基于集体矩阵分解的关系学习

摘要

关系学习关注的是给定一个数据库的实体集和被观测实体集之间的关系，预测一个关系的未知值。关系学习的一个例子是电影评分预测，其中实体包括用户、电影、类型和演员。关系隐含于用户对电影的评分、电影的类型和电影中演员的角色。给定成对的关系数据，例如 #用户 x #电影 评分矩阵，一个常用的预测技术是低秩矩阵分解（low-rank matrix factorization）。在用多个矩阵表示的多重关系域中，我们可以利用一个关系中的信息，同时预测另一个关系，以提高预测精度。为此，我们提出集体矩阵分解模型：同时分解若干个矩阵，当一个实体涉及多个关系时，因子是共享的。每个关系可以有不同的数值类型和误差分布。所以，我们允许参数和输出是非线性关系，其中使用布雷格曼散度(Bregman divergences)测量误差。我们使用标准交替投影算法来扩展我们的模型。此外，我们提出随机优化方法来处理大规模稀疏矩阵。我们的模型归纳了若干已有的矩阵分解方法，因此产生了新的大规模优化算法来处理这个问题。我们的模型可以处理成对的关系模式和各种各样的误差模型。我们证实了它的效率和在关系间共享参数的好处。

1. 引言

关系型数据包含实体和实体间的关系。在许多案例中，例如关系型数据库实体类型和关系类型的数量是固定的。在这个领域中，两个重要的任务是链路预测（link prediction），判定两个实体间是否存在关系；链路回归（link regression），若存在关系判定关系的数值。许多相关的领域只有一个或者两个实体类型：文档与单词；用户与物品；学术论文实体间的连接表示为计数、评分和引用。在这些领域中，我们可以表示这些连接为 m x n 矩阵 X：X 的行对应着实体的一个类型，X 的列对应着其它类型，X_{ij} 的元素表示实体 i 和 j 是否存在关系。X 的低秩矩阵分解的形式为 $X \approx f(UV^T)$，其中 $U \in \Re^{m \times k}$，$V \in \Re^{n \times k}$。这里 k>0 是秩，f 是一个非线性映射函数。选择不同的 f、不同的≈定义形成不同的模型：使用最小化平法误差和恒等连接产生奇异值分解（对应着高斯误差模型），而其它的选择如广义线性模型[26,14,17]，误差模型如泊松分布、伽玛分布和伯努利分布。

在多于一个关系矩阵的领域里，一种方法是分别拟合每个关系。但是这种方法没有利用到关系和关系之间的相关性。举个例子，一个用户、电影、类型的领域可能有两个关系：一个表示用户评分的整型矩阵，数字范围为 1-5；一个二值矩阵，代表每部电影属于的类型。我们想要发掘他们间的关系来提升预测效果。

为此，我们将广义线性模型扩展到任意关系域。我们将每个关系矩阵与一个泛型线性映射函数关联起来，当一个实体类型涉及多个关系时，我们将不同模型的因子联系在一起。我们把这种方法称为集体矩阵分解（collective matrix factorization）

我们证明了集体矩阵分解的一般方法可以高效地工作在大且稀疏的数据集上，通过使用关系模式（relational schema）和非线性映射函数。此外，我们还展示了，当关系相互关联时，比起分别分解每个矩阵，集体矩阵分解能够实现更高的预测精度。

2. 矩阵分解的统一视角

集体矩阵分解的基本构件是单一矩阵分解，它对两个实体类型之间的单个关系建模。如果有 m 个类型为 ε_1 的实体和 n 个类型为 ε_2 的实体，我们定义 $X \in \Re^{m \times n}$ 为观测到的矩阵，$U \in \Re^{m \times k}$，$V \in \Re^{n \times k}$ 为它的低秩矩阵分解因子。一个因子分解算法可以通过以下的若干个选择来定义，这足以包含大多数现有的方法

1. 选择映射函数 f: $\Re^{m \times n}$ -> $\Re^{m \times n}$
2. 损失函数 $D(X, f(UV^T)) \geq 0$，它衡量预测值 $f(UV^T)$和答案 X 的误差
3. 可选的数据权重 $W \in \Re_+^{m \times n}$ ， 如果使用这一项，那它就必须是损失的一个参数
4. 因子的硬约束，$(U, V) \in C$
5. 正则化惩罚项，$R(U, V) \geq 0$

对于模型 $X \approx f(UV^T)$，我们目标是

$$\underset{(U,V)\in C}{\operatorname{argmin}}[D(X, f(UV^T)) + R(U,V)] \qquad (1)$$

