

# 知识图谱推理

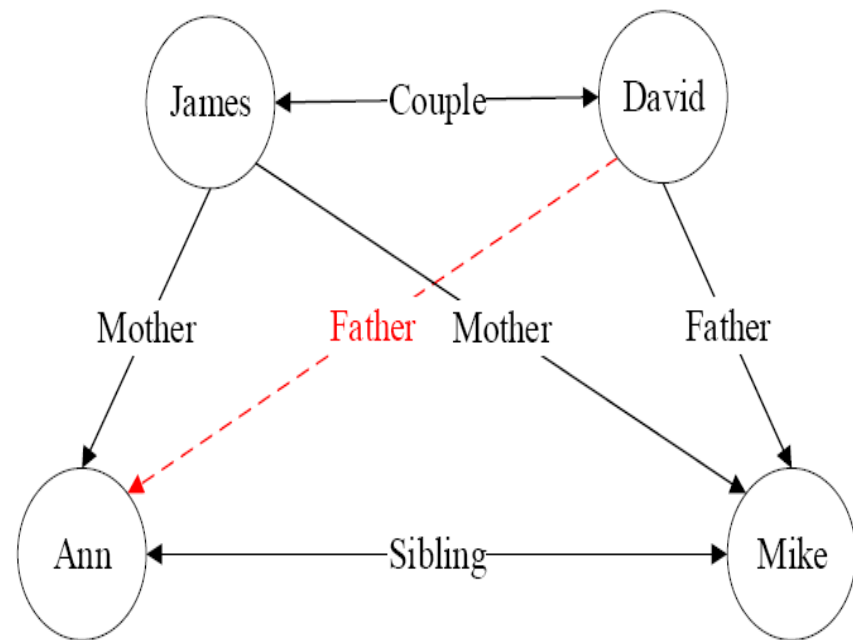
2022.03.10

## 目录

- 1. 理论知识
  - 1.1 基本概念
  - 1.2 归纳学习
  - 1.3 路径排序推理（了解）
  - 1.4 基于分布式表示的知识推理（了解）
- 2. 实验任务（选做）
  - 利用FOIL（First Order Inductive Learner）对给定的知识图谱实现规则推理，并经过“演绎”得到新的知识

## 1.1 基本概念

- 知识图谱可视为包含多种关系的图。在图中，每个节点是一个实体（如人名、地名、事件和活动等），任意两个节点之间的边表示这两个节点之间存在的关系。
- 一般而言，可将知识图谱中任意两个相连节点及其连接边表示成一个三元组（*triplet*），即（*left\_node*, *relation*, *right\_node*），例：（David, *Father*, Mike）。
- 这种三元组也可以表示为一阶逻辑（first order logic, FOL）的形式。



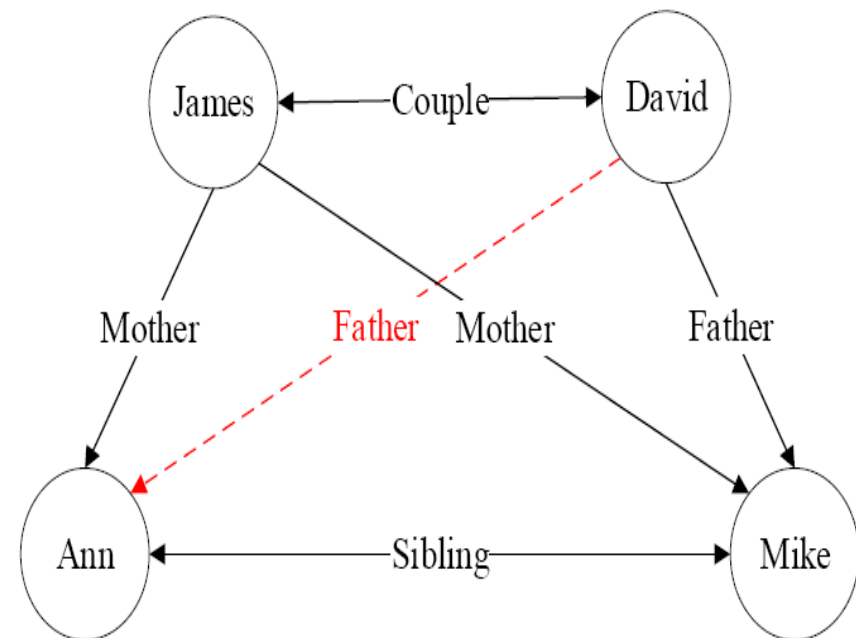
一个简单的家庭关系知识图谱

## 1.1 基本概念

- 因此，可用一阶逻辑的形式来描述三元组  $\langle \text{James}, \text{Couple}, \text{David} \rangle$ ，即  $\text{Couple}(\text{James}, \text{David})$ 。
- $\text{Couple}(x, y)$  是一阶谓词， $\text{Couple}$  是图中实体间具有的关系， $x$  和  $y$  是谓词变量。

