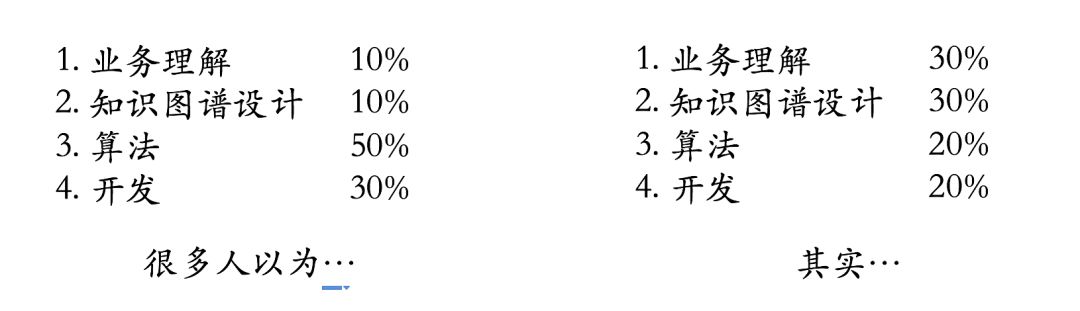
# 金融知识图谱的搭建

接下来我们看一个实际的具体案例，讲解怎么一步步搭建可落地的金融风控领域的知识图谱系统。 首先需要说明的一点是，有可能不少人认为搭建一个知识图谱系统的重点在于算法和开发。但事实并不是想象中的那样，其实最重要的核心在于对业务的理解以及对知识图谱本身的设计，这就类似于对于一个业务系统，数据库表的设计尤其关键，而且这种设计绝对离不开对业务的深入理解以及对未来业务场景变化的预估。 当然，在这里我们先不讨论数据的重要性。



一个完整的知识图谱的构建包含以下几个步骤：

1. 定义具体的业务问题

2. 数据的收集 & 预处理

3. 知识图谱的设计

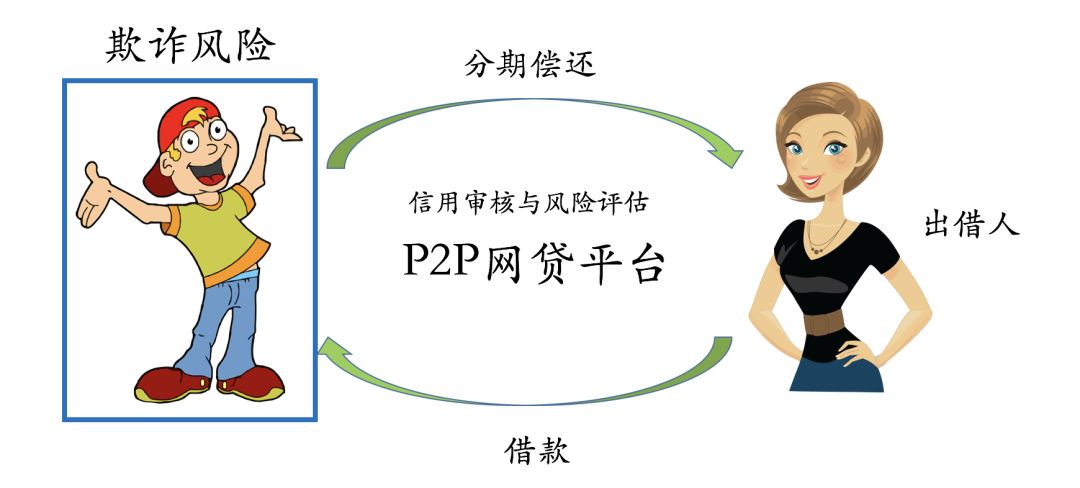
4. 把数据存入知识图谱

5. 上层应用的开发，以及系统的评估

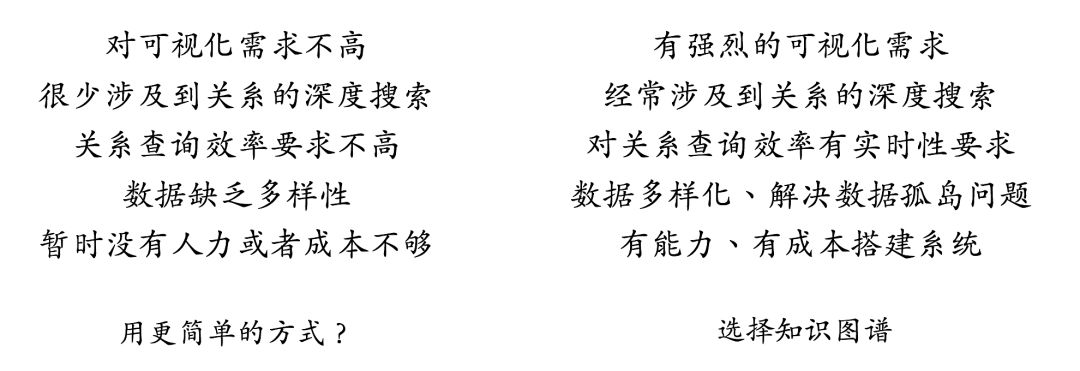
下面我们就按照这个流程来讲一下每个步骤所需要做的事情以及需要思考的问题。

## ****定义具体的业务问题****

在P2P网贷环境下，最核心的问题是风控，也就是怎么去评估一个借款人的风险。在线上的环境下，欺诈风险尤其为严重，而且很多这种风险隐藏在复杂的关系网络之中，而且知识图谱正好是为这类问题所设计的，所以我们“有可能”期待它能在欺诈，这个问题上带来一些价值。