损失 D(.,.)量化了模型≈的效果。第二个参数通常是凸的，并且分解为一个 X 元素的加权和。例如，weighted SVD[32]的损失为

$$D_W(X, UV^T) = \| W \odot (X - UV^T) \|_{Fro}^2$$

其中⊙表示矩阵间的元素点积。

预测映射函数 f 允许数据 X 与 UV^T 是非线性关系。f 和 D 的选择与 X 的分布假设密切相关。参见 2.1 节。常见的线性模型正则项，如 p-norms，很适合用在矩阵分解中；其它特别为了因子分解而提出来的正则项，例如，UV^T 的迹，奇异值之和，被认为是秩的连续代理[33]。为了清晰起见，我们将硬约束 C 从正则化分离开。硬约束的例子包括正交性；行、列和块的随机性（例如，在矩阵中，每一行 U 和 V 的总和为 1）；非负性；稀疏和基数。

2.1 布雷格曼散度

一大类矩阵分解算法给布雷格曼散度做了限制 D：例如，奇异值分解[16]和非负矩阵分解[21]。

定义 1 （[17]） 对于一个 closed,proper 凸函数 F: $\Re^{m \times n}$ -> $\Re$，广义的布雷格曼散度为

$$D_F(Z \| Y) = F(Z) + F^*(Y) - YoZ$$

其中 AoB 是矩阵点积 $tr(A^TB) = \Sigma_{ij} A_{ji} B_{ij}$。$F^*$ 是凸对偶 $F^*(\mu) = \sup_{\theta \in domF}[<\theta, \mu> - F(\theta)]$

如果 F^* 是可微的，那么这与标准定义等价[10,11]，除了标准定义使用参数 Z 和$\nabla F^*(Y)$而不是 Z 和 Y。如果 F 分解为 Z 成分的和，那么我们可以定义一个加权的散度，重定义 F 为和的单个成分

$$D_F(Z \| Y, W) = \sum_{ij} W_{ij} (F(Z_{ij}) + F^*(Y_{ij}) - Y_{ij} Z_{ij})$$

包含加权版本的平方损失例子，F(x) = x^2 ，I-divergence，F(x) = xlogx-x。我们首要焦点是在可分解的常规布雷格曼散度[6]，它与指数族的最大似然一致。

定义 2 一个参数化的分布族 $\psi_F=\{p_F(x|\theta):\theta\}$，如果密度函数有以下形式则是常规指数分布族

$$\log p_F(x|\theta)=\log p_0(x)+\theta^T x-F(\theta)$$

其中 θ 是分布的自然参数向量，x 是最小化的显著统计向量，$F(\theta)$是 log-partition 函数

$$\mathrm{F}(\theta)=\log\int p_0(x)\cdot\exp(\theta^T x)dx$$

在 ψ_F 中的分布可以通过它的自然参数唯一识别。对于常规的指数族

$$\log p_F(x|\theta)=\log p_0(x)+F^*(\theta)-\mathrm{D}_{F^*}(x\|f(\theta))$$

其中匹配预测映射函数是 $f(\theta)=\nabla F(\theta)$[15,4,14,6] 。在使用匹配预测映射函数时，最小化布雷格曼散度等价于最大似然对应的指数族分布。

矩阵分解因子和指数族之间的关系被当作数据矩阵 X 样例的集合，$X=\{X_{11},...,X_{mm}\}$. 建立 $X=f(UV^T)$模型，我们有由带有自然参数$(UV^T)_{ij}$的分布 ψ_F 描绘的 X_{ij}。

可分解的损失，它可以被表达为损失元素的和，其遵从矩阵的可交换性[2,3]。如果对 X 的行和列进行排列不改变 X 的分布，则矩阵 X 是行和列交换的。例如，如果 X 是一个文档-单词计数矩阵，那么两个文档在矩阵中的相对位置是不重要的，行是可交换的，单词也如此。矩阵可交换性的一个令人惊讶的结果是，X 的分布可以由一个全局矩阵平均值、行和列效应（例如，行偏置项，列偏置项）、per-element 效应（例如上文的 UV^T 的自然参数）的函数来描述。per-element 效应自然地带来可分解的损失。当一个维度索引着随时间变换的量时，可分解性不是一个合理的假设。

2.2 例子

矩阵分解最简单的例子是奇异值分解: 数据权重是常数，预测映射是恒等函数，散度是平方误差的和，因子是不正则化的。硬约束是因子是正交的，而标准正交保证了全局最优的唯一性（相当于排列和符号变换），可以用高斯消元法或幂方法来发现。[16]

不同的矩阵分解会改变上述的一个或多个选项。非负矩阵分解[21]将下述目标最大化

$$\mathrm{X}o\log(UV^T)-1oUV^T \qquad (2)$$

其中 **1** 是元素全为 1 的矩阵。最小化等式 2 等价于最小化约束为 $U,V\geqslant 0$ 的 I-divergence $D_H(X||\log(UV^T))$。这里 $H(x)=x\log(x)-x$。预测映射函数 $f(\theta)=\log(\theta)$。