- 主要任务：关系推理（链路预测）

从图中已有关系可推知David和Ann具有父女关系，但这一关系在图中初始图（无红线）中并不存在，是需要推理的目标，即获得新的知识。



一个简单的家庭关系知识图谱

## 1.2 归纳学习

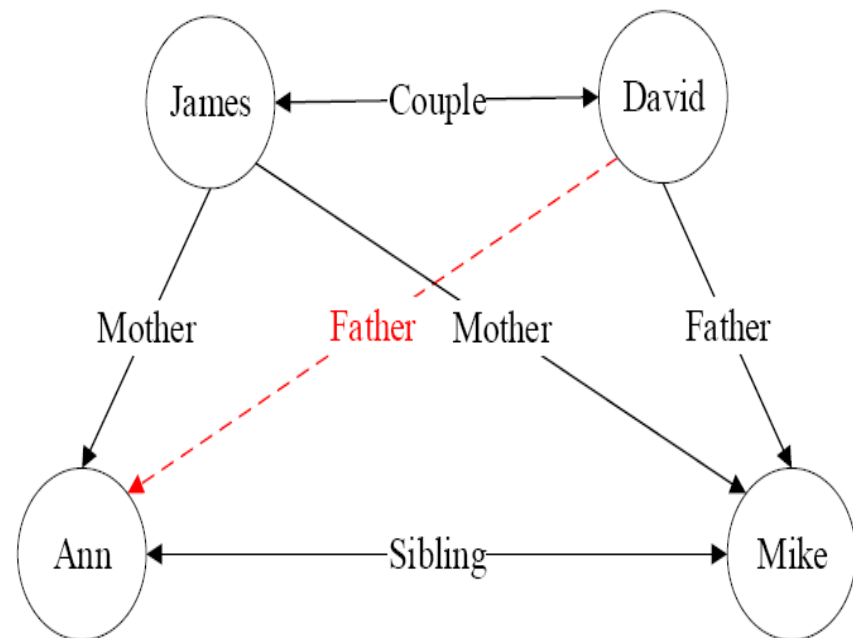
- 问题：如何从知识图谱中推理得到

*father(David, Ann)*



$(\forall x)(\forall y)(\forall z)(Mother(z, y) \wedge Couple(x, z))$

如果能够学习得到这条规则，该有多好？  
(从具体例子中学习，这是归纳推理的范畴)



一个简单的家庭关系知识图谱

## 1.2 归纳学习

### ■ 归纳逻辑程序设计 (inductive logic programming, ILP) 算法

- 归纳逻辑程序设计 (ILP) 是机器学习和逻辑程序设计交叉领域的研究内容。
- ILP使用一阶谓词逻辑进行知识表示，通过修改和扩充逻辑表达式对现有知识归纳，完成推理任务。
- 作为ILP的代表性方法，FOIL (First Order Inductive Learner) 通过**序贯覆盖**实现规则推理。

## 1.2 归纳学习

■ FOIL (First Order Inductive Learner) 算法

■ 推理手段: *positive examples* + *negative examples* + *background knowledge examples*  $\Rightarrow$  **hypothesis**

$$(\forall x)(\forall y)(\forall z)(Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y))$$



前提约束谓词  
(学习得到)



目标谓词  
(已知)

## 1.2 归纳学习-FOIL

### ■ FOIL (First Order Inductive Learner) 算法

*positive examples* + *negative examples* + *background knowledge examples*  $\Rightarrow$  **hypothesis**