在进入下一个话题的讨论之前，**要明确的一点是，对于自身的业务问题到底需不需要知识图谱系统的支持**。因为在很多的实际场景，即使对关系的分析有一定的需求，实际上也可以利用传统数据库来完成分析的。所以为了避免使用知识图谱而选择知识图谱，以及更好的技术选型，以下给出了几点总结，供参考。



## ****数据收集 & 预处理****

下一步就是要确定数据源以及做必要的数据预处理。针对于数据源，我们需要考虑以下几点：

1. 我们已经有哪些数据？

2. 虽然现在没有，但有可能拿到哪些数据？

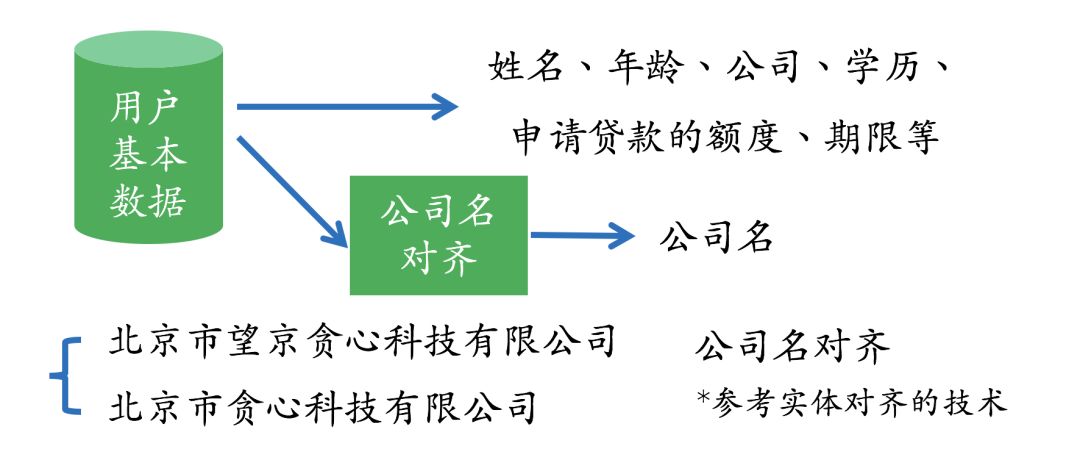
3. 其中哪部分数据可以用来降低风险？

4. 哪部分数据可以用来构建知识图谱？

在这里需要说明的一点是，并不是所有跟反欺诈相关的数据都必须要进入知识图谱，对于这部分的一些决策原则在接下来的部分会有比较详细的介绍。

对于反欺诈，有几个数据源是我们很容易想得到的，包括用户的基本信息、行为数据、运营商数据、网络上的公开信息等等。假设我们已经有了一个数据源的列表清单，则下一步就要看哪些数据需要进一步的处理，比如对于非结构化数据我们或多或少都需要用到跟自然语言处理相关的技术。 用户填写的基本信息基本上会存储在业务表里，除了个别字段需要进一步处理，很多字段则直接可以用于建模或者添加到知识图谱系统里。对于行为数据来说，我们则需要通过一些简单的处理，并从中提取有效的信息比如“用户在某个页面停留时长”等等。 对于网络上公开的网页数据，则需要一些信息抽取相关的技术。

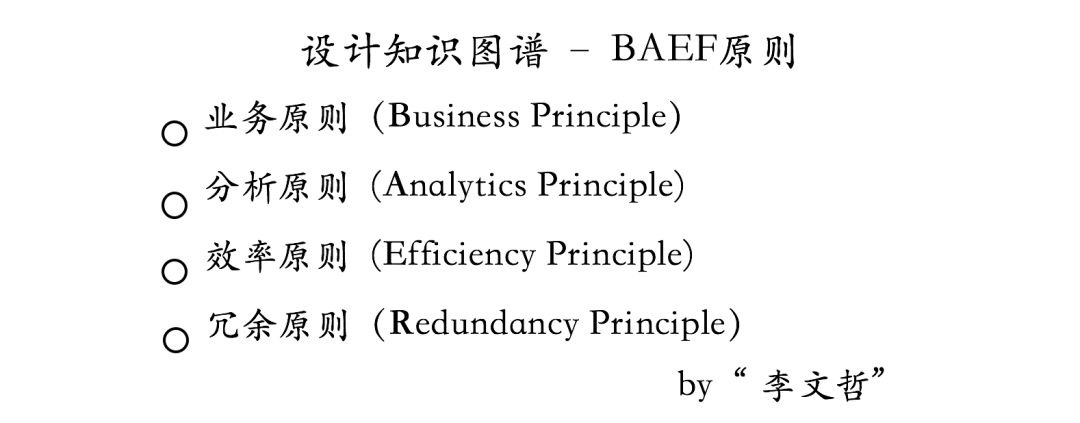
举个例子，对于用户的基本信息，我们很可能需要如下的操作。一方面，用户信息比如姓名、年龄、学历等字段可以直接从结构化数据库中提取并使用。但另一方面，对于填写的公司名来说，我们有可能需要做进一步的处理。比如部分用户填写“北京贪心科技有限公司”，另外一部分用户填写“北京望京贪心科技有限公司”，其实指向的都是同一家公司。所以，这时候我们需要做公司名的对齐，用到的技术细节可以参考前面讲到的实体对齐技术。﻿