我们考虑的矩阵分解的范围比[17]更广，但是相同的交替牛顿-投影方法(见第 4 - 5 节)可以推广到所有的以下场景，以及集体矩阵分解（i）在布雷格曼矩阵分解中通常不考虑因子的约束，因为由此造成损失不再是常规的布雷格曼散度。约束允许我们将非负矩阵分解(non-negative matrix factorization)[21]或矩阵协聚类(matrix co-clustering)的方法引入到我们的框架中。（ii）非布雷格曼矩阵分解，如 max - margin 矩阵分解[30]，可以利用大规模优化技术，见 4 - 5 节。（iii）行和列的偏差，其中 U 的列与固定不变的 V 列成对(反之亦然)。如果预测映射和损失对应着伯努利分布，那么边际损失就是偏置的特例（iv）基于 plate 模型的方法，例如 pLSI [19]，可以放到我们的框架中，也可以放到分解数据矩阵的方法中。虽然这些功能可以添加至集体矩阵分解中，但是这方面我们主要关注的是相关性问题。

3. 关系模式

一个关系模式包含 t 个实体类型，$\varepsilon_1 \ldots \varepsilon_t$。类型 i 有 n_i 个实体，$\{x_e^{(i)}\}_{e=1}^{n_i}$。两个类型之间的关系是 $\varepsilon_i \sim_u \varepsilon_j$；下标 u∈N，可以让我们区分两个相同关系之间的多重关系，而且但没有歧义时，可以省略。在这篇论文，我们只考虑二元关系。$\varepsilon_i \sim_u \varepsilon_j$ 矩阵有 n_i 行，n_j 列，符号为 $X^{(i,j,u)}$。如果没有观测到关系，则会在矩阵相应位置填充 0（所以 $X^{(i,j,u)}$ 是稀疏矩阵），当学习参数时，给予他们零权重。按照惯例，我们假定 i≤j。不失一般性，我们假定任意实体类型相关联；如果不是，我们则可以单独对关系模式中的连接成分进行拟合。这与全连接的实体关系模型（fully connected entity-relationship model）[12]一致。

我们通过隐因子的乘积来拟合每个关系矩阵，$X^{(i,j)} \approx f^{(i,j)}(U^{(i)}(U^{(j)})^T)$。其中 $U^{(i)} \in \Re^{n_i \times k_{ij}}$，$U^{(j)} \in \Re^{n_j \times k_{ij}}$，$k_{ij} \in \{1,2,\ldots\}$。除非另外的标记，否则预测映射函数 $f^{(i,j)}$ 是矩阵的 element-wise 函数。如果 ε_j 参与超过一个关系，我们允许模型使用 $U^{(j)}$ 的一部分列来计算。这种灵活性允许我们在不同的隐特征维度有关系，或者若在 ε_i 和 ε_j 之间有多个关系，不用预测每个关系的相同值。在实现中，我们为每个关系，参与其计算的每个因子的列索引存储了一个列表，但是为了避免混乱，我们在符号标记上，忽略了这方面。

4. 集体因子分解

为了简洁，我们在三个实体类型模式 $\varepsilon_1 \sim \varepsilon_2 \sim \varepsilon_3$ 中 引入集体矩阵分解，并且使用简化的标记：两个数据矩阵 $X = X^{(12)}$、$Y = X^{(23)}$，维度 $m = n_1$、$n = n_2$、$r = n_3$。因子 $U = U^{(1)}$、$V = U^{(2)}$、$Z = U^{(3)}$。隐特征维度 $k = k_{12} = k_{23}$。X 的权重为 W，Y 的权重 W^{hat}。其中 ε_2 参与了两个关系，我们使用因子 V 在相同的公式中：$X \approx f_1(UV^T)$ $Y \approx f_2(VZ^T)$。

这种模式的一个例子是协同过滤：ε_1 是用户，ε_2 是电影，ε_3 是类型。X 是观测到的评分矩阵，Y 指示电影属于哪种类型（每一列对应一种类型，一部电影可以属于多种类型）。

布雷格曼矩阵分解的一个模型[17]处理下列可分解的损失函数，以达目标 $X \approx f_1(UV^T)$：

$$L_1(U,V \mid W) = D_{F_1}(UV^T \parallel X, W) + D_G(0 \parallel U) + D_H(0 \parallel V)$$

