大数据的最后一公里——数据可视化技术

中国工商银行软件开发中心上海研发部　黄玺磊

在金融行业中，将大数据技术运用在辅助决策、风险防控、客户画像、反欺诈等领域已经十分常见。在大数据技术价值链中，数据展现处在链条的末端，直接作用于最终用户决策的过程中，正是整个大数据技术的“最后一公里”。在各种数据展现方法中，数据可视化技术是最容易为人类所接受的表现形式。因此数据可视化技术的优劣将直接影响数据的最终应用与决策。

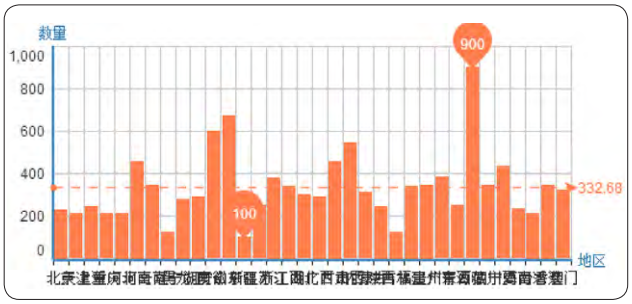


图 1　常规图表示例

一、数据可视化技术与传统图表比较

与数据可视化技术相比，常规的图表与图形仅仅能展现一两个维度的数据，数据的理解接受效率一般。如用常规柱状图表现的中国全部省级省级行政区数据，有时连文字都无法较好地展现（如图 1 所示）。

可视化技术在形式上能够灵活组合多维度数据描述数据场景（如地理位置与数值等结合分析），来提高同一幅图形上的数据容量并作多维结合分析。如相同的数据，以地图结合色度展现则十分清晰（如图 2 所示）。同时，可视化技术能够以模式化图形（如

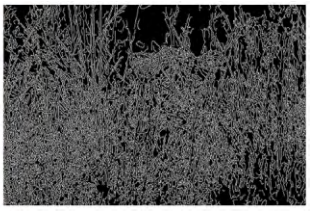
图 2　可视化技术示例一

股票 K 线图）来提供更高的数据阅读理解速度（如图 3 所示）。

图 3 可视化技术示例二

人类在漫长的大自然进化选择淘汰过程中积累了不少的视觉带宽优势，草原环境中的狮子，人眼识别十分迅速，而目前机器算法在场景识别上仍较困难（如图 4 所示）。因此，使用可视化技术较文字与数据表格等形式，更能够充分发挥人类的视觉带宽优势。

非洲草原中的一只狮子



目前较先进的 Canny 算法进行边缘识别后的图像

图 4 人眼识别与机器算法场景识别比较

二、数据可视化技术概述

1. 数据可视化定义

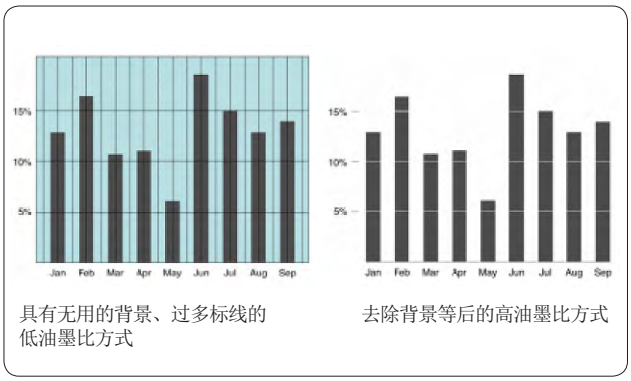
数据可视化是一种通过将数据编码为可视对象如点、线、颜色、位置关系、动态效果等，并将对象组成图形来传递数据信息的技术。其目的是以清晰且高效的方式将信息传递给用户，是利用人眼的感知能力对数据进行交互的可视化表达以增强数据认知的技术。笔者认为：数据可视化是一种充分利用人类视觉带宽（包括人机交互行为中的动态视觉）来提升人类数据理解能力、协助数据思考决策的技术。

2. 数据展现的发展历史

在人类数据展现技术的历史中，以行列方式组织的数据表格出现在约公元 2 世纪时。古罗马天文学家克劳狄乌斯·托勒密在其所著的长达 13 卷的《天文学大成》中发展了地心说，并详细记述了恒、行星运动及日、月蚀等的计算过程。他又从中选取了最有用的天文学计算内容，以表格形式重新出版为《实用天文表》（Handy Book）。这个表格式手册非常有名，从希腊文翻译成拉丁文、阿拉伯文、波斯语和梵文，从手抄变成机器印刷，传播了十几个世纪，远比《天文学大成》的传播范围广。

