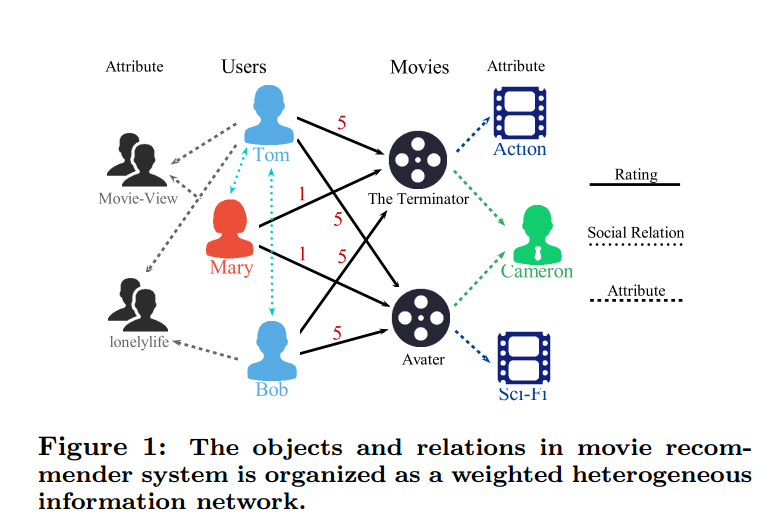
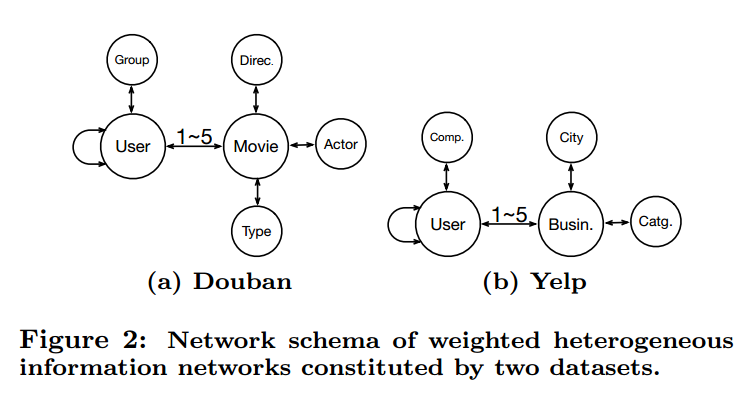
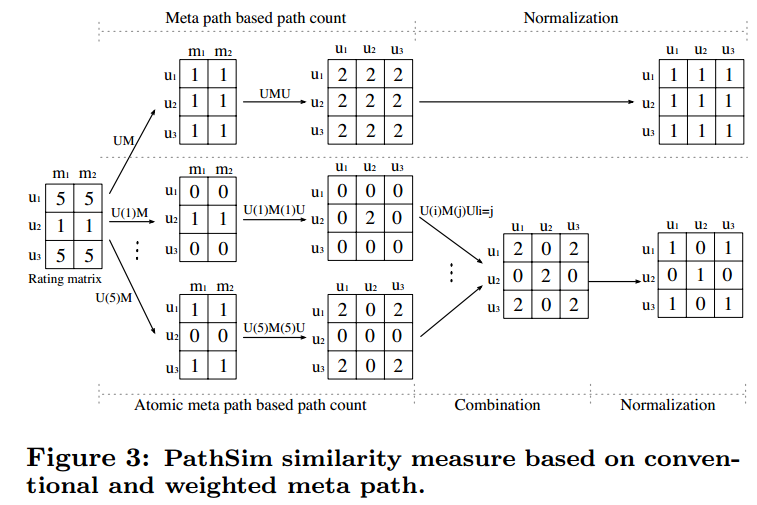
**Semantic Path based Personalized Recommendation on Weighted Heterogeneous Information Networks**