## ****知识图谱的设计****

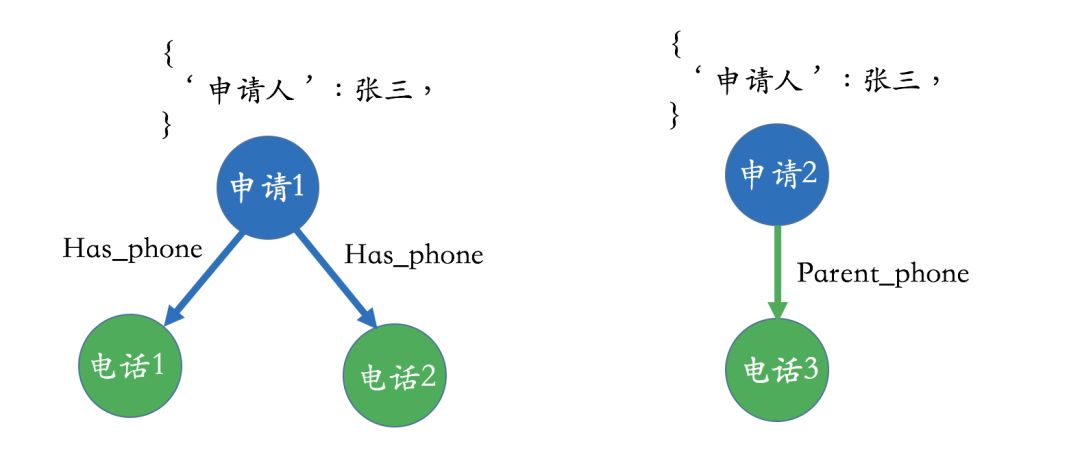
图谱的设计是一门艺术，不仅要对业务有很深的理解、也需要对未来业务可能的变化有一定预估，从而设计出最贴近现状并且性能高效的系统。在知识图谱设计的问题上，我们肯定会面临以下几个常见的问题：**1. 需要哪些实体、关系和属性？ 2.  哪些属性可以做为实体，哪些实体可以作为属性？ 3. 哪些信息不需要放在知识图谱中？**

基于这些常见的问题，我们从以往的设计经验中抽象出了一系列的设计原则。这些设计原则就类似于传统数据库设计中的范式，来引导相关人员设计出更合理的知识图谱系统，同时保证系统的高效性。

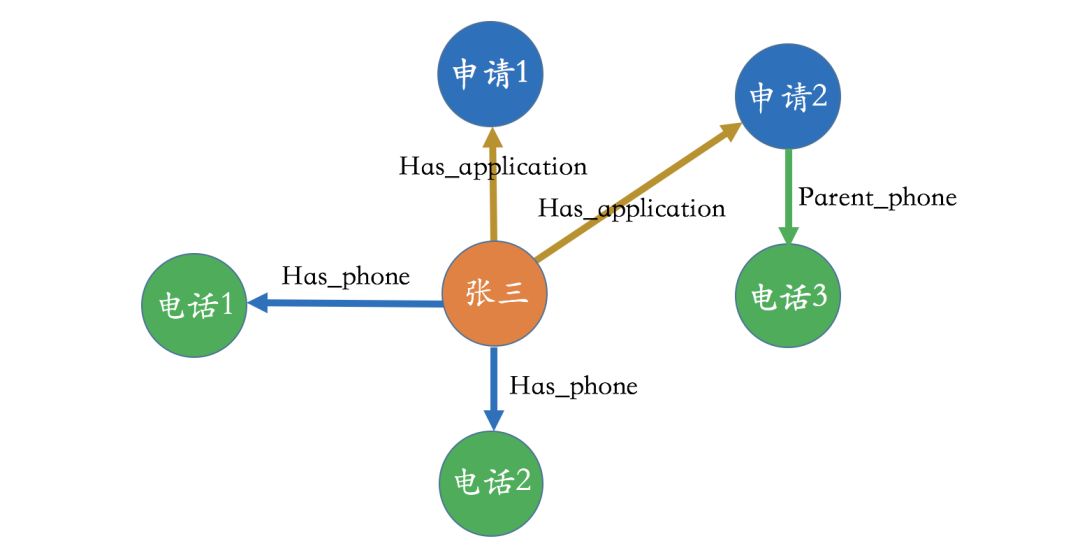


接下来，我们举几个简单的例子来说明其中的一些原则。 首先是，业务原则（Business Principle），它的含义是 “**一切要从业务逻辑出发，并且通过观察知识图谱的设计也很容易推测其背后业务的逻辑，而且设计时也要想好未来业务可能的变化**”。

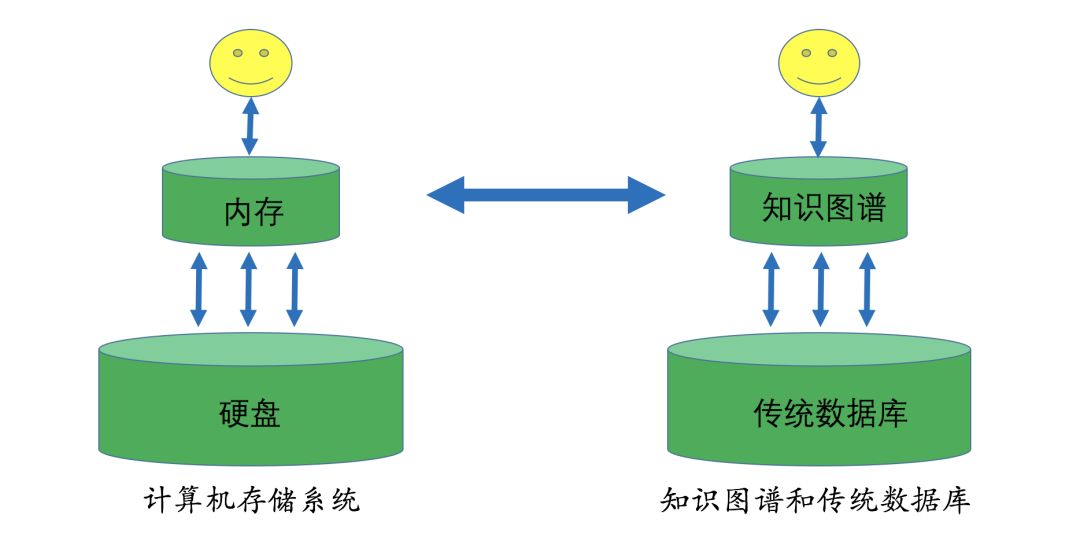
举个例子，可以观察一下下面这个图谱，并试问自己背后的业务逻辑是什么。通过一番观察，其实也很难看出到底业务流程是什么样的。做个简单的解释，这里的实体-“申请”意思就是application，如果对这个领域有所了解，其实就是进件实体。在下面的图中，申请和电话实体之间的“has\_phone”，“parent phone”是什么意思呢？



