透视表配置的描述。可以看见此时会显示数据记录的总条数、图形透视表的透视结构信息、Pane 中每个 Cell 的轴信息与标记信息。



图 6-12 图形透视表配置的描述解释示意图

6.2.2.4 标记类型的推荐

如图 6-13，当 X 轴上放置离散维度字段“地区”、Y 轴上放置连续度量字段“求和(利润)”、颜色属性使用离散维度字段“产品类别”时，根据 5.1.2 小节中的逻辑，可以推导出应当使用 Bar 作为最合适的标记类型。



图 6-13 标记类型推荐实例 1

而当 X 轴和 Y 轴都是连续度量字段、标签使用了离散维度字段时，则会推导出应当使用 Scatter 作为最合适的标记类型，如图 6-14 所示。

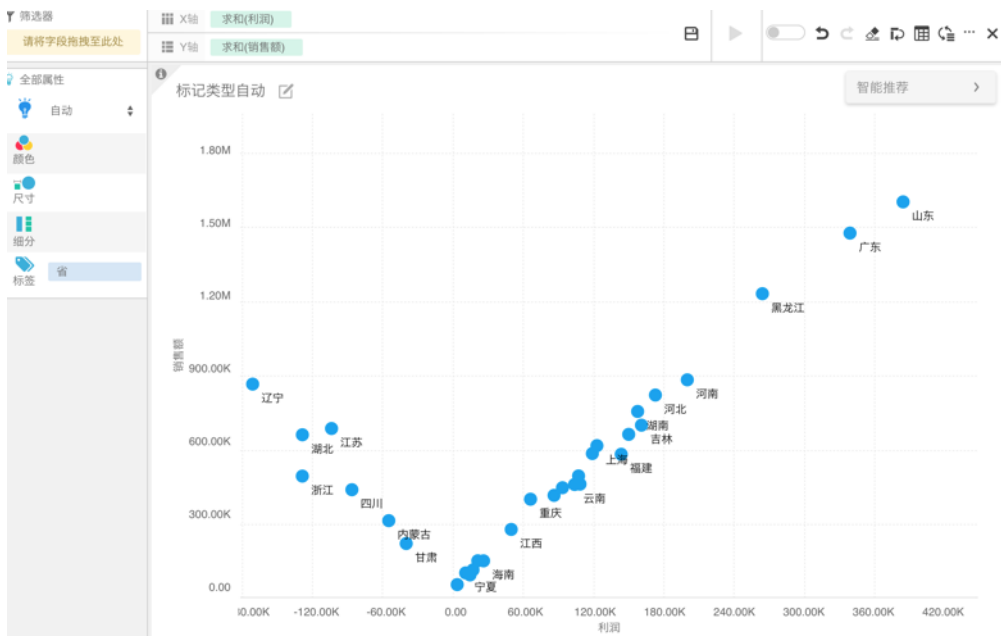


图 6-14 标记类型推荐实例 2

6.2.2.5 单字段推荐

如图 6-15 所示，假设当前 X 轴不存在字段、Y 轴存在连续度量字段“求和(销售额)”、尺寸使用连续度量字段“求和(利润)”时，如果新的感兴趣的字段“地区”是离散维度字段的话，双击这个感兴趣的字段，根据 5.2 小节中的逻辑，会将它直接放到 X 轴上，形成如图 6-16 所示的可视化局面。