其中，$G(u) = \lambda u^2/2$，$H(v) = \gamma v^2/2$，$\lambda, \gamma > 0$，对应者 l_2 正则。忽略因子种不改变的项，损失为

$$L_1(U,V \mid W) = W \odot (F(UV^T) - XoUV^T) + G^*(U) + H^*(V)$$

同时，如果 Y 是被单独分解的，那么损失为

$$L_2(V,Z \mid \tilde{W}) = D_{F_2}(VZ^T \parallel Y, \tilde{W}) + D_H(0 \parallel V) + D_I(0 \parallel Z)$$

因为 V 是共享的因子，所以我们合并损失

$$L(U,V,Z \mid W, \tilde{W}) = \alpha L_1(U,V \mid W) + (1-\alpha) L_2(V,Z \mid \tilde{W}) \qquad (3)$$

其中 $\alpha \in [0,1]$ 衡量关系的相对重要性。

损失的每一项，L_1和L_2都是可分解并且是二次可导的，这是在 4.1 节描述的交替投影技术所需要的条件。尽管等式 3 很简单，它有一些有趣的影响。给定 $x^{(1)}_i$ 和 $x^{(1)}_j$的 X_{ij}的分布，和给定 $x^{(2)}_j$ 和 $x^{(3)}_k$的 Y_{jk}的分布 ，不需要对于 $x^{(2)}_j$的边际分布相一致。扩展行列可交换的概念，每个实体 $x^{(2)}_j$ 对应着一条记录，它可能与类型 ε_1和 ε_3 的实体有关系。令 $F_{2,1}$表示涉及到实体 ε_1的对应关系的特征，$F_{2,3}$表示涉及到实体 ε_3的对应关系的特征。如果特征是二值的，那么它们指示 $x^{(2)}_j$ 中是否一个实体参与了一个关系。$x^{(2)}_j$ 潜在表示是 $V_{j\cdot}$ ，其中 $U|V^T_{j\cdot}$ 和 $V_{j\cdot}Z^T$ 分别确定了 $F_{2,1}$和 $F_{2,3}$的分布。

4.1 参数估计

等式 3 是凸的。我们扩展交替投影算法到矩阵分解，固定除了 $\mathrm{L}(\mathrm{U},\mathrm{V},\mathrm{Z}\,|\,\mathrm{W},\tilde{\mathrm{W}})$ 以外其它参数，使用 Newton-Raphson step 更新因子。

对于每个因子对损失求微分：

$$\nabla_{\mathrm{U}}L=\alpha(W\odot(\mathrm{f}_1(UV^T)-X))V+\nabla G^*(U) \qquad (4)$$

$$\nabla_{\mathrm{V}}L=\alpha(W\odot(\mathrm{f}_1(UV^T)-X))^{\mathrm{T}}U+(1-\alpha)(\tilde{W}\odot(\mathrm{f}_2(\mathrm{VZ}^{\mathrm{T}})-\mathrm{Y}))\mathrm{Z}+\nabla H^*(V) \qquad (5)$$

$$\nabla_{\mathrm{Z}}L=(1-\alpha)(\tilde{W}\odot(\mathrm{f}_2(VZ^T)-Y))^TV+\nabla I^*(Z) \qquad (6)$$

将梯度设置为零，更新 U，V 和 Z。梯度步骤不要求散度可分解，也不要求损失是可微分的；将前文的梯度简单替换为次梯度。对于 U 的 l_2正则为 $G(U)=\lambda||U||^2/2$，$\nabla G^*(U)=U/\lambda$。因子的梯度是梯度与因子参与组成的矩阵的线性组合。

粗略地检查一下方程 4－6 就可以看出牛顿的步骤是不可行的。关于 U 的 Hessian 会涉及到 nk 个参数。然而，如果 L_1和 L_2都是可分解的函数，那么我们就可以证明，对于单个因子 U，几乎所有 L 的二阶导数都为零。此外，牛顿法对因子的更新也减少了对 U、V 和 Z 的 row-wise 优化。对于模型的子集，方程 4 － 6 是可微分的，损失是可分解的，定义

$$\mathrm{q}(U_{i\cdot})=\alpha(W_{i\cdot}\odot(\mathrm{f}_1(\mathrm{U}_{\mathrm{i}\cdot}\mathrm{V}^{\mathrm{T}})-\mathrm{X}_{\mathrm{i}\cdot}))\mathrm{V}+\nabla\mathrm{G}^*(U_{i\cdot})$$

$$\mathrm{q}(\mathrm{V}_{i\cdot})=\alpha(W_{\cdot i}\odot(\mathrm{f}_1(\mathrm{UV}^{\mathrm{T}}_{\mathrm{i}\cdot})-\mathrm{X}_{\cdot\mathrm{i}})^{\mathrm{T}}\mathrm{U}+(1-\alpha)(\tilde{W}_{i\cdot}\odot(\mathrm{f}_2(\mathrm{V}_{\mathrm{i}\cdot}\mathrm{Z}^{\mathrm{T}})-\mathrm{Y}_{\mathrm{i}\cdot}))Z+\nabla\mathrm{H}^*(V_{i\cdot})$$

$$\mathrm{q}(\mathrm{Z}_{i\cdot})=(1-\alpha)(\tilde{W}_{\cdot i}\odot(\mathrm{f}_2(\mathrm{VZ}^{\mathrm{T}}_{\mathrm{i}\cdot})-\mathrm{Y}_{\cdot\mathrm{i}}))^{\mathrm{T}}V+\nabla\mathrm{I}^*(Z_{i\cdot})$$