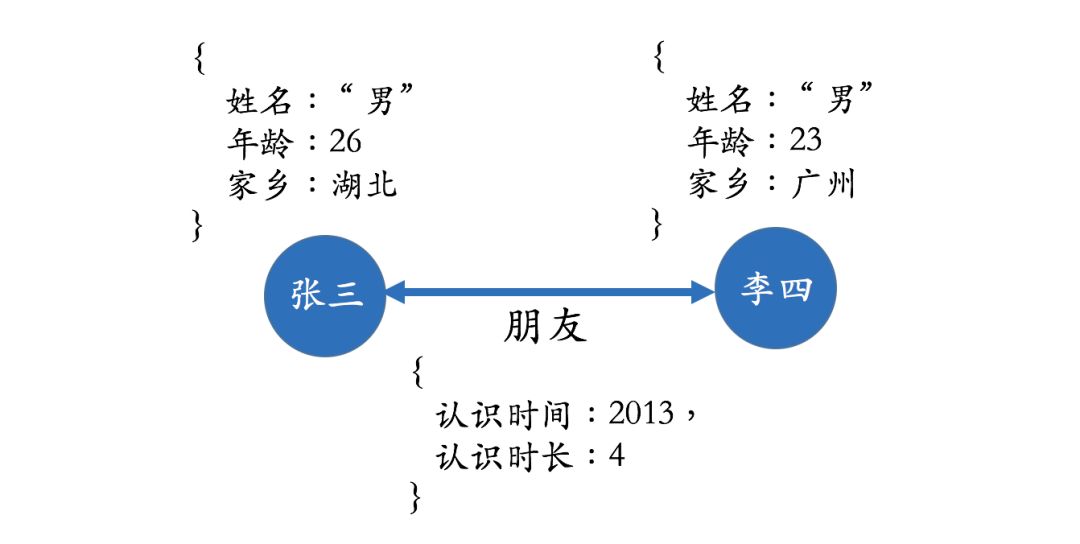
﻿接下来再看一下下面的图，跟之前的区别在于我们把申请人从原有的属性中抽取出来并设置成了一个单独的实体。在这种情况下，整个业务逻辑就变得很清晰，我们很容易看出张三申请了两个贷款，而且张三拥有两个手机号，在申请其中一个贷款的时候他填写了父母的电话号。总而言之，一个好的设计很容易让人看到业务本身的逻辑。



﻿接下来再看一个原则叫做效率原则（Efficiency Principle）。**效率原则让知识图谱尽量轻量化、并决定哪些数据放在知识图谱，哪些数据不需要放在知识图谱**。在这里举一个简单的类比，在经典的计算机存储系统中，我们经常会谈论到内存和硬盘，内存作为高效的访问载体，作为所有程序运行的关键。这种存储上的层次结构设计源于数据的局部性-“locality”，也就是6.



比如在下面的知识图谱中，我们完全可以把一些信息比如“年龄”，“家乡”放到传统的关系型数据库当中，因为这些数据对于：a. 分析关系来说没有太多作用   b.  访问频率低，放在知识图谱上反而影响效率



另外，从分析原则（Analytics Principle）的角度，我们不需要把跟关系分析无关的实体放在图谱当中；从冗余原则（Redundancy Principle）的角度，有些重复性信息、高频信息可以放到传统数据库当中。

## ****1.4 把数据存入知识图谱****

存储上我们要面临存储系统的选择，但由于我们设计的知识图谱带有属性，图数据库可以作为首选。但至于选择哪个图数据库也要看业务量以及对效率的要求。如果数据量特别庞大，则Neo4j很可能满足不了业务的需求，这时候不得不去选择支持准分布式的系统比如OrientDB, JanusGraph等，或者通过效率、冗余原则把信息存放在传统数据库中，从而减少知识图谱所承载的信息量。 通常来讲，对于10亿节点以下规模的图谱来说Neo4j已经足够了。

## ****1.5 上层应用的开发****

等我们构建好知识图谱之后，接下来就要使用它来解决具体的问题。对于风控知识图谱来说，首要任务就是挖掘关系网络中隐藏的欺诈风险。**从算法的角度来讲，有两种不同的场景：一种是基于规则的；另一种是基于概率的**。鉴于目前AI技术的现状，基于规则的方法论还是在垂直领域的应用中占据主导地位，但随着数据量的增加以及方法论的提升，基于概率的模型也将会逐步带来更大的价值。