# 3. THE SEMREC SOLUTION

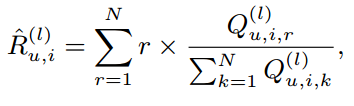
## 3.1 Basic Idea

在本节中，我们提出了基于语义路径的个性化推荐方法（SemRec）来预测项目的分数。具体来说，SemRec首先根据加权或不加权的元路径评估用户的相似度，然后根据类似用户的评分得出预测得分。**在不同的元路径下，用户可以获得不同的推荐结果。如何有效地结合由不同元路径产生的这些建议是具有挑战性的。**我们需要在各种元路径上设置不同的偏好。这导致为每个元路径分配首选权重。当上下文清晰时，我们将偏好权重缩写为加权，而不会混淆加权元路径中的链接权重。学习权重有两方面的困难。（1）优先权重。也就是说，**学习的权重应该体现路径的重要性，反映用户的偏好**。然而，基于不同路径的相似性评估具有明显的偏差，使路径偏好难以反映路径重要性。例如，基于具有密集关系的路径，相似性评估可能都是高的，而基于具有稀疏关系的另一路径，相似性评估可能都是低的。因此，基于不同路径的相似度评价不能反映两个对象的相似度。SemRec设计了一个标准化的评级强度运算来消除相似性偏差，这使得权重更好地反映路径重要性。（2）个性化的权重。也就是说，最好学习**每个用户**的权重偏好。然而，个性化的权重学习可能受到评级稀疏性问题的困扰，因为许多用户几乎没有评级信息。为了缓解个性化权重学习中的评分稀疏问题，**提出了类似用户权重偏好的一致性规则**。也就是说，我们**假设两个相似的用户在元路径上具有一致的权重偏好**。虽然这是合理的，但以前很少使用。两个用户基于路径相似，这意味着路径对这两个用户具有相似的影响。也就是说，这些用户在路径上具有一致的偏好。

遵循这个原则，我们设计了一个新的权重正则化项，有效地减轻了个性化权重学习中的评分稀疏性。

在以下几节中，我们首先设计基于单一路径的基本推荐方法。然后提出了基于多路径的个性化推荐方法的三个层次：所有用户的统一权重，每个用户的个性化权重，以及权重正则化的个性化权重。

## 3.2 单一元路径



算法过程

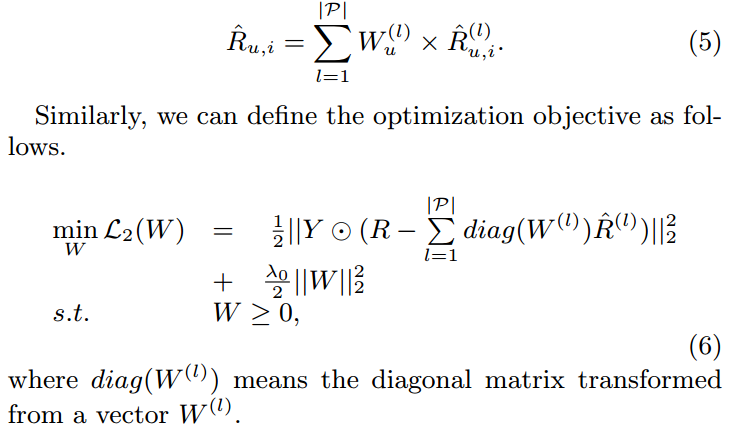
## 3.3 多条路径

存在多条路径

### 3.3.1 Unified weight learning for all users

统一每条路径权重

### 3.3.2 Personalized weight learning for individual user



将向量W变为对角矩阵

### 3.3.3 Personalized weight learning with weight regularization

尽管式（6）考虑了用户的个性化权重，但对于那些评级信息很少的用户来说，可能很难有效地学习权重。训练样本通常比U×I要小得多。训练样本通常不足以进行重量学习，特别是对于那些冷启动用户和项目。根据上述类似用户的权重偏好的一致性规则，用户的路径权重应该与他的类似用户的路径权重一致。对于评分信息较少的用户，由于用户的相似度信息通过元路径更加可用，因此可以从其相似用户的权重中学习路径权重。因此，我们设计一个权重正则化项如下，这迫使**用户的权重与他的类似用户的权重的平均值一致**。

# 4. EXPERIMENTS

## 4.3 Comparison Methods

为了展示所提议的SemRec的有效性，我们比较了SemRec的四种变体。除了具有权重正则化的个性化权重学习方法（称为SemRecReg）之外，还包括三个特殊情况的SemRec：基于单一路径的方法（称为SemRecSgl），统一的所有用户的权重学习方法（称为SemRecAll） 用户（称为SemRecInd）。作为基线，下面举例说明四种代表性的评级预测方法。请注意，前k个推荐方法[15，2]不包括在这里，因为它们解决了不同的问题。