因为除了一个因子其它都固定，这里先考虑 $q(U_i)$关于任意 $U:\nabla_{U_{js}}q(U_i)$ 的导数。因为当 j=i 时，U_{js}只在 $q(U_i)$中出现，所以当 j≠i 时导数为零。因此 Hessian 矩阵 $\nabla^2_u L$ 是区块对角化的，其中非零块对应着 U 的一行。区块对角化矩阵的逆是每个块的逆，所以对于 U 的牛顿法导数方向，$[\nabla_u L][\nabla^2_u L]^{-1}$，可以被简化为使用 $[q(U_i)][q'(U_i)]^{-1}$来更新 U_i的每一行。对于 V 和 Z 也是一样，因为损失是每个矩阵损失之和，且导数是一个线性算子。

对于损失 L 的任何（局部）优化对应着等式 $\{\mathrm{q}(\mathrm{U}_{\mathrm{i}\cdot})\}_{\mathrm{i}=1}^{\mathrm{m}},\{\mathrm{q}(\mathrm{V}_{\mathrm{i}\cdot})\}_{\mathrm{i}=1}^{\mathrm{n}},\{\mathrm{q}(\mathrm{Z}_{\mathrm{i}\cdot})\}_{\mathrm{i}=1}^{\mathrm{r}}$ 的根。U_i

的牛更新步骤

$$\mathrm{U}_{i.}^{new} = U_{i.} - \eta \cdot q(U_{i.})[q^{'}(U_{i.})]^{-1} \qquad (7)$$

其中我们建议使用 Armijo criterion[28]来设置 η。
为了简介描述 Hessian 矩阵，我们为正则引入一些符号

$$\mathrm{G}_i \equiv diag(\nabla^2 G^*(U_{i.}))$$

$$\mathrm{H}_i \equiv diag(\nabla^2 H^*(V_{i.}))$$

$$\mathrm{I}_i \equiv diag(\nabla^2 I^*(Z_{i.}))$$

为重构误差项引入符号

$$\mathrm{D}_{1,i} \equiv diag(W_{i.} \odot \mathrm{f}_1^{'}(\mathrm{U_{i.}V^T}))\text{，}\ \mathrm{D}_{2,i} \equiv diag(W_{.i} \odot \mathrm{f}_1^{'}(\mathrm{UV_{i.}^T}))$$

$$\mathrm{D}_{3,i} \equiv diag(\tilde{W}_{i.} \odot \mathrm{f}_2^{'}(\mathrm{V_{i.}^T Z}))\text{，}\ \mathrm{D}_{4,i} \equiv diag(\tilde{W}_{.i.} \odot \mathrm{f}_2^{'}(\mathrm{VZ_{i.}^T}))$$

对于损失 L 的 Hessian 矩阵为

$$\mathrm{q}^{'}(U_{i.}) \equiv \nabla q(U_{i.}) = \alpha V^T D_{1,i} V + G_i$$

$$\mathrm{q}^{'}(\mathrm{Z}_{i.}) \equiv \nabla q(Z_{i.}) = (1-\alpha) V^T D_{4,i} V + I_i$$

$$\mathrm{q}^{'}(\mathrm{V}_{i.}) \equiv \nabla q(V_{i.}) = \alpha U^T D_{2,i} U + (1-\alpha) Z^T D_{3,i} Z + H_i$$

每次更新 U,V 和 Z 至少减少等式 3 的一项。循环迭代更新直到局部最优解。在实践中，我们简化了更新方式，只进行一次牛顿法而不是一直运行到收敛。

4.2 权重

除了权衡重建矩阵的不同部分的重要性外，W 和 $\mathrm{W^{hat}}$ 还有其他用途。首先，通过使用 $(\mathrm{nm})^{-1}$ 对 X 每个元素，使用 $(\mathrm{nr})^{-1}$ 对 Y 每个元素进行缩放，可以使用数据权重来将目标转化为每个元素的损失。这确保了更大的矩阵不会仅仅因为它们规模大而支配模型。其次，权重可以用来校正 $L_1(U,V)$ 和 $L_2(V,Z)$ 的规模差异。如果是普通的布雷格曼散度，我们可以使用相应的对数似然当作一致的规模。如果不是，则计算

$$\mathrm{D}_{\mathrm{F_1}}(UV^T \parallel X, W) / D_{F_2}(VZ^T \parallel Y, \tilde{W})$$