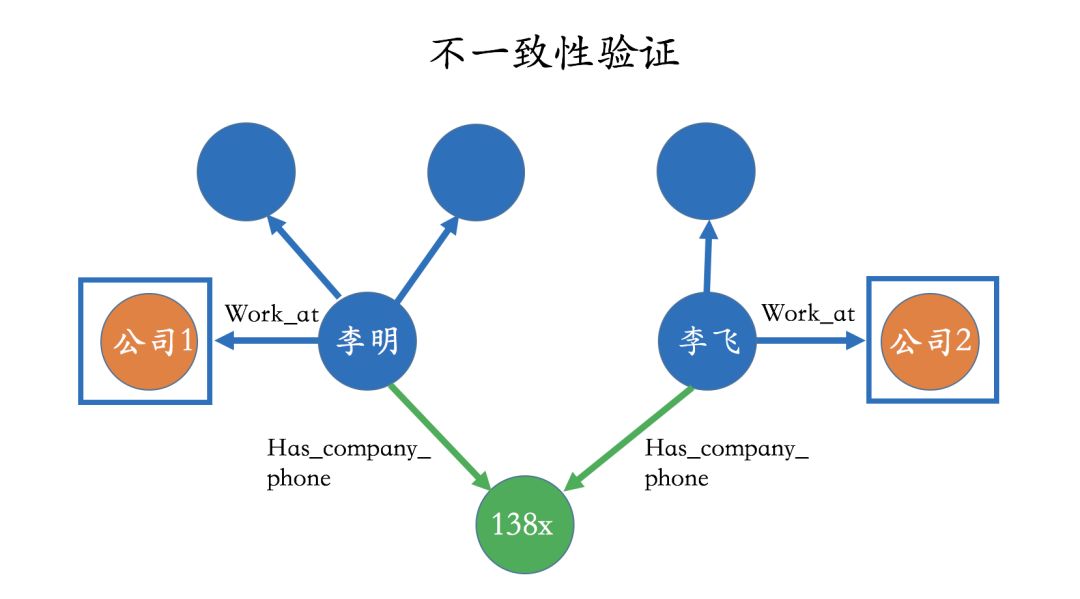
所以这部分数据可以放到内存中来提升访问的效率。 类似的逻辑也可以应用到知识图谱的设计上：我们把常用的信息存放在知识图谱中，把那些访问频率不高，对关系分析无关紧要的信息放在传统的关系型数据库当中。 **效率原则的核心在于把知识图谱设计成小而轻的存储载体。**

### 1.5.1 基于规则的方法论

首先，我们来看几个基于规则的应用，分别是不一致性验证、基于规则的特征提取、基于模式的判断。

**不一致性验证**

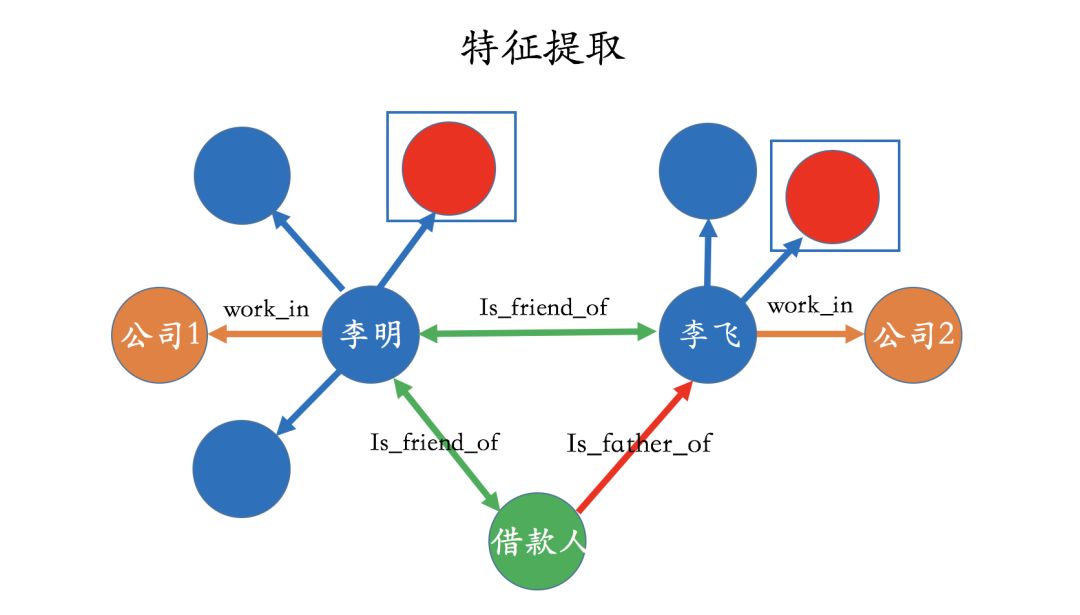
为了判断关系网络中存在的风险，一种简单的方法就是做不一致性验证，也就是通过一些规则去找出潜在的矛盾点。这些规则是以人为的方式提前定义好的，所以在设计规则这个事情上需要一些业务的知识。比如在下面的这个图中，李明和李飞两个人都注明了同样的公司电话，但实际上从数据库中判断这俩人其实在不同的公司上班，这就是一个矛盾点。 类似的规则其实可以有很多，不在这里一一列出。