但以图形的方式来表现定量信息，直到 17 世纪才出现。这应归功于法国哲学家与数学家笛卡尔，是他首先在数学中发明了二维坐标系统。18 世纪后半叶，苏格兰工程师与经济学家 William Playfair 发明了许多至今我们仍常用的图形形式。他最早发明了从左至右的折线图来表现时间上的数值变化，又发明了柱状图，并在他离世前发明了饼图。

这些量化数据的图形逐步发展，但直到 20 世纪上半叶，这些方法本身并没有太大的进化。1967 年，Jacques Bertin 出版的《图像符号学》成为了后续可视化发展的奠基石，因为在这本书中他描述了信息可视化表达的直觉性、清晰性、精确性和效率性。1977 年，普林斯顿大学统计学教授约翰·图基在真正意义上带来了可视化在定量数据表达上的能力，他建立了一种称为 EDA 探索性数据分析的新统计模式，并将可视化技术运用其中。1983 年，著名的 Edward Tufte 编写了开创性的著作 *The Visual Display of Quantitative Information*，书中他提出了有效表达数据的“数据油墨比”的说法，并指出过去在可视化上的低效作法（数据油墨比 = 用于展现数据的墨水 / 图形上所使用的总墨水量 = 用于展现数据信息的不可再减少的墨水比例 =1.0– 图形上可被删去的墨水比例，如图 5 所示）。随即不久之后，被称为“信



具有无用的背景、过多标线的低油墨比方式　　去除背景等后的高油墨比方式

图 5　不同数据油墨比方式

图 6　*Is it Better to Rent or Buy?* 截图

息可视化”的学科正式进入了学术界中。

21 世纪，数据可视化已经被通俗化，但同时由于在商业软件（如 Excel）中的固化图表形式，数据表达常常很低效。值得庆幸的是，随着大数据、开源软件、前端浏览器等业界理念 / 技术的发展，数据可视化又重新以更丰富、更有力的形式回到人们的视野。

三、数据可视化设计特性

优秀的可视化设计特性包括“油墨比”高效、视角清晰、组合维度、对比恰当、动态可交互等，从而充分利用人类的视觉带宽，提升用户对数据的思考能力和理解效率，有效帮助数据决策。限于篇幅，下面以动态可交互为例做简单介绍。

在金融业中可交互场景是描述产品等的一种良好方式，在网点多屏设备等环境下可以有着广泛运用场景。*Is it Better to Rent or Buy?* 就是以数据驱动的文档（Data-Driven Document，D3）作者 Mike Bostock 的一个作品：通过以交互方式根据用户选择，决定在居住住房上是应投资购买还是租用的数据故事（如图 6 所示）。截图中展现了房价和预定居住年限，这两个对租房还是买房的决定性价格影响因素（这个数据故事中还包括了贷款利率、未来投资收益、交易税率等大大小小 21 个因素），左侧由用户滑动滑块对各因素给出选择，在右侧显示决定点价格的变化。这种用户可以自行操作、实时变化、辅助决策的可视化效果，是可视化中可交互性这一大特性的良好体现。

在出版业及宣传场合中，可视化设计所做的往往更多是一个故事化的数据展现。常见的如支付宝所做的十年账单等。通过数据故事结合 H5 进行宣传的手法已是互联网业常用的数据展现设计方法。而在金融业的分析管理应用中，数据故事是比较少见的一种可视化设计，而更多的是对分析数据的直接讲解。

四、在金融业使用的可视化集成仪表盘设计

以可视化集成仪表盘为目标导向的设计方式，将来自多个来源的数据展现，统一组织在一个屏幕内，这在包括金融业在内的商务智能（BI）中是一种传统且值得推荐的作法。目前在银行业内常见的就是类似此类仪表盘式样的设计，如按地域划分等级、交易热度类等的多图形结合的仪表盘设计（如图 7 所示）。又如警示类、目标达标类的运行指标的仪表盘设计（如图 8 所示）。

Stephen Few 的 *Information Dashboard Design* 对仪表盘的定义是：一种对重要信息的视觉展示，这些信息是为了达成一个或多个目标而被统一组织在一个屏幕内，以便能够一眼就就得到掌控。在这本可视化仪表盘设计的经典著作中，Stephen Few 提出了以下 13 种常见