图 6-15 单字段推荐实例的配置 1

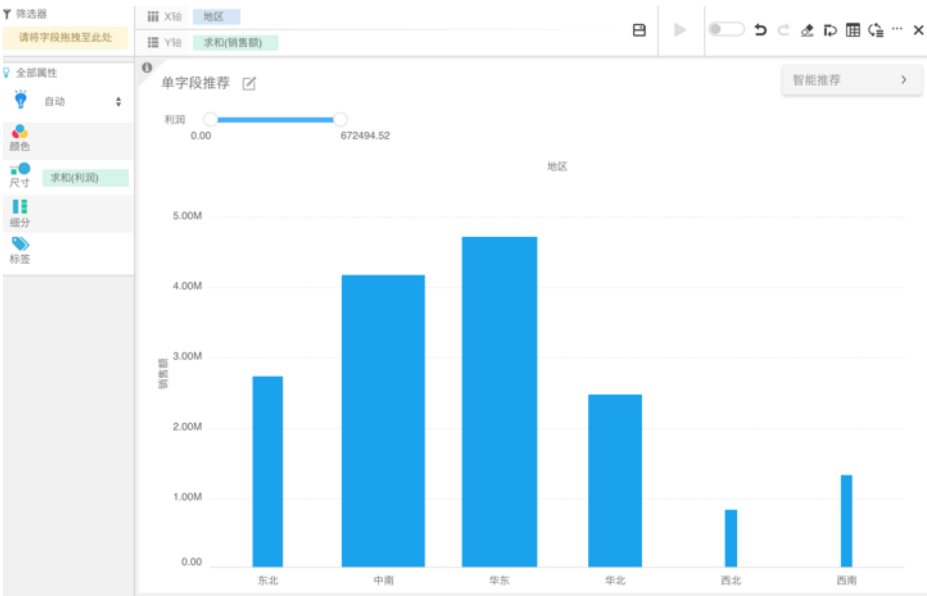


图 6-16 单字段推荐实例结果图 1

而如果当前的可视化局面是 X 轴上放置了连续度量字段“求和(利润)”、Y 轴上放置了连续度量字段“求和(销售额)”、标签上使用了离散维度字段“省”，如果新的感兴趣的字段是连续度量字段“求和(折扣)”，那么双击这个新的感兴趣的字段后就会形成散点图矩阵，如图 6-17 和图 6-18 所示。



图 6-17 单字段推荐实例的配置 2

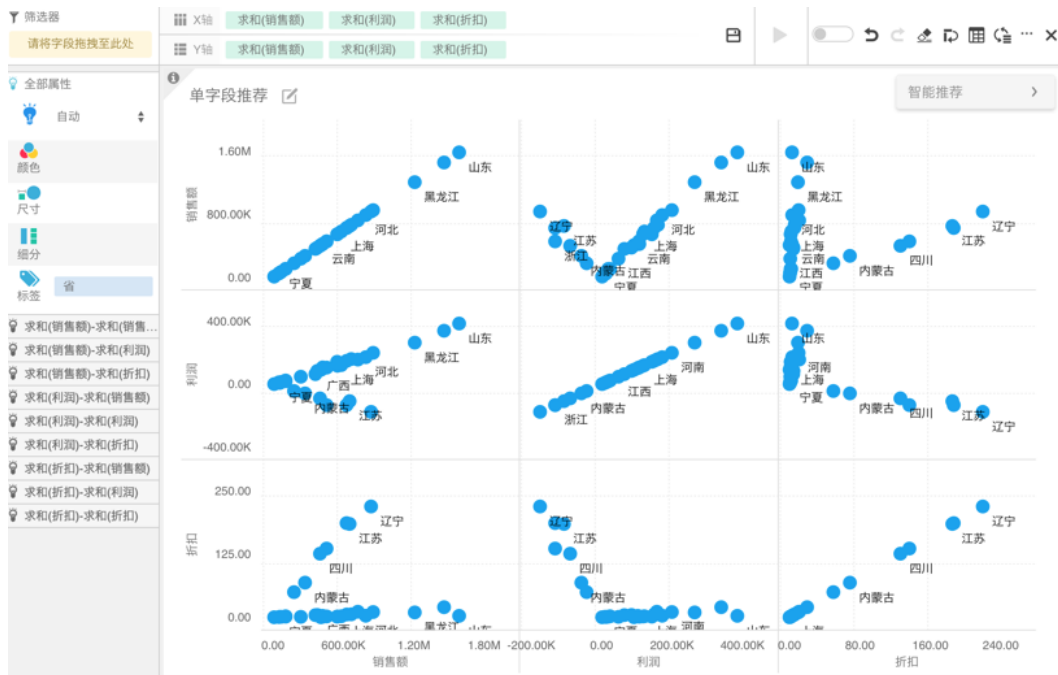


图 6-18 单字段推荐实例结果图 2

6.2.2.6 多字段推荐

如图 6-19，当选择了两个离散维度字段“客户类别”、“地区”，以及一个连续度量字段“求和(利润)”时，可以看到多字段推荐区域中有一部分图表类型置灰了，而另一部分没置灰，分别代表在这些字段下是不可用的图表类型或是可用的图表类型。其中蓝色框高亮的是堆叠柱形图，代表了这种场景下最推荐的图表类型。