﻿

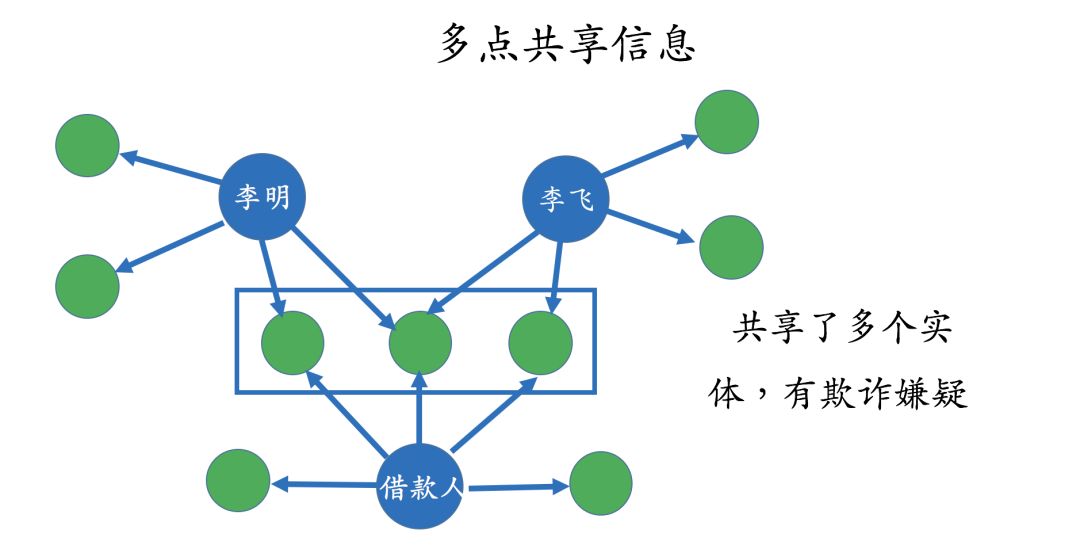
**基于规则提取特征**

我们也可以基于规则从知识图谱中提取一些特征，而且这些特征一般基于深度的搜索比如2度，3度甚至更高维度。比如我们可以问一个这样的问题：“申请人二度关系里有多少个实体触碰了黑名单？”，从图中我们很容观察到二度关系中有两个实体触碰了黑名单（黑名单由红色来标记）。等这些特征被提取之后，一般可以作为风险模型的输入。在此还是想说明一点，如果特征并不涉及深度的关系，其实传统的关系型数据库则足以满足需求。

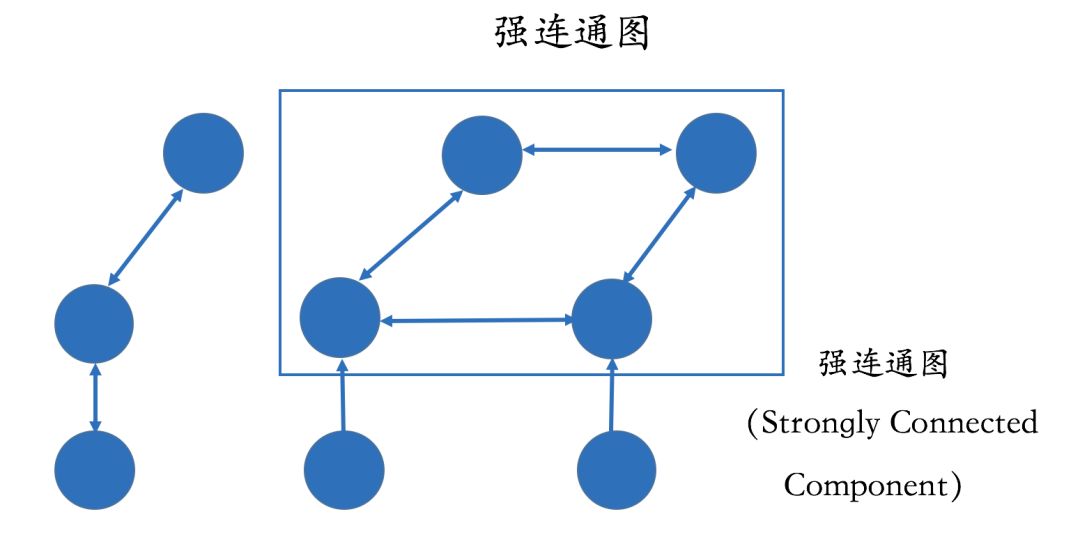


**基于模式的判断**

这种方法比较适用于找出团体欺诈，它的核心在于通过一些模式来找到有可能存在风险的团体或者子图（sub-graph），然后对这部分子图做进一步的分析。 这种模式有很多种，在这里举几个简单的例子。 比如在下图中，三个实体共享了很多其他的信息，我们可以看做是一个团体，并对其做进一步的分析。



﻿再比如，我们也可以从知识图谱中找出强连通图，并把它标记出来，然后做进一步风险分析。强连通图意味着每一个节点都可以通过某种路径达到其他的点，也就说明这些节点之间有很强的关系。