平均均匀随机参数 U,V 和 Z，提供了对两个损失相对规模的适当估计。第三种数据权重的用途是在缺失值上。如果一个关系的值是未知的，那么它对应的权重为零。

4.3 推广到任意模式

将三因子模型推广到任意成对关系模式，其中二元关系被表示为边的集合：E={(i,j): ε_i、ε_j∧i<j}。令[U]表示隐因子集合，[W]表示权重矩阵。模型的损失为

$$\mathrm{L([U]|[W])} = \sum_{(\mathrm{i,j}) \in \mathrm{E}} \alpha^{(\mathrm{i,j})} (D_{F^{(i,j)}}(U^{(i)}(U^j)^T \parallel X^{(ij)}, W^{(ij)})) + \sum_{i=1}^{\mathrm{t}} (\sum_{j:(i,j) \in E} \alpha^{(ij)}) D_{G^{(i)}}(0 \parallel U^{(i)})$$

其中 $F^{(ij)}$定义一个特定重构的损失，$G^{(i)}$定义正则化矩阵的损失。相对权重 $\alpha^{(ij)} \geqslant 0$ 衡量每个矩阵在重构中的重要性。由于损失是单个损失的线性函数，而微分算子是线性的，梯度和牛顿更新都可以用类似于 4.1 节的方式导出。当 $U^{(i)}$作为列因子而不是行因子时，注意区分。

5. 随机近似

在优化集体分解模型的过程中，我们正处在一个不寻常的情况下，我们主要关心的不是计算 Hessian 的成本，而是计算梯度本身的成本：如果 k 是最大的嵌入维度数，那么对于一行 $U_r^{(i)}$ 的一个梯度更新步骤的复杂度是 $O(k\sum_{j:\varepsilon_i \sim \varepsilon_j} n_j)$，而牛顿法更新的复杂度是 $O(k^3 + k^2\sum_{j:\varepsilon_i \sim \varepsilon_j} n_j)$。通常 k 比实体的数量小得多，所以牛顿法更新复杂度只是 k 的一个系数。（以上的结论是以稠密矩阵为前提，对于稀疏的关系，我们可以用与实体 $x_r^{(i)}$有关的实体类型 ε_j 的数量代替 n_j，但是结论仍是相同的）

$U_r^{(i)}$的梯度计算代价最大的部分，是为实体 $x_r^{(i)}$参与的每个被观测到的关系计算预测值，所以我们对所有加权预测误差求和。一种降低复杂度的方法是只在关系子集中计算误差。这个技术被称为随机近似（stochastic approximation）[7]。最有名的随机近似算法是随机梯度下降；但是由于对 Hessian 矩阵求逆并不是我们复杂度的重要部分，我们推荐用随机牛顿法代替。

这里考虑三因子模型中 U_i 的更新。这个更新可以看作是回归，其中数据是 X_i，特征是 V 的列。数据的一个采样 $s \subseteq \{1,\ldots,n\}$，第 τ 次迭代的采样梯度为

$$\hat{\mathrm{q}}_\tau(U_{i.}) = \alpha(W_{is} \odot (\mathrm{f}(\mathrm{U_{i.}V_{s.}^T}) - \mathrm{X_{is}}))V_{s.} + \nabla G^*(U_{i.})$$

同时，给定子集 $p \subseteq \{1,\ldots,n\}$、$q \subseteq \{1,\ldots,r\}$，对其它因子的采样梯度为

$$\hat{\mathrm{q}}_\tau(\mathrm{V}_{i.}) = \alpha(W_{pi} \odot (\mathrm{f}(\mathrm{U_{p.}V_{v.}^T}) - \mathrm{X_{pi}}))^T U_{p.} + (1-\alpha)(\tilde{W}_{iq} \odot (\mathrm{f}(\mathrm{V_{i.}Z_{q.}^T}) - \mathrm{Y_{iq}}))Z_{q.} + \nabla H^*(V_{i.})$$

$$\hat{\mathrm{q}}_\tau(\mathrm{Z}_{i.}) = (1-\alpha)(\tilde{W}_{si} \odot (\mathrm{f}(\mathrm{V_{s.}Z_{i.}^T}) - \mathrm{Y_{si}}))^T V_{s.} + \nabla I^*(\mathrm{Z}_{i.})$$

随机梯度下降法，第 τ 次迭代更新为

$$\mathrm{U}_{i.}^{\tau+1} = \mathrm{U}_{i.}^{\tau} - \tau^{-1}\hat{q}_\tau(U_{i.})$$