#### ■ 依据目标谓词，可以得到以下：

- 正例
- 反例
- 背景知识

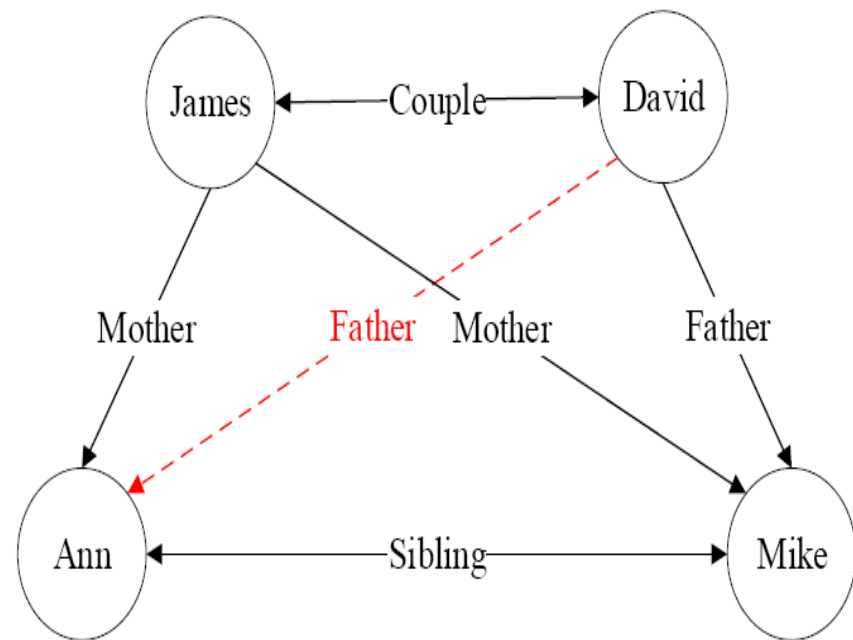


## 1.2 归纳学习-FOIL

■ 目标谓词:  $Father(x, y)$

■ 构造正例

- 在知识图谱中找到目标谓词的实例化结果，如右图，只有一个正例  $Father(David, Mike)$



一个简单的家庭关系知识图谱

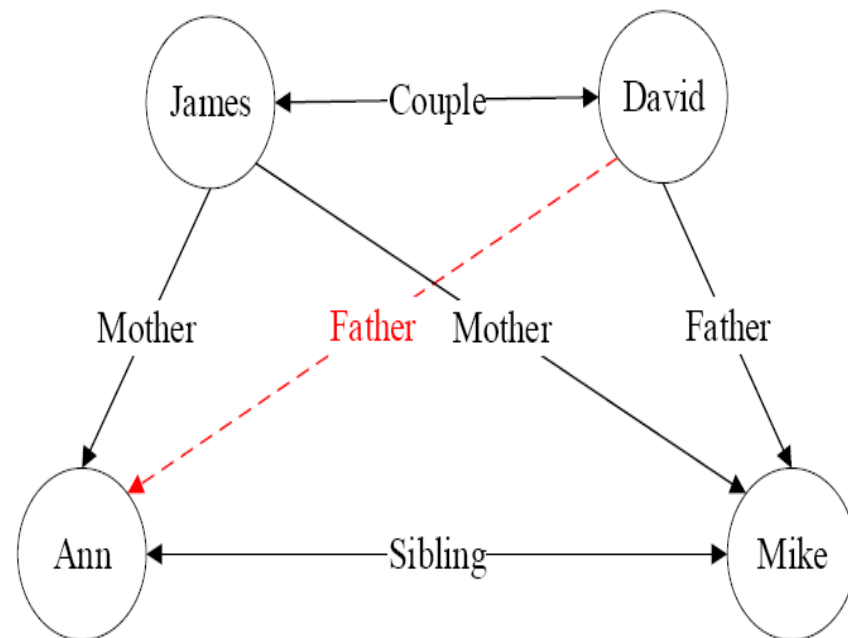
## 1.2 归纳学习-FOIL

■ 目标谓词:  $Father(x, y)$

■ 构造反例

■ 反例在知识图谱中一般不会显式给出，但可从知识图谱中构造出来。如从知识图谱中已经知道  $Couple(David, James)$  成立，则  $Father(David, James)$  可作为目标谓词  $P$  的一个反例，记为  $\neg Father(David, James)$ 。

■ 原则：只能在已知两个实体的关系且确定其关系与目标谓词相悖时，才能将这两个实体用于构建目标谓词的反例，而不能在不知两个实体是否满足目标谓词前提下将它们来构造目标谓词的反例。



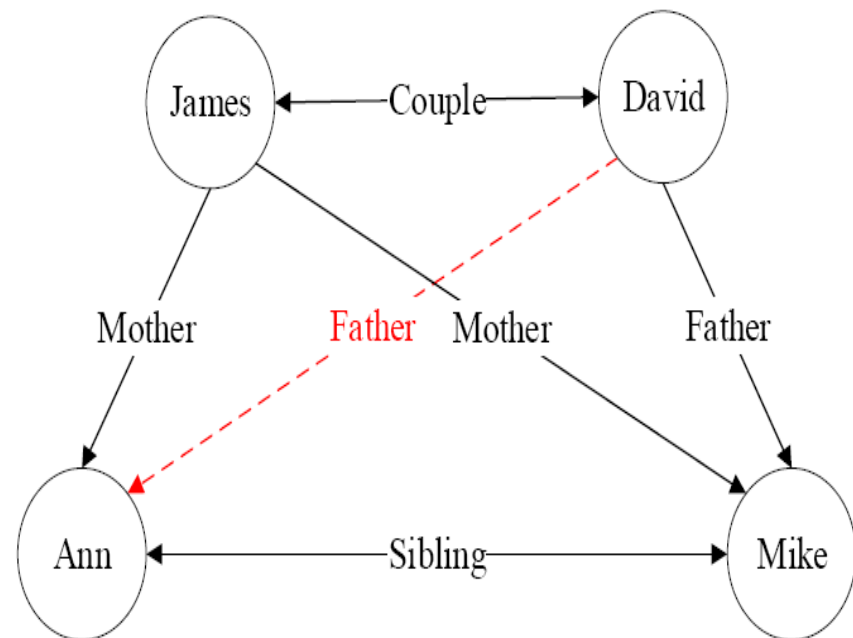
一个简单的家庭关系知识图谱

## 1.2 归纳学习-FOIL

■ 目标谓词:  $Father(x, y)$

■ 背景知识

- 在知识图谱中找到目标谓词以外的其他谓词实例化结果，如右图， $Sibling(Ann, Mike)$



一个简单的家庭关系知识图谱

## 1.2 归纳学习-FOIL

### ■ FOIL (First Order Inductive Learner) 算法

- 推理思路（过程）：从一般到特殊，逐步给目标谓词添加前提约束谓词，直到所构成的推理规则不覆盖任何反例。从一般到特殊指的是：对目标谓词或前提约束谓词中的变量**依据背景知识**赋予具体值，如将 $(\forall x)(\forall y)(\forall z)(Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y))$ 这一推理规则所包含的目标谓词 $Father(x, y)$ 中 $x$ 和 $y$ 分别赋值为David和Ann，进而进行推理。

$$(\forall x)(\forall y)(\forall z)(Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y))$$