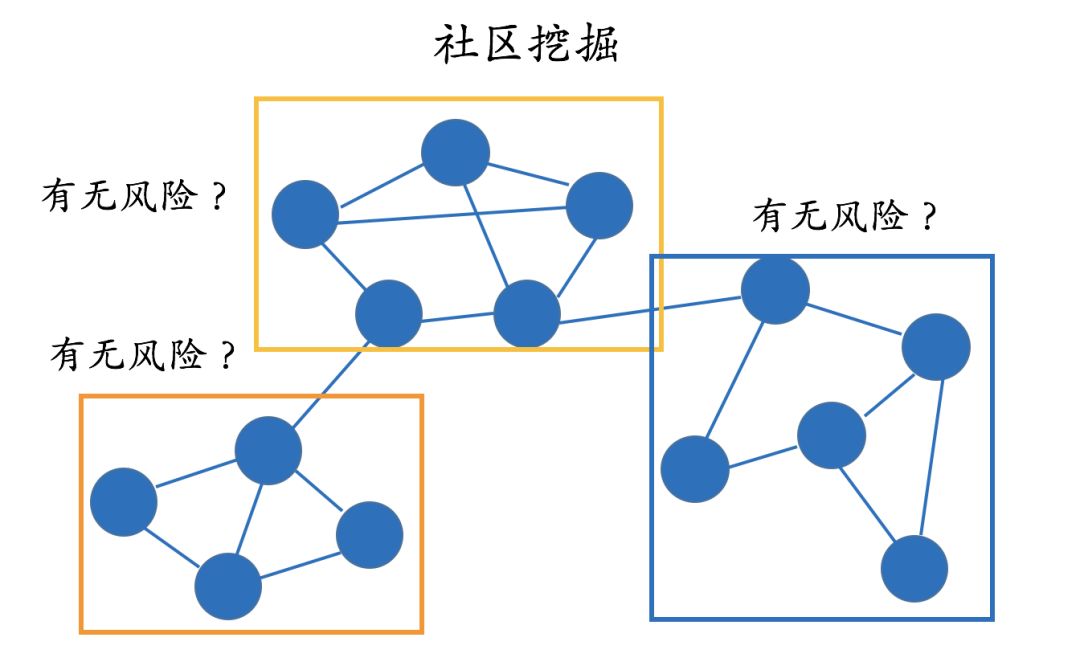


### 1.5.2 基于概率的方法

除了基于规则的方法，也可以使用概率统计的方法。 比如社区挖掘、标签传播、聚类等技术都属于这个范畴。 对于这类技术，在本文里不做详细的讲解，感兴趣的读者可以参考相关文献。

社区挖掘算法的目的在于从图中找出一些社区。对于社区，我们可以有多种定义，但直观上可以理解为社区内节点之间关系的密度要明显大于社区之间的关系密度。下面的图表示社区发现之后的结果，图中总共标记了三个不同的社区。一旦我们得到这些社区之后，就可以做进一步的风险分析。

由于社区挖掘是基于概率的方法论，好处在于不需要人为地去定义规则，特别是对于一个庞大的关系网络来说，定义规则这事情本身是一件很复杂的事情。



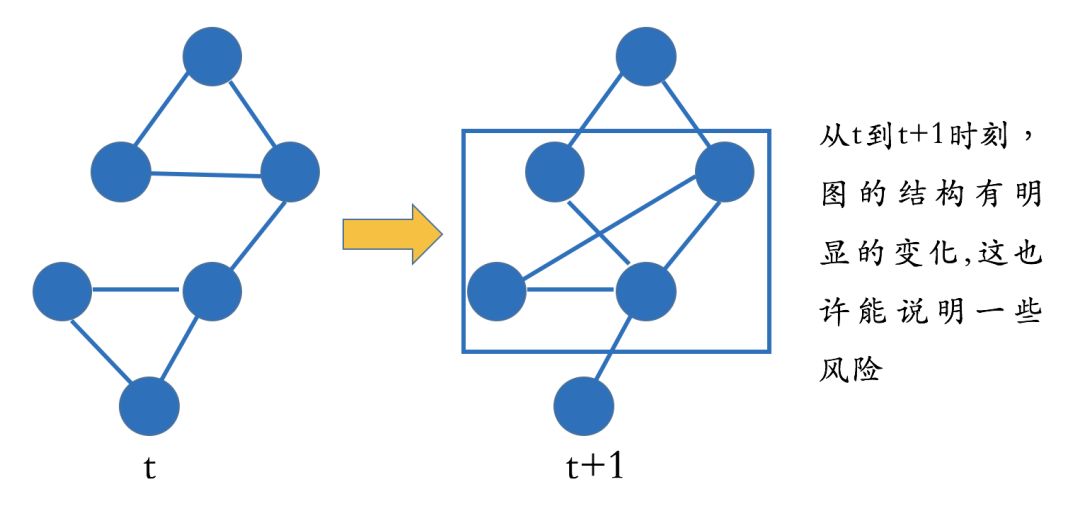
标签传播算法的核心思想在于节点之间信息的传递。这就类似于，跟优秀的人在一起自己也会逐渐地变优秀是一个道理。因为通过这种关系会不断地吸取高质量的信息，最后使得自己也会不知不觉中变得更加优秀。具体细节不在这里做更多解释。

相比规则的方法论，基于概率的方法的缺点在于：需要足够多的数据。如果数据量很少，而且整个图谱比较稀疏（Sparse），基于规则的方法可以成为我们的首选。尤其是对于金融领域来说，数据标签会比较少，这也是为什么基于规则的方法论还是更普遍地应用在金融领域中的主要原因。

### 1.5.3 基于动态网络的分析

以上所有的分析都是基于静态的关系图谱。所谓的静态关系图谱，意味着我们不考虑图谱结构本身随时间的变化，只是聚焦在当前知识图谱结构上。然而，我们也知道图谱的结构是随时间变化的，而且这些变化本身也可以跟风险有所关联。

在下面的图中，我们给出了一个知识图谱T时刻和T+1时刻的结构，我们很容易看出在这两个时刻中间，图谱结构（或者部分结构）发生了很明显的变化，这其实暗示着潜在的风险。那怎么去判断这些结构上的变化呢？ 感兴趣的读者可以查阅跟“dynamic network mining”相关的文献。