图 6-19 多字段推荐实例的字段

此时如果我们点击堆叠柱形图的图标，系统就会自动推导出最合适的图形透视表的配置，并将其绘制出，其结果如图 6-20 所示。可见本文系统把“地区”放在了 X 轴，“求和(利润)”放在了 Y 轴，“客户类别”放在了颜色。



图 6-20 多字段推荐实例结果图 1

当然我们也可以选择其他的图表类型，例如饼图，本文系统也会产生在饼图下最合适的图形透视表的配置，如图 6-21 所示。可以看到，在这种情况下本文系统把连续度量字段“求和(利润)”放在了角度属性上，使得绘制出的图形更适合于饼图。但是对比上面绘制出的堆叠柱形图，可以发现在此种情况下确实是堆叠柱形图的可视化效果更好。

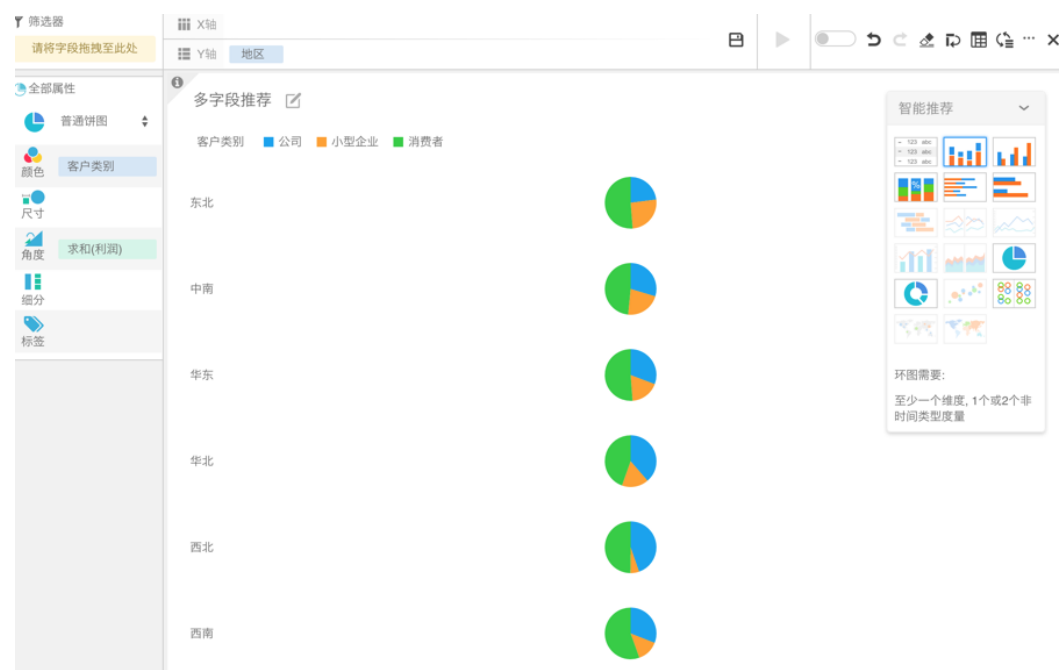


图 6-21 多字段推荐实例结果图 2

6.3 本章小结

本章介绍了商业数据智能可视化平台网易有数的系统架构和交互界面，并在以该产品的呈现效果为例，使用 Superstore 数据源，展现了本文的图形透视表配置的生成与推荐的最终执行结果。可以知道，使用本文系统可以正确生成数据透视表的透视结构、图形设计、文字描述，并且在用户需要时可以推荐给用户尽可能合适的图形透视表配置。