图 7 仪表盘设计示例一

设计错误，这不单对于仪表盘，对于可视化设计也是很有借鉴价值的。

（1）超越一屏的边界（仪表盘应保持在一屏内）。

（2）没有为数据提供足够的上下文（数据应可比较、可说明变化程度，而非独立数字）。

（3）过度显示不必要的细节或精度（无需关注的细节数据不需要在仪表盘级别出现）。

（4）数据度量方式选择不当（应明确给出可观察的数据，而非由用户再计算）。

（5）图形呈现方式不当（如应该用饼图，却用柱状图）。

（6）引入无意义的变化（过多搭配难理解的图表类型或不能直接看出的比值等）

（7）使用糟糕设计的显示媒介（如乱用3D图形等）。

（8）不准确或不精确地显示数据（如故意非等差 / 非等比 / 非 0 值起始地显示数值 Y 轴等）。

（9）糟糕地布置数据位置（应赋予数据的优先级，由近及远地查看次序）。

（10）低效或没有强调重要的数据（如没有高亮 / 突出或者高亮 / 突出到处泛滥）。

（11）堆砌无用的装饰（加入不必要的、不说明数据的装饰）。

（12）滥用颜色（颜色使用的艺术）。

图 8 仪表盘设计示例二

（13）无吸引力的视觉显示。

五、可视化工具的分类与常见工具

可视化工具产品从使用方法的角度来说，可以分为工具开箱即用与代码开发使用方式两种；从适应范围来说，可以分为泛用类和领域专业类（如只关注于关系分析的 Gephi）；从产品底层架构来说，可以分为平台性工具（需要有整体性的数据建模过程、提供复杂分析服务）和插件性工具（直接对数据进行分析、简单数据处理）；从服务目标上来说，可以分为统计图形（一次开发为一个单一图形服务）和信息图（Infographic 为信息整体的数据故事服务）。下面举例说明几个常见的可视化工具。

1.D3.js（开发使用 – 泛用 – 插件性 – 统计图形）

目前业界最热门的可视化开发技术是 D3.js。D3 由就职于业界可视化标杆的《纽约时报》的 Mike Bostock 开发。D3 的含义为以数据驱动的文档，即数据是开发代码的中心。D3 以数据为中心，通过在数据上附加 HTML5 SVG 与 CSS、DOM 对象操作手段等来完成绘图工作，是一种先进高效的开发思想。

D3 社区相当活跃，既有各类的扩展插件、示例实现，也有自 D3 进一步开发形成的整合框架等。基于 D3.js 的开发往往都会先寻找已有的可视化效果，并以之为基础进行借鉴性的开发，而不需要自己从头设计从头开发。官网 Demo、作者的 bl.ock.org 代码网站、d3list 整合站、Stack Overflow 网站上的提问解答等，提供了上以万计的可直接借鉴的可视化效果与代码例子。

同时 D3 底层基于 JavaScript 代码的方式进行开发。在解决用户界面前端问题的领域，以 JavaScript 统一技术栈的作法日趋明显。使用 D3 在各种问题域与其他前端技术联合都会比较容易。如在地图领域，常被地图业界使用的 GeoJSON/TopoJSON 数据，与 D3 均存在良好的插件接口，因此以 D3 从各机构的公开数据基础上绘制地图或变换地图投影都很容易。

在开发形式上，D3 与 eCharts、FusionCharts 等以配置为主的开发组件不同，D3 以代码为主进行开发。这就决定了 D3 在刚开始学习时具有稍高的学习难度，但在掌握之后，其扩展能力、实现可视化细节的能力较配置为主的组件有着巨大的优势。同时值得注意的是，D3 只支持 IE9 及以上版本。

2.eCharts（开发使用 – 泛用 – 插件性 – 统计图形）

eCharts 是百度开源的 Enterprise Charts 商业产品图表库，基于纯 JavaScript 开发实现。由于其具有中文版本的例子与文档，因此在国内的使用范围较广。2013 年 6 月发布第一个版本，其最新版本是 3.2.3 版，但 3.x 版本变化较大，目前常用的仍是 2.x 版本。维护团队是百度 EFE 前端团队。从项目问题数量和解决速度来看，维护的稳定程度较 D3 等成熟框架来说不算高。

eCharts 使用一种配置式为主的代码开发模式，入手难度较 D3 框架低。对网页代码开发稍有一定了解就能进行配置开发，这使得一些非科技的业务人员也有可能使用。其封装的默认图形模板很多，配置开发效率也较 D3 的 SVG、DOM 操作开发更高。

eCharts 底层基于浏览器的 HTML5 Canvas，Canvas 是与 D3 的 SVG 特性不同的另一方向的 HTML5 绘图底层。Canvas 基于像素绘图，在数据量巨大时，比基于图形对象绘图的 SVG 性能更高，但对绘图内容难以进行对象化事件触发、对象级变换。对此百度 EFE 提供了一个 ZRender 底层在 Canvas 上封装部分对象化的实现，但距真正的 SVG 特性仍存在一些缺失。