前提约束谓词  
(学习得到)



目标谓词  
(已知)

## 1.2 归纳学习-FOIL

■ FOIL (First Order Inductive Learner) 算法

■ 在逐步添加前提约束谓词过程中，如何判断哪些谓词好呢？

利用FOIL信息增益值 (information gain) 选择谓词作为前提约束谓词。

FOIL信息增益值计算方法如下：

$$FOIL\_Gain = \widehat{m}_+ \cdot \left( \log_2 \frac{\widehat{m}_+}{\widehat{m}_+ + \widehat{m}_-} - \log_2 \frac{m_+}{m_+ + m_-} \right)$$

其中， $\widehat{m}_+$ 和 $\widehat{m}_-$ 是增加前提约束谓词后所得新推理规则覆盖的正例和反例的数量， $m_+$ 和 $m_-$ 是原推理规则所覆盖的正例和反例数量。

## 1.2 归纳学习-FOIL

### ■ FOIL (First Order Inductive Learner) 算法

#### ■ 总结回顾

##### 1. 构建正例、反例、背景知识

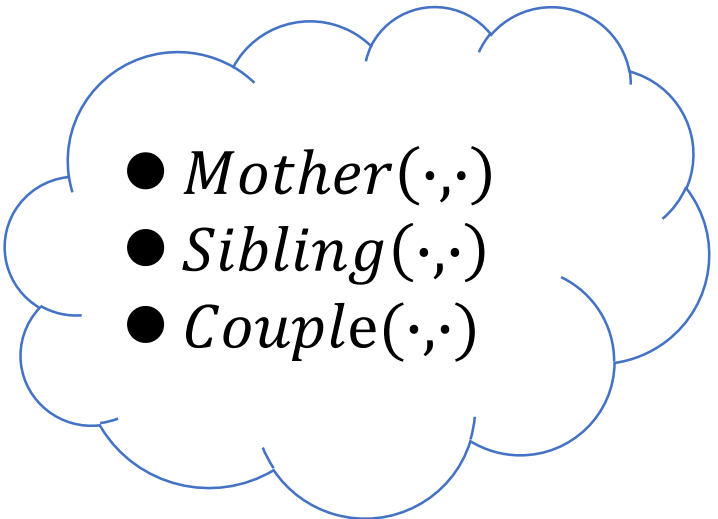
背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例 集合	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)
--------------	---	--------------------	---

## 1.2 归纳学习-F0IL

### ■ F0IL (First Order Inductive Learner) 算法

#### ■ 总结回顾

2. 依次将谓词加入到推理规则中作为前提约束谓词，如：
3. 基于正例、反例、背景知识计算所得新推理规则的F0IL增益值。
4. 基于计算所得F0IL增益值来选择最佳前提约束谓词。
5. 若推理规则覆盖训练样本集合中正例，不覆盖任意反例，则学习结束。



- *Mother(·,·)*
- *Sibling(·,·)*
- *Couple(·,·)*

# 1.2 归纳学习-FOIL

## ■ FOIL 举例

给定目标谓词，此时推理规则只有目标谓词，因此推理规则所覆盖的正例和反例的样本数分别是训练样本中正例和反例的数量，即1和4，因此， $m_+ = 1$ ， $m_- = 4$ 。

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 3$	0.32
	$Sibling(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 2$	0.74
	$Couple(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	<b><math>Couple(x, z)</math></b>	<b><math>\widehat{m}_+ = 1</math></b>	<b><math>\widehat{m}_- = 1</math></b>	1.32
	$Couple(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(z, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA



## 1.2 归纳学习-FOIL

### ■ FOIL 举例

- 将 $Mother(x, y)$ 作为前提约束谓词加入，可得到推理规则  
 $Mother(x, y) \rightarrow Father(x, y)$
- 在背景知识中， $Mother(x, y)$ 有两个实例  
 $Mother(\text{James}, \text{Ann})$   
 $Mother(\text{James}, \text{Mike})$
- 对于 $Mother(\text{James}, \text{Ann})$ 这一实例，  
 $x = \text{James}, y = \text{Ann}$ ，将 $x$ 和 $y$ 代入  
 $Father(x, y)$ 得到 $Father(\text{James}, \text{Ann})$ ，  
 可知在训练样本中 $Father(\text{James}, \text{Ann})$ 是一个反例

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 3$	0.32
	$Sibling(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 2$	0.74
	$Couple(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	<b><math>Couple(x, z)</math></b>	<b><math>\widehat{m}_+ = 1</math></b>	<b><math>\widehat{m}_- = 1</math></b>	1.32
	$Couple(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(z, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA

## 1.2 归纳学习-FOIL

### ■ FOIL 举例

- 将  $Mother(x, y)$  作为前提约束谓词加入，可得到推理规则  
 $Mother(x, y) \rightarrow Father(x, y)$
- 在背景知识中， $Mother(x, y)$  有两个实例  
 $Mother(\text{James}, \text{Ann})$   
 $Mother(\text{James}, \text{Mike})$
- 对于  $Mother(\text{James}, \text{Mike})$  这一实例， $x = \text{James}$ ， $y = \text{Mike}$ ，将  $x$  和  $y$  代入  $Father(x, y)$  得到  $Father(\text{James}, \text{Mike})$ ，可知在训练样本中  $Father(\text{James}, \text{Mike})$  是一个反例

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 3$	0.32
	$Sibling(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 2$	0.74
	$Couple(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	<b><math>Couple(x, z)</math></b>	<b><math>\widehat{m}_+ = 1</math></b>	<b><math>\widehat{m}_- = 1</math></b>	1.32
	$Couple(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(z, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA

# 1.2 归纳学习-FOIL

■ FOIL 举例

$$Mother(x,y) \rightarrow Father(x,y)$$

覆盖正例和反例数量分别为0和2,  
即 $\widehat{m}_+ = 0, \widehat{m}_- = 2$

由于 $\widehat{m}_+ = 0$ , 代入 $FOIL\_Gain$ 公  
式时会出现负无穷的情况, 此时  
 $FOIL\_Gain$  记为NA (Not Available)

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x,y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x,y) \leftarrow$	$Mother(x,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(x,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(y,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(y,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z,y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 3$	0.32
	$Sibling(x,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(x,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(y,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(y,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z,y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 2$	0.74
	$Couple(x,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	<b><math>Couple(x,z)</math></b>	<b><math>\widehat{m}_+ = 1</math></b>	<b><math>\widehat{m}_- = 1</math></b>	1.32
	$Couple(y,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(y,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(z,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(z,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA

## 1.2 归纳学习-FOIL

### ■ FOIL 举例

如果将 $Couple(x, z)$ 作为前提约束谓词加入，可得到如下推理规则  
 $Couple(x, z) \rightarrow Father(x, y)$

在背景知识中， $Couple(x, z)$ 只有一个实例 $Couple(David, James)$ ，即 $x=David$ ， $z=James$ ，将其代入 $Father(x, y)$ 得到 $Father(David, y)$ 。

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 3$	0.32
	$Sibling(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 2$	0.74
	$Couple(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	<b><math>Couple(x, z)</math></b>	<b><math>\widehat{m}_+ = 1</math></b>	<b><math>\widehat{m}_- = 1</math></b>	<b>1.32</b>
	$Couple(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(z, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA

# 1.2 归纳学习-FOIL

## ■ FOIL 举例

在训练样本中存在正例 *Father* (David, Mike)以及反例¬*Father*(David, James)，即 *Couple*(*x*, *z*) → *Father*(*x*, *y*)覆盖正例和反例数量分别为1和1。信息增益值为：

$$\widehat{m}_+ \cdot \left( \log_2 \frac{\widehat{m}_+}{\widehat{m}_+ + \widehat{m}_-} - \log_2 \frac{m_+}{m_+ + m_-} \right)$$

$$= 1 \cdot \left( \log_2 \frac{1}{1 + 1} - \log_2 \frac{1}{1 + 4} \right) = 1.32$$

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
<i>Father</i> ( <i>x</i> , <i>y</i> ) ←	空集	$m_+ = 1$	$m_- = 4$	<i>FOIL_Gain</i>
<i>Father</i> ( <i>x</i> , <i>y</i> ) ←	<i>Mother</i> ( <i>x</i> , <i>y</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	<i>Mother</i> ( <i>x</i> , <i>z</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	<i>Mother</i> ( <i>y</i> , <i>x</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	<i>Mother</i> ( <i>y</i> , <i>z</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	<i>Mother</i> ( <i>z</i> , <i>x</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	<i>Mother</i> ( <i>z</i> , <i>y</i> )	$\widehat{m}_+ = 1$	$\widehat{m}_- = 3$	0.32
	<i>Sibling</i> ( <i>x</i> , <i>y</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	<i>Sibling</i> ( <i>x</i> , <i>z</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	<i>Sibling</i> ( <i>y</i> , <i>x</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	<i>Sibling</i> ( <i>y</i> , <i>z</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	<i>Sibling</i> ( <i>z</i> , <i>x</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	<i>Sibling</i> ( <i>z</i> , <i>y</i> )	$\widehat{m}_+ = 1$	$\widehat{m}_- = 2$	0.74
	<i>Couple</i> ( <i>x</i> , <i>y</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	<b><i>Couple</i>(<i>x</i>, <i>z</i>)</b>	<b><math>\widehat{m}_+ = 1</math></b>	<b><math>\widehat{m}_- = 1</math></b>	<b>1.32</b>
	<i>Couple</i> ( <i>y</i> , <i>x</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	<i>Couple</i> ( <i>y</i> , <i>z</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	<i>Couple</i> ( <i>z</i> , <i>x</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	<i>Couple</i> ( <i>z</i> , <i>y</i> )	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA

## 1.2 归纳学习-FOIL

### ■ FOIL 举例

<b>Back-ground knowledge</b>	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)
<b>Positive and negative samples</b>	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)

- $Couple(x, z)$  加入后信息增益最大
- 将  $Couple(x, z)$  加入推理规则，得到  $Couple(x, z) \rightarrow Father(x, y)$  新推理规则
- 将训练样例中与该推理规则不符的样例去掉。  
这里不符指当  $Couple(x, z)$  中  $x$  取值为 David 时，与  $Father(David, \quad)$  或  $\neg Father(David, \quad)$  无法匹配的实例。
- 训练样本集中只有正例  $Father(David, Mike)$  和负例  $\neg Father(David, James)$  两个实例

## 1.2 归纳学习-FOIL

### ■ FOIL 举例

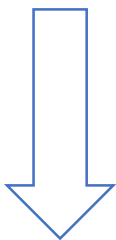
- $Mother(z, y)$  加入信息增益最大
- 将  $Mother(z, y)$  加入，得到新推理规则  
 $Mother(z, y) \wedge Couple(x, z)$   
 $\rightarrow Father(x, y)$
- 当  $x=David$ 、 $y=Mike$ 、 $z=James$  时，该推理规则覆盖训练样本集中正例  $Father(David, Mike)$  且不覆盖任意反例，因此算法学习结束。

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
现有规则	拟加入前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow Couple(x, z)$		$m_+ = 1$	$m_- = 1$	1.32
$Father(x, y) \leftarrow Couple(x, z)$	$\wedge Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Mother(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Mother(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Mother(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Mother(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Mother(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 0$	1
	$\wedge Sibling(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Sibling(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Sibling(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Sibling(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Sibling(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Sibling(z, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Couple(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$\wedge Couple(x, z)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 1$	0
	$\wedge Couple(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Couple(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Couple(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Couple(z, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA

# 1.2 归纳学习-FOIL

## ■ FOIL 举例

*Mother(z,y) ∧ Couple(x,z)*  
*→ Father(x,y)*



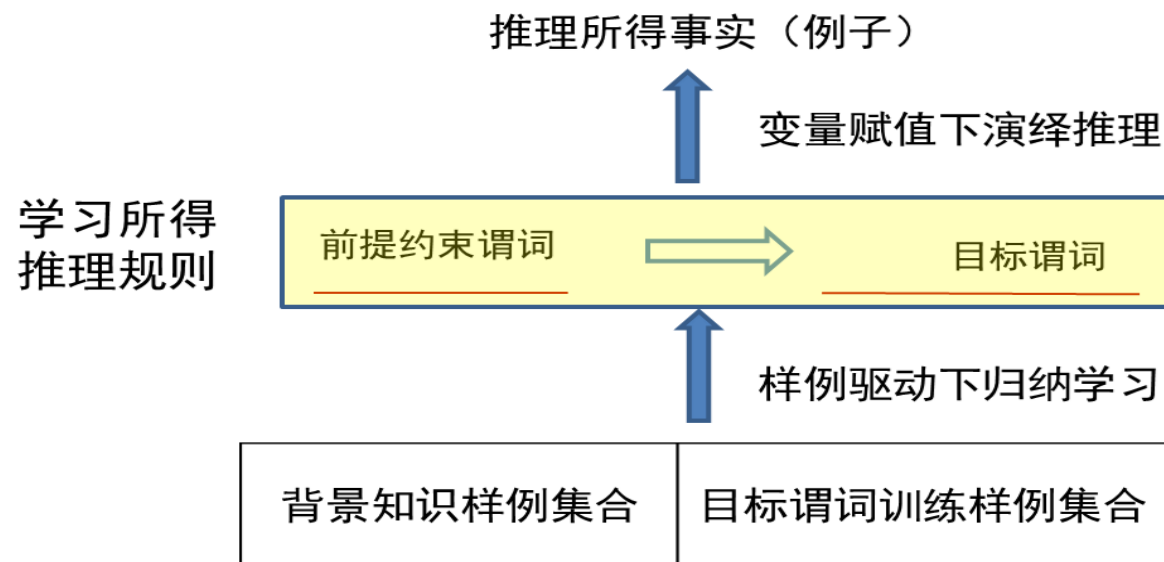
已知:  
*Mother*(James, Ann)  
*Couple*(David,James)

于是: *Father*(David, Ann)