并且同时更新其它几个因子。注意，我们使用的是一个固定的、递减的学习率，而不是线性搜索：梯度的样本估计并不总是朝着下降的方向。比起线性搜索，固定的方式的另一个优势是，前者的计算成本很高。

我们用非均匀分布、不放回、由数据权重的分布来采样数据。换言之，对于 $U_{i.}$ 的一行，X_{ij} 的概率为 $W_{ij}/\Sigma_j W_{ij}$。这种抽样分布提供了令人信服的相关解释：为了更新隐因子 $x_r^{(i)}$，我们只采样涉及到 $x_r^{(i)}$ 的被观测到的关系。例如，为了更新用户的隐因子，我们只对用户评价的电影进行采样。我们为每一行 U 使用一个单独的示例:这样，误差就在行间独立，它们之间的影响会被取消。实际上，这意味着实际上每次迭代训练损失都会降低。

随着抽样，梯度更新的复杂度不再与 $x_r^{(i)}$ 相关的实体数量呈线性关系，只与被采样的实体数量相关。这个方法的其它优点是，当我们同一时间采样一个实体时，$|s|=|p|=|q|=1$，随机梯度法变成在线算法，这不再需要将数据预存在内存中。

如上所述，我们可以通过将随机梯度下降法改成随机 Newton-Raphson 更新[7,8]，来提高收敛的程度。对于三因子模型，stochastic Hessian 矩阵为

$$\hat{q}_{\tau}^{'}(U_{i.}) = \alpha V_{s.}^{T} \hat{D}_{1,i} V_{s.} + G_i$$

$$\hat{q}_{\tau}^{'}(Z_{i.}) = (1-\alpha) V_{s.}^{T} \hat{D}_{4,i} V_{s.} + I_i$$

$$\hat{q}_{\tau}^{'}(V_{i.}) = \alpha U_{p.}^{T} \hat{D}_{2,i} U_{p.} + (1-\alpha) Z_{q.}^{T} \hat{D}_{3,i} Z_{q.} + H_{i.}$$

其中

$\hat{D}_{1,i} \equiv diag(W_{is} \odot f_1^{'}(U_{i.} V_{s.}^{T}))$，$\hat{D}_{2,i} \equiv diag(W_{pi} \odot f_1^{'}(U_{p.} V_{i.}^{T}))$

$\hat{D}_{3,i} \equiv diag(\tilde{W}_{iq} \odot f_2^{'}(V_{i.}^{T} Z_{q.}))$，$\hat{D}_{4,i} \equiv diag(\tilde{W}_{si} \odot f_2^{'}(V_{s.} Z_{s.}^{T}))$

为了满足收敛条件，具体见 5.1 节，我们对 Hessian 使用指数加权移动平均：

$$\bar{q}_{\tau+1}(.) = (1 - \frac{2}{\tau+1}) \bar{q}_{\tau}(.) + \frac{2}{\tau+1} \hat{q}_{\tau+1}^{'}(.) \qquad (8)$$

当每个步骤的样本与嵌入维数比较小的时候，sherman - morrison - woodbury 引理(例如，[7])可以被用于提高效率。

当每一步的样本比起嵌入的维度小时，Sherman-Morrison-Woodbur 引理(如 [7])可以被用于提高效率。这个随机牛顿更新法类似等式 7，除了 η=1/τ，梯度被替换成样本估计 $\hat{p}$，Hessian 矩阵被替换成样本估计 $\bar{q}$。

5.1 收敛性

我们考虑了随机牛顿法的三种性质，将它们合在一起，这是将经验损失 L 收敛到局部最优的充分条件[8]。通过将 Hessian 设置为单位矩阵，$\bar{q}(.) = I_k$，这些条件也得到了满足，也就是随机梯度法。

Local Convexity（局部凸）：损失必须是局部凸，其最小值必须包含在其域内。在交替投影中，对于任何一个布雷格曼散度，损失都是凸的;对于普通散度也是一样，R 是它的定义域。非普通散度，如 Hinge 损失，也满足该特性。

Uniformly Bounded Hessian（一致有界 Hessian）：样本的特征值在某些区间[-c,c]的概率为 1。该条件通过测试样本 Hessian 的数量是否低于一个较大的固定值来满足。Hessian 是可逆的。使用 l_2 范数总是产生一个满秩的 Hessian 矩阵。特征值条件意味着 q 的元素和它的逆是一致有界的。