不过，配置化的开发模式决定了开发使用者难以对其内部进行扩展和细节调整；在实现某些界面要求高的可视化需求时，一旦配置 API 未提供需要自行扩展、细节调整时，由于 eCharts 本身的配置与实现代码间封装层次很深，容易遇到技术难度陡升而无法扩展定制的瓶颈问题。

总之，eCharts 作为一种开源（目前开源但百度保留所有权利）、可快速开发、具备中文环境且支持低版

本 IE8 的可视化组件技术，值得在一些无过多定制需求、需快速开发的项目中使用。

3.Tableau Desktop（工具开箱即用 – 泛用 – 平台性 – 信息图）

Tableau 是一家总部设在美国华盛顿西雅图的软件公司。大家常说的 Tableau 实际上是指其六个产品中的 Tableau Desktop，是用于在 PC 桌面上独立运行的数据分析及编绘数据故事的商业可视化工具。

Tableau Desktop 的主旨是“每个人都能使用的分析工具”。在使用中，主要借助导入数据后的拖拉等操作就可以把数据展现为较美观的交互式图形故事。因此 Tableau Desktop 的使用成本很低，稍有数据分析制图经验的人员就可以较快速入手。

Tableau Desktop 能够引入很多类型的数据源，从 Excel、传统数据库到 Hadoop 都有支持接口，数据加载和处理的门槛不高，操作简单。但同时，这也是它的局限所在，Tableau 只支持对已经整理好的数据进行简单处理，并不能进行开发级的数据细节控制或可视化效果调节。

总体上来看，Tableau Desktop 这款商业软件主要专注的是结构化数据的快速可视化，使用者无需编程就能快速构建较为美观的数据可视化效果，并构建可交互的数据故事界面。但它也仅限于模式化的辅助分析，辅助人们进行视觉化的思考，并不如专业性分析工具的统计分析功能强大，也不具备类似 D3 的开发性可视化工具的扩展和表现细节能力。

4.Gephi（工具开箱即用 – 专用 – 平台性 – 统计图形）

Gephi 被誉为是复杂关系网络分析界的 PhotoShop。Gephi 以工具方式运行，而非开发方式，对使用人员开发能力要求低。Gephi 是标准的工具 + 开源插件模式，当人们需要新效果、功能时，可以在其官方的插件市场上寻找他人已完成的扩展效果。

Gephi 作为一个工具形式的可视化分析手段，不需要使用人员具备较强的大数据或前端开发经验，只需结合 Excel 就能得到较好的关联性分析数据效果，能够完成简单程度的分组聚类、热点权重、调色展现、过滤查询等功能。若结合 Excel 以外的更强大的数据分析工具，就能够对更大规模的数据进行处理。同时，其开源插件架构生态也是一个特色，可大大扩展其基本的使用能力。

但作为工具形式，Gephi 需人工加载数据处理，不适合在应用开发中分析动态变化的数据，而更适用于研究分析性的使用。同时其工具实现代码并不能简单直接修改，这就意味着在遇到无法适应的细节需求时的适应能力不如 D3、Raphael 等以代码开发方式为主的工具强。

Gephi 作为一种可视化分析工具，对于使用的人们来说，不需要开发经验，仅需要一定的数据分析能力，这是其最大特色，因此在关联数据分析领域中入门门槛低并被广泛使用。

六、可视化应用的需求特性与平台化

在可视化实际应用过程中，现阶段的需求常因业务人员受启发性因素而提出，如业务人员看到某个较好的微信效果图或其他公司产品图形后提出。因此其最初具体需求与效果的适用性往往不太明确。在进一步开发确认时，需求调整往往会十分细腻，业务人员针对线型、字体、配色、文字排布等细节效果的微调经常会深入、反复进行。业务人员也可能提出将可视化应用以工具方式提供的需求，使其能够直接以客户端方式快速装载离线数据以直接制作可视化图形。因此可视化应用的需求仍是较为复杂的。

同时，可视化的具体实现技术也各具特点，并没有“泛用”技术能够适应所有的可视化场景。选择与场景对应的“专用”技术能够事半功倍、效果良好地满足业务需求。如进行应用需求为可扩展过滤功能的关联性分析时，使用专用于进行关联性展现功能的 Sigma.js 开发