第7章 总结与展望

7.1 本文的工作和贡献

本文在研究国内外数据可视化成果的基础上，设计并实现了图形透视表配置的生成与推荐系统，主要的工作和贡献如下：

- (1) 针对现阶段很多企业繁复的数据分析流程，本文设计并实现了图形透视表配置的生成与推荐系统。该系统可以直接针对用户的可视化请求生成对应的图形透视表配置，并在用户对数据了解很少的情况下提供图形透视表配置的推荐功能，极大地辅助了用户进行数据探索的过程。
- (2) 本文提出了新的表代数算子，并用改进后的表代数表达式辅助生成图形透视表的透视结构及数据集切分工作。
- (3) 本文改进了图形语言的合并算子，并基于改进后的图形语言生成了图形语句，然后通过图形设计转换算法将图形语句转换为图形设计。
- (4) 本文归纳了图形透视表配置中的显性信息与隐性信息，并将其转换为文字提供给用户，使用户能更清楚地了解到生成的图形透视表配置的信息。
- (5) 针对用户对不同场景下标记类型的使用原则不太明确的情形，设计了一套标记类型推荐规则，推荐用户使用最合适的标记类型。
- (6) 本文归纳了数据特征组合的原则，并以此设计并实现了单字段推荐启发式算法，解决了“下一步怎么看”的问题，即用户可以在一个已有的可视化局面上增加一个新的感兴趣的字段，使这个新字段能放在最合适的位置。
- (7) 本文讨论并总结了图表类型的必要条件；基于优先原则与一些先验知识设计了图表类型的优先级规则；设计并实现了基于图形语言的图形透视表多

字段推荐算法，解决了“应该怎么开始看的问题”，即对用户感兴趣的若干个字段，推荐最合适的图表类型，并对任何图表类型，推荐最合适的图形透视表配置。

- (8) 本文使用 Superstore 数据集对系统中的几个关键功能进行测试和说明，验证了本文系统能够正确生成图形透视表的配置，并在需要时可以推荐最合适的图形透视表配置。
- (9) 本文系统已经作为网易有数项目的一个子模块上线，并为网易许多产品提供了智能可视化支持，帮助这些产品决策并获益。

7.2 未来的研究展望

如何使数据可视化尽可能智能一直是这个领域在不断探索的问题。过去人们总结了许多数据可视化的经验和知识，但是这些经验并不能在任何场合都有很好的效果，如何使数据可视化对不一样的用户个体都能尽可能有效是一个很重要的问题。现阶段机器学习领域正在起步，其与可视化领域的结合是个亟待研究的课题，如果机器学习能帮助用户更进一步地进行更智能的数据探索，那么它将为社会贡献的财富将是无可估量的。

参考文献

- [1] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, Hartmut Ziegler. Challenges in Visual Data Analysis[C]. In IV '06: Proceedings of the conference on Information Visualization. IEEE Computer Society, 2006.
- [2] C Stolte, P Hanrahan. Polaris: a system for query, analysis and visualization of multi-dimensional relational databases[C]. In Information Visualization, 2000. InfoVis 2000. IEEE Symposium on. IEEE, 2000, 5–14.
- [3] Chris Stolte, Diane Tang, Pat Hanrahan. Polaris: a system for query, analysis, and visualization of multidimensional databases[J]. Communications of the ACM, 2008, 51(11):75–84.
- [4] FineBI[EB/OL]. <http://www.finebi.com/>.
- [5] Moojnn[EB/OL]. <http://www.moojnn.com/>.
- [6] Datastory Cube[EB/OL]. <https://cube.datastory.com.cn/>.
- [7] Aliyun Data[EB/OL]. <https://data.aliyun.com/>.
- [8] BDP[EB/OL]. <https://www.bdp.cn>.
- [9] W, Hong W, Liu S, Qu H, Yuan X, Zhang J, Zhang K. Information visualization and visual analytics: challenges and opportunities. Science China: Information Science, 2013,43(1):178–184
- [10] Yuan XR. Big data visualization and visual analysis, 2013 (in Chinese). <http://www.chinacloud.cn/upload/2013-12/13122814565172.pdf>
- [11] Tableau Software[EB/OL]. <http://www.tableau.com/zh-cn>.
- [12] Power BI[EB/OL]. <https://powerbi.microsoft.com/>
- [13] QlikView[EB/OL]. www.qlik.com/
- [14] Edward Segel, Jeffrey Heer. Narrative Visualization: Telling Stories with Data[J]. IEEE Transactions on Visualization and Computer Graphics, 2010, 16(6):1139–