Convergence of the Hessian（Hessian 的收敛性）：Hessian 的收敛标准有两种。任何一个都足以证明随机牛顿的收敛性。(i)Hessian 样例的逆序列收敛于真实 Hessian 的概率：$\lim_{\tau \to \infty} (\bar{q}_{\tau})^{-1} = (q^{'})^{-1}$；(ii)从它的平均值上看，Hessian 样本的扰动是有界的。令 $P_{\tau-1}$ 包含随机梯度迭代的历史：第 τ-1 轮迭代的数据样本和参数。令 $g_t = o_s(f_t)$ 表示一个接近一致有界的随机数量级。随机 o 符号类似于普通的 o 符号，除了我们可以忽略为零的事件，且 $E[o_s(f_t)] = f_t$。替代的收敛标准是度量表达式的集合：

$$\mathrm{E}[\bar{q}_\tau \mid P_{\tau-1}] = \bar{q}_\tau + o_s(1/\tau)$$

对于等式八，这个条件很容易验证：

$$\mathrm{E}[\bar{q}_\tau \mid P_{\tau-1}] = (1-\frac{2}{\tau})\bar{q}_{\tau-1} + \frac{2}{\tau} E[\hat{q}'_\tau \mid P_{\tau-1}]$$

P_{t-1}包含 q_{t-1}。任何来自均值的扰动都是因为第二项。如果 q 是可逆的，那么它的元素是一致有界的，对于$\mathrm{E}[\hat{q}_\tau \mid P_{\tau-1}]$的元素也是一样。由于这项都是有界元素，并且被缩放至 2/τ，扰动为 $O_s(1/\tau)$。只要移动的平均 q 保持可逆，就可能得到不可逆的 Hessian 矩阵。以上证明了一个因子的收敛性，假设其他因子都是固定的，使其最小化期望损失。对于交互投影迭代算法，我们只能收敛到经验损失 L 的局部最优。

6. 相关工作

集体矩阵分解为关系数据提供了矩阵分解的统一视角：不同的方法对应于单个矩阵的不同分布假设，不同的模式结合因素，以及不同的优化过程。我们把我们的工作与以前的方法区分为三点：(i)相互竞争的方法通常会施加一个聚类约束，而我们同时涵盖了聚类和因子分析(虽然我们的实验侧重于因子分析)；(ii)我们的随机牛顿方法使我们可以利用损失的分解性来处理大而稀疏的关系；(iii)我们的报告更全面，涵盖了更广泛的模型、模式和损失。特别地，对于(iii)，我们的模型强调，分解两个矩阵与三个或更多相比是没有什么区别的。并且，我们的优化过程可以使用任何二阶可导的可分解损失，包括布雷格曼散度。例如，如果将模型限制为单个关系 ε_1- ε_2，就可以还原成 2.2 节中所有提到的单矩阵模型。虽然我们的交替投影方法在概念上很简单，并且能够利用可分解性，但对于单个矩阵的分解，有一整套替代方案。其中最流行的方法是 majorization[22]，它迭代地最小化了与目标相切的凸上界函数的序列，包括 NMF 的乘法更新[21]和 EM 算法，它用在 pLSI[19]和加权 SVD[32]中。对(U,V)使用梯度或二阶方法，直接优化解决了非凸问题，如 fast variant of max-margin matrix 分解[30]。

更普遍的是三实体类型模型 $\varepsilon_1 \sim \varepsilon_2 \sim \varepsilon_3$。这种模式的一个众所周知的例子是 pLSI-pHITS[13]，它对文档-单词计数和文档-文档引用建模：ε_1=words，ε_2= ε_3=documents。给定关系 $\varepsilon_1 \sim \varepsilon_2$和 $\varepsilon_2 \sim \varepsilon_3$以及对应的整数关系矩阵 $X^{(12)}$和 $X^{(23)}$，似然函数为

$$L = \alpha X^{(12)} o \log(UV^T) + (1-\alpha) X^{(23)} o \log(VZ^T) \qquad (9)$$

其中，参数 U、V、Z 对应着概率 $u_{ik}=p(x_i^{(1)}|h_k)$，$v_{ik}=p(h_k|x_i^{(2)})$，$z=p(x_i^{(3)}|h_k)$，$\{h_1,\ldots,h_k\}$。U，V^T和 Z 的每一列概率和为 1。由于不同的实体可以参与不同数量的关系中（例如，一些一些单词比其它的更普遍），所以数据矩阵 $X^{(12)}$和 $X^{(23)}$通常是正规化后的。我们可以用权重矩阵进行正规化编码。等式 9，是加权融合了 two probabilistic LSI[19]与共享的隐因子 h_k。由于每个 pLSI 模型都是我们通用模型的单矩阵示例，所以双矩阵的版本可以放在我们的框架内。

矩阵协同聚类技术有一个随机约束:如果一个实体在一个簇中增加其成员，它必须减少其在其他簇中的成员。矩阵和关系聚类的例子包括 pLSI、pLSI - pHITS、Long etal 的对称块模型等[23,24,25]和 Bregman tensor