推理规则		推理规则涵盖的 正例和反例数		FOIL信息增益值
现有规则	拟加入前提 约束谓词	正例	反例	信息增益值
<i>Father(x,y) ← Couple(x,z)</i>		$m_+ = 1$	$m_- = 1$	1.32
<i>Father(x,y) ← Couple(x,z)</i>	$\wedge \textit{Mother}(x,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Mother}(x,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Mother}(y,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Mother}(y,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Mother}(z,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textbf{Mother(z,y)}$	$\widehat{m}_+ = \mathbf{1}$	$\widehat{m}_- = \mathbf{0}$	<b>1</b>
	$\wedge \textit{Sibling}(x,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Sibling}(x,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Sibling}(y,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Sibling}(y,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Sibling}(z,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Sibling}(z,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Couple}(x,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$\wedge \textit{Couple}(x,z)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 1$	0
	$\wedge \textit{Couple}(y,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Couple}(y,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Couple}(z,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge \textit{Couple}(z,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA



## 1.2 归纳学习-FOIL



给定目标谓词，FOIL算法从实例（正例、反例、背景知识样例）出发，不断测试所得推理规则是否还包含反例，一旦不包含，则学习结束，由此充分展示了“归纳学习”的能力。在学得推理规则后，利用背景知识再给推理规则中的变量赋予具体例子，经过“演绎”得到新的知识

## 1.3 路径排序推理（了解）

■ 与FOIL算法不同，路径排序推理算法（PRA）的基本思想是将实体之间的关联路径作为特征，来学习目标关系的分类器。路径排序算法的工作流程主要分为三步：

(1) **特征抽取**：生成并选择路径特征集合。生成路径的方式有随机游走（random walk）、广度优先搜索、深度优先搜索等。

(2) **特征计算**：计算每个训练样例的特征值 $P(s \rightarrow t; \pi_j)$ 。该特征值可以表示从实体节点 $s$ 出发，通过关系路径 $\pi_j$ 到达实体节点 $t$ 的概率；也可以表示为布尔值，表示实体 $s$ 到实体 $t$ 之间是否存在路径 $\pi_j$ ；还可以是实体 $s$ 和实体 $t$ 之间路径出现频次、频率等。

(3) **分类器训练**：根据训练样例的特征值，为目标关系训练分类器。当训练好分类器后，即可将该分类器用于推理两个实体之间是否存在目标关系。

## 1.3 路径排序推理（了解）

例 2.16 对图2.3.1中的知识图谱采用路径排序算法。

(1) 目标关系: *Father*

(2) 对于目标关系*Father*，生成四组训练样例，一个为正例、三个为负例：

正例: (David, Mike)

负例: (David, James), (James, Ann), (James, Mike)

(3) 从知识图谱采样得到路径，每一路径链接上述每个训练样例中两个实体：

(David, Mike)对应路径: *Couple* → *Mother*

(David, James)对应路径: *Father* → *Mother*<sup>-1</sup> (*Mother*<sup>-1</sup>与*Mother*为相反关系)