- 1148.
- [15] Li Yu, Aidong Lu, William Ribarsky, Wei Chen. Automatic Animation for Time-Varying Data Visualization[J]. Computer Graphics Forum, 2010, 29(7):2271–2280.
- [16] Mike Cammarano, Xin Dong, Bryan Chan, Je Klingner, Justin Talbot, Alon Halevey, Pat Han- rahan. Visualization of heterogeneous data[J]. IEEE Transactions on Visualization and Computer Graphics, 2007, 13(6):1200–1207.
- [17] Wong P C, Bergeron R D. 30 Years of Multidimensional Multivariate Visualization[C]//Scientific Visualization, 1994: 3-33.
- [18] Elmqvist N, Dragicevic P, Fekete J D. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation[J]. Visualization and Computer Graphics, IEEE Transactions on, 2008, 14(6): 1539-1148.
- [19] Rao R, Card S K. The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information[C]. Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 1994: 318-322.
- [20] Jacques Bertin. Semiology of graphics: diagrams, networks, maps[J]. 1983.
- [21] Jock Mackinlay. Automating the design of graphical presentations of relational information[J]. ACM Transactions on Graphics, 1986, 5(2):110–141.
- [22] J D Mackinlay. Automatic design of graphical presentations[D]. 1987.
- [23] Stuart Russell. Complete Guide to MRS. Technical report, DTIC Document, 1985.
- [24] William E Byrd Daniel P Friedman. Relational programming in minikanren: techniques, applications, and implementations[D]. Citeseer, 2009.
- [25] Claire E Alvis, Jeremiah J Willcock, Kyle M Carter, William E Byrd, Daniel P Friedman. ckanren minikanren with constraints[J]. 2011.
- [26] Jason Hemann Daniel P Friedman. μkanren: A minimal functional core for

- relational programming[J]. schemeworkshop.org, 2013.
- [27] Steven F Roth, Joe Mattis, Xavier Mesnar. Graphics and Natural Language as Components of Automatic Explanation[J]. ACM SIGCHI Bulletin, 1988, 20(1):76.
- [28] Steven F Roth, Joe Mattis. Data characterization for intelligent graphics presentation[C]. In the SIGCHI conference. New York, New York, USA: ACM Press, 1990, 193–200.
- [29] David Gotz, Zhen Wen. Behavior-driven visualization recommendation.[J]. IUI, 2009, 315–324.
- [30] Stephen M Casner. Task-analytic approach to the automated design of graphic presentations[J]. ACM Transactions on Graphics, 1991, 10(2):111–151.
- [31] Jill Freyne, Barry Smyth. Creating Visualizations: A Case-Based Reasoning Perspective.[J]. AICS, 2009, 6206(Chapter 11):82–91.
- [32] Stolte C, Tang D, Hanrahan P. Polaris: A system for query, analysis, and visualization of multidimensional relational databases[J]. Visualization and Computer Graphics, IEEE Transactions on, 2002, 8(1): 52-65.
- [33] Jock Mackinlay, Pat Hanrahan, Chris Stolte. Show Me: Automatic presentation for visual analysis[J]. IEEE Transactions on Visualization and Computer Graphics, 2007, 13(6):1137-1144.
- [34] NetEase Youdata[EB/OL]. <https://youdata.163.com/>.
- [35] NEV[EB/OL]. <http://nev.netease.com/>.
- [36] Leland Wilkinson. The Grammar of Graphics[M]. 2005, 2nd edition.
- [37] L Wilkinson, D J Rope, D B carr. The language of graphics[J]. Journal of Computational and Graphical Statistics, 2000, 9(3):530-543.
- [38] Hadley Wickham. A Layered Grammar of Graphics[J]. Journal of Computational and Graphical Statistics, 2010, 19(1):3-28.
- [39] ggplot2[EB/OL]. <http://ggplot2.org/>.
- [40] Kevin J Lynagh. A grammer of graphics[J]. 2012, 1-2.