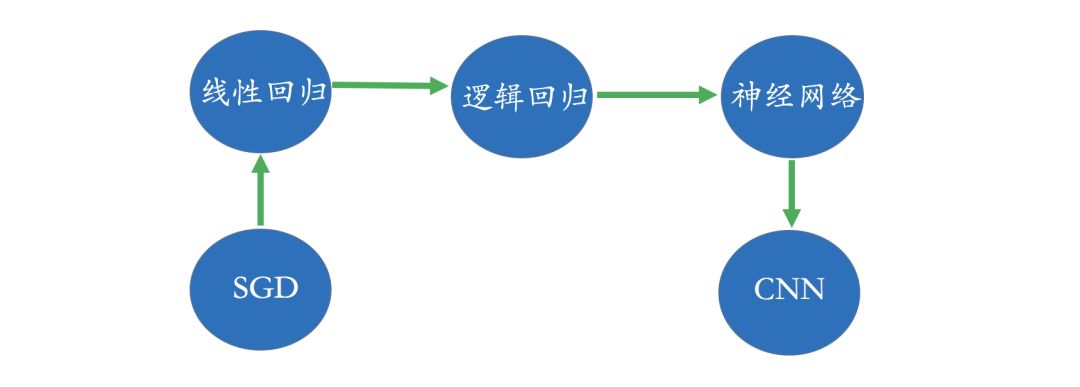


﻿

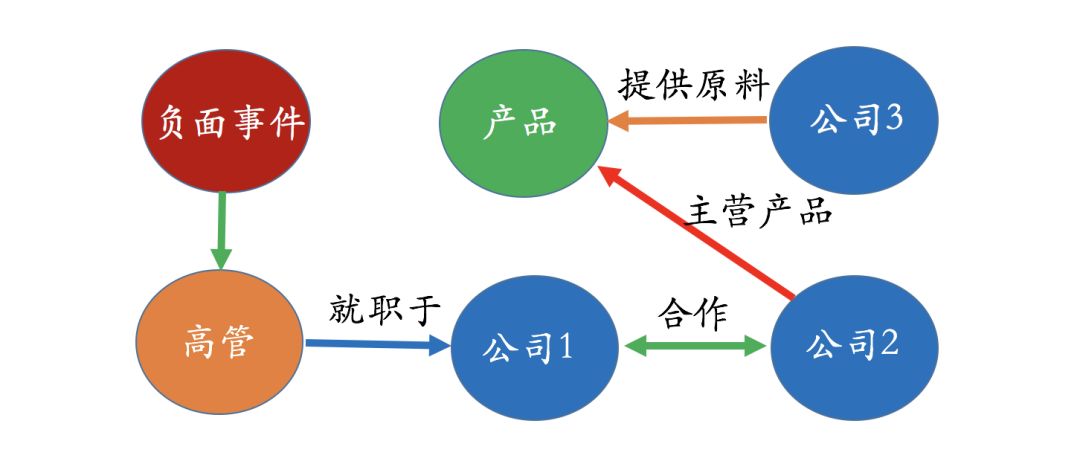
## ****1.6 知识图谱在其他行业中的应用****

除了金融领域，知识图谱的应用可以涉及到很多其他的行业，包括医疗、教育、证券投资、推荐等等。其实，只要有关系存在，则有知识图谱可发挥价值的地方。 在这里简单举几个垂直行业中的应用。

比如对于教育行业，我们经常谈论个性化教育、因材施教的理念。其核心在于理解学生当前的知识体系，而且这种知识体系依赖于我们所获取到的数据比如交互数据、评测数据、互动数据等等。为了分析学习路径以及知识结构，我们则需要针对于一个领域的概念知识图谱，简单来讲就是概念拓扑结构。在下面的图中，我们给出了一个非常简单的概念图谱：比如为了学习逻辑回归则需要先理解线性回归；为了学习CNN，得对神经网络有所理解等等。所有对学生的评测、互动分析都离不开概念图谱这个底层的数据。



﻿ 在证券领域，我们经常会关心比如“一个事件发生了，对哪些公司产生什么样的影响？” 比如有一个负面消息是关于公司1的高管，而且我们知道公司1和公司2有种很密切的合作关系，公司2有个主营产品是由公司3提供的原料基础上做出来的。



其实有了这样的一个知识图谱，我们很容易回答哪些公司有可能会被这次的负面事件所影响。当然，仅仅是“有可能”，具体会不会有强相关性必须由数据来验证。所以在这里，知识图谱的好处就是把我们所需要关注的范围很快给我们圈定。接下来的问题会更复杂一些，比如既然我们知道公司3有可能被这次事件所影响，那具体影响程度有多大？ 对于这个问题，光靠知识图谱是很难回答的，必须要有一个影响模型、以及需要一些历史数据才能在知识图谱中做进一步推理以及计算。

## ****1.7 实践上的几点建议****

首先，知识图谱是一个比较新的工具，它的主要作用还是在于分析关系，尤其是深度的关系。所以在业务上，首先要确保它的必要性，其实很多问题可以用非知识图谱的方式来解决。

知识图谱领域一个最重要的话题是知识的推理。 而且知识的推理是走向强人工智能的必经之路。但很遗憾的，目前很多语义网络的角度讨论的推理技术（比如基于深度学习，概率统计）很难在实际的垂直应用中落地。其实目前最有效的方式还是基于一些规则的方法论，除非我们有非常庞大的数据集。

最后，还是要强调一点，**知识图谱工程本身还是业务为重心，以数据为中心。不要低估业务和数据的重要性。**

## ****1.8 结语****

知识图谱是一个既充满挑战而且非常有趣的领域。只要有正确的应用场景，对于知识图谱所能发挥的价值还是可以期待的。我相信在未来不到2，3年时间里，知识图谱技术会普及到各个领域当中。

文章来源：<https://www.jiqizhixin.com/articles/2018-06-20-4>