(James, Ann)对应路径: *Mother* → *Sibling*

(James, Mike)对应路径: *Couple* → *Father*

(4) 对于每一个正例/负例，判断上述四条路径可否链接其包含的两个实体，将可链接（记为1）和不可链接（记为

0）作为特征，于是每一个正例/负例得到一个四维特征向量：

(David, Mike): {[1, 0, 0, 0], 1}

(David, James): {[0, 1, 0, 0], -1}

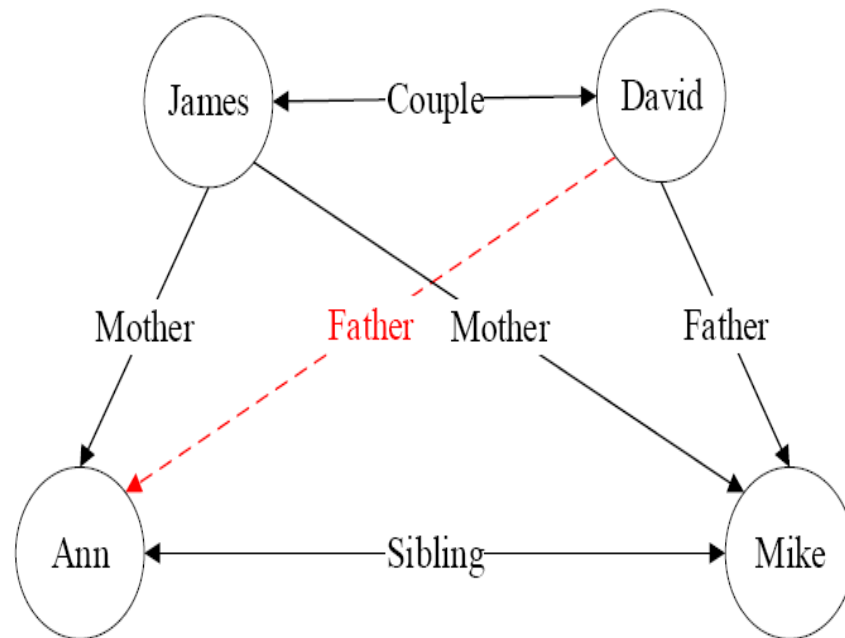
(James, Ann): {[0, 0, 1, 0], -1}

(James, Mike): {[0, 0, 1, 1], -1}

(5) 依据(4)中的训练样本，训练分类器*M*。

(6) 预测。对于图2.3.1中形如(David, Ann)的样例，得到其特征值为[1, 0, 0, 0]，将特征向量输入到分类器*M*中，如

果分类器*M*给出分类结果为1，则*Father*(David, Ann)成立。



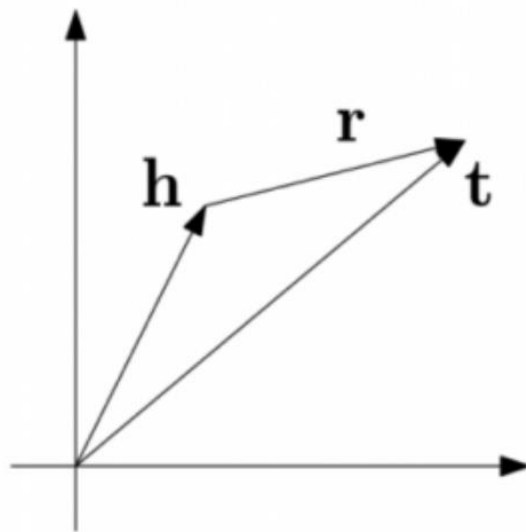
## 1.4 基于分布式表示的知识推理（了解）

- **主要方法：**在深度学习中，将知识图谱中的实体和关系的语义信息用低维向量（分布式表示）表示，即将其映射到稠密的低维向量空间，同时对映射过程进行约束，保留知识图谱中节点之间的拓扑结构从而有效地对知识图谱进行表示和推理。
- **三个步骤：**
  - 表示实体和关系：实体为向量空间中的点，关系为向量空间中的运算
  - 定义评分函数：为 $(h, r, t)$ 定义一个评分函数 $f(h, r, t)$
  - 学习实体和关系表示（模型训练）
- **经典的两类嵌入技术：**
  - 平移距离模型：使用基于距离的评分函数
  - 语义匹配模型：使用基于相似度的评分函数

## 1.4 基于分布式表示的知识推理（了解）

### ■ 平移距离模型的代表：TransE

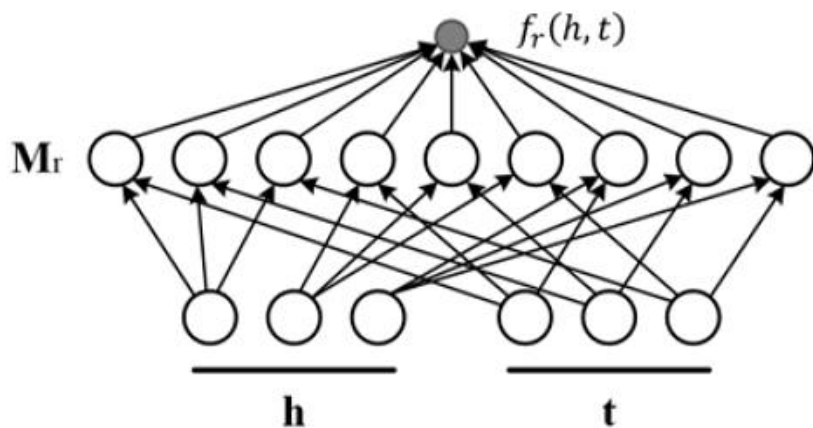
将每个三元组实例 (head, relation, tail) 中的关系relation看做从实体head到实体tail的翻译，通过不断调整h、r和t (head、relation和tail的向量)，使  $(h + r)$  尽可能与  $t$  相等，即  $h + r = t$ 。



## 1.4 基于分布式表示的知识推理（了解）

### ■ 语义匹配模型的代表：RESCAL

计算每个三元组实例 (head, relation, tail) 中三个元素之间的匹配性分数



## 2. 实验任务（选做）

- 编写程序，实现FOIL（First Order Inductive Learner）算法，对如下给定的知识图谱和目标谓词进行规则学习，并得到新的以目标谓词为关系的事实（用一阶逻辑表示）。

- 给定的知识图谱所有三元组的一阶逻辑如下：

Father(Jack,Dell)

Father(Dell,Stephen)

Grandfather(Jack,Stephen)

Father(Dell,Seth)

- 目标谓词

Grandfather(x,y)

## 2. 实验任务（选做）

### ■ 要求的输入输出格式如下

#### ■ 输入格式

- 第1行：知识图谱的三元组数目 $n$
- 第2~( $n + 1$ )行：三元组的一阶逻辑表示，例如  $L(David, Mike)$
- 第 $n + 2$ 行：目标谓词，即  $Grandfather(x, y)$

#### ■ 输出格式

- 第1行：以目标谓词和前提约束谓词组成的规则推理，例如：  $Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y)$
- 第2行及以后：每一行输出一条推理所得事实，用一阶逻辑表示，例如  $H(David, Ann)$

```
4
Father(Jack,Dell)
Father(Dell,Stephen)
Grandfather(Jack,Stephen)
Father(Dell,Seth)
Grandfather(x,y)
Father(x,z)  $\wedge$  Father(z,y)  $\rightarrow$  Grandfather(x,y)
Grandfather(Jack,Seth)
```



Thanks!