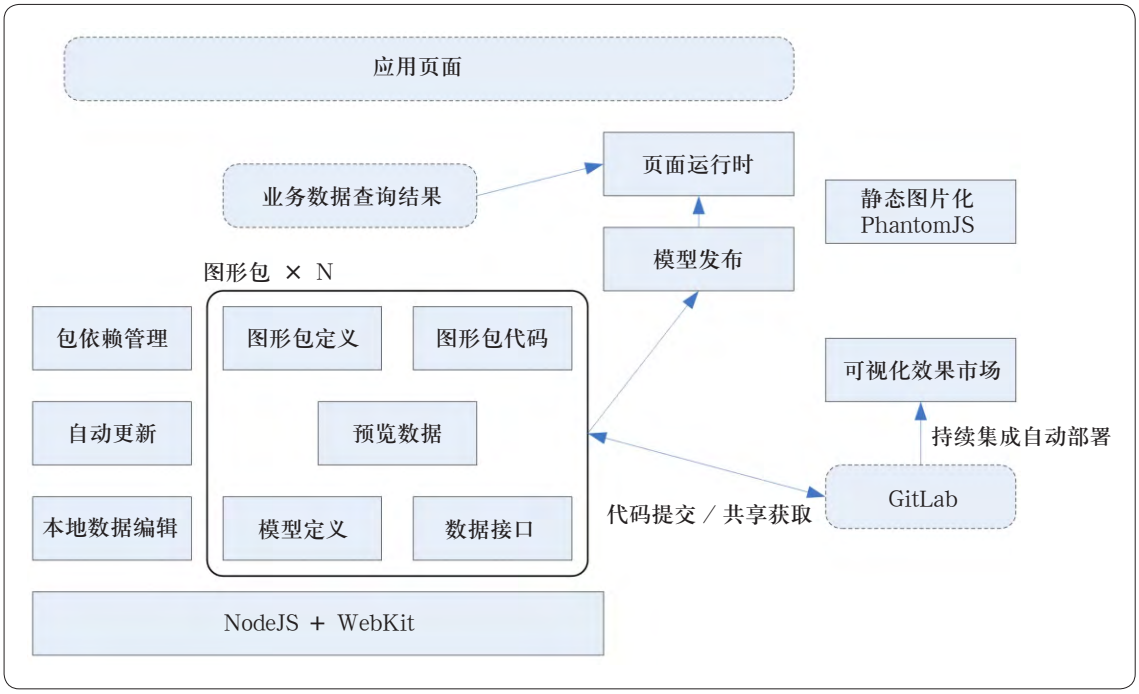


图 9 可视化平台示意

工具，只需要管理好数据，再使用其页面模板中的过滤数据特性，就能实现较好的关联性展现；若使用 eCharts 等泛用开发工具则需扩展过滤数据特性，其扩展实现较为复杂，还需自行设计过滤特性的页面交互方式。因此可视化在技术上有一定深度和广度。

针对可视化需求与技术上的特点，软件开发中心研发了可视化平台（如图 9 所示），同时支持网页端应用开发和在客户端的业务单机可视化使用。应用开发直接基于浏览器，而客户端以 NodeJS+WebKit 模式的 NW.js 技术为底层，以此实现不同平台下的代码统一复用。以包管理模式管理具体可视化效果插件，使得可视化效果开发人员能够以统一模式开发插件包。以内部开源 GitLab 模式管理插件包，建立持续集成模式，使得开发人员提交的插件包能自动持续集成到可视化效果市场网站上，方便业务用户直接浏览查看，启发业务需求使用。

可视化平台既满足了业务离线客户端使用的需求，也具备了效果网站直接启发业务需求；同时在开发上，能够以一套图形包同时适应客户端与应用页面开发；包以开源与持续集成自动部署方式自动发布。这样就完成了启发需求—业务体验—开源扩展开发—代码提交—自动集成回市场—回到启发需求的整个过程，形成良性生态闭环。

七、可视化技术前景展望

在金融业大数据的浪潮中，可视化以其独特的魅力始终占据着重要的一部分。在分析类应用场景或实时展现业务中，如基于图数据库的担保成圈分析、关联分析，实时交易监控等场景下，可视化都是与用户最终接触交互的“最后一公里”，发挥着不可替代的作用。

同时，随着银行实体网点的不断退出，客户“网上”化比例升高，而客户端设备尤其是移动端设备性能的提升，这些条件都将可视化运用推到金融业的直接对客领域。数据可视化技术将在用户端绽放出愈发璀璨的光芒。FCC