对于这两个数据集，我们采用5条有意义的元路径，其长度不超过4，因为较长的元路径没有意义，并且不能产生良好的相似性度量[12]。表3显示了包含加权和未加权元路径的路径。对于SemRec，我们使用PathSim [12]作为相似性度量来计算用户之间的相似度。SemRec中的参数λ0为0.01，最佳性能为λ1为103。其他方法中的参数在这些数据集上设置的性能最好。

## 4.4 Effectiveness Experiments

对于豆瓣数据集，我们使用不同的训练数据设置（20％，40％，60％，80％）来显示不同数据稀疏度的比较结果。例如，培训数据20％，意味着随机选择用户项目评分矩阵中20％的评分作为预测剩余80％的训练数据。从表2可以看出，豆瓣数据集的评分关系密集，而Yelp评分关系非常稀疏。所以我们在Yelp上使用更多的训练数据（60％，70％，80％，90％）。随机选择独立重复10次，平均结果见表4。请注意，SemRecSgl报告这五条路径上的最佳性能。

从结果中，我们可以看到，在大多数情况下，所有版本的SemRec都优于其他方法。特别是，SemRecReg在所有条件下始终达到最佳性能。例如，在豆瓣的20％的培训中，SemRecReg在RSME上的表现优于19.55％，在MAE上表现为15.89％。与PMF相比，CMF通过将异构信息与矩阵分解相结合来提高推荐性能。然而，在所有条件下，其性能比拟议的SemRec要差得多，特别是在较少的训练集上。作为与SemRec最相似的方法，HeteMF也具有良好的性能，但其性能仍然比拟议的SemRecReg差。这些都意味着提出的SemRec有更好的机制来整合异构信息。

另外，不同版本的SemRec有不同的表现。一般来说，除了SemRecInd之外，具有多个路径的SemRec（例如，SemRecAll和SemRecReg）具有比单个路径（即，SemRecSgl）的SemRec更好的性能，这表明SemRec的权重学习能够有效地整合由不同路径生成的相似性信息。由于评级稀疏性，在大多数情况下，SemRecInd的性能比SemRecAll差。此外，SemRecRec优于SemRecInd的更好的性能证实了重量正则化术语的好处。总而言之，SemRecReg在所有条件下始终达到最佳性能。其原因在于SemRecReg不仅实现了所有用户的个性化权重学习，而且通过权重正则化避免了评级的稀疏性。

此外，我们记录这些方法在学习过程中的平均运行时间。对于两种基于相似性的方法（例如，SemRec和HeteMF），我们不考虑相似性评估的运行时间，因为它可以事先离线完成。对于四个版本的SemRec，当权重学习任务变得更加复杂时，他们的运行时间会增加。SemRecSgl和SemRecAll都非常快，适用于在线学习。与CMF和HeteMF相比，SemRecInd和SemRecReg的运行时间仍然可以接受。我们可以通过在实际应用中平衡SemRec的效率和有效性来选择合适的模型。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 普一 |  |  |  |  |  |  |
|  |  | 莫 | 罗 | 斯 |  |  |  |  |  |
|  |  |  | 德 |  |  |  |  |  |  |
|  |  |  | 摩 |  | 葛 |  |  |  |  |
|  |  | 祖 | 尔 | 法 | 拉 | 克 |  |  |  |
|  |  | 尔 |  |  | 卡 |  |  |  |  |
|  |  | 格 |  |  | 温 | 蕾 | 莎 |  |  |
|  |  | 拉 |  |  | 泉 |  |  |  |  |
|  | 萨 | 布 | 拉 | 金 |  |  |  |  |  |
|  |  |  |  | 加 |  |  | 崔 |  |  |
|  |  |  |  | 洛 |  |  | 利 |  |  |
|  |  |  |  | 斯 |  |  | 艾 |  |  |
|  |  | 哈 |  |  |  |  | 克 |  |  |
|  |  | 里 |  |  |  |  | 斯 |  |  |
|  |  | 森 |  |  |  |  |  |  |  |
|  |  | 福 |  |  |  |  |  |  |  |
|  |  | 特 |  |  |  |  |  |  |  |