- [41] Gramham Wills, Lelend Wilkinson. Autovis: automatic visualization[J]. Information Visualization, 2010, 9(1):47-69.
- [42] Vega: A Visualization Grammar[EB/OL]. <http://vega.github.io/>.
- [43] Chris Stolte, Diane Tang, Pat Hanrahan. Multiscale visualization using data cubes[J]. IEEE Transactions on Visualization and Computer Graphics, 2003, 9(2):176-187.
- [44] Xiaoyu Tane. Design and Implement of Automatic Graphic Inference Technology in Intelligent Visualization System[D]. 2016.
- [45] Pat Hanrahan. Visql: A language for query, analysis and visualization[C]. In Proceedings of the 2006 ACM SIGMOD. ACM, 2006, 721-721.
- [46] D J Duke, R Borgo, M Wallace, C runciman. Huge Data But Small Programs: Visualization Design via Multiple Embedded DSLs[C]. PADL, 2009.
- [47] Carl Nolan. Manipulate and query olap data using adomd and multidimensional expressions (mdx) provides a rich and powerful syntax for querying and manipulating the multidimensional[J]. Microsoft Systems Journal-US Edition, 1999, 29-40.
- [48] Don Box, Anders Hejlsberg. LinQ: .NET language-integrated query[J]. MSDN Developer Centre, 2007, 89.
- [49] James Cheney, Sam Lindley, Philip Wadler. A practical theory of language-integrated query[J]. ACM SIGPLAN Notices, 2013, 48(9):403-416.
- [50] C Stolte, P Hanrahan. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases[C]. In Information Visualization, 2000. InfoVis 2000. IEEE Symposium on. IEEE, 2000, 5-14.
- [51] Chris Stole, Diane Tang, Pat Hanran. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases[J]. Communications of the ACM, 2008, 51(11):75-84.

攻读硕士学位期间主要的研究成果

[1]

致谢

从二零一四年踏入浙大校门开始，转眼就到了即将毕业的时候。这段时间，我不断感受着、也接受着浙大的爱与鼓舞；我学会了很多知识，也学会了如何生活。在这里，不论是浓郁的学术氛围，还是敬爱的老师、可爱的同学，都让我受到许多感动。和大多数同学一样，吃过食堂，唱过校歌，学过通宵；有过兴奋，有过拼搏，有过不舍。可能是我太爱这里，感觉这里的一切都是那么熟悉和动人，竟让我沉醉在这份感动里不可自拔，却也不得不挥手诉说再见。

实验室导师组胡天磊老师、陈珂老师、陈刚老师、寿黎但老师、伍赛老师、江大伟老师是我最想要感谢的，无论是日常交流、或者是学术探讨，老师们都秉持着严谨的治学态度，又不乏风趣幽默。他们渊博的学识、高尚的师德、崇高的人格给我留下了深刻的印象，令我心向往之。本论文从选题到最终截稿的每一个步骤都是在实验室导师组的指导和帮助下完成的。在此谨向导师组致以崇高的敬意和感谢！

特别感谢网易有数开发团队对本文系统最后展现效果的帮助，特别是 Insight 小组的邓际锋、李诺、张淞、李超亚、徐慧、张佃鹏，他们在论文的实现过程中提供了许多帮助，在此表示感谢！

感谢实验室师兄师姐们对我的照顾，特别是唐晓瑜师姐对我的学习和人生目标都给出的极大的建议和帮助。感谢实验室同窗的关心和帮助，特别是王改革、张也、吴联坤在写论文过程中对我的鼓励和支持。感谢我的室友杨晓海、庞博、林炆平，以及浙大计算机学院 15 级和 16 级的研究生学弟学妹们，感谢有你们的陪伴和鼓舞！

再次感谢母校浙江大学的栽培，我会永远记得浙大求是校训，记得竺老校长的两个问题，记得大不自多的浙大校歌，记得在浙大相识的每一个人，记得浙大的繁花灿灿、一草一木。永世不忘。

署名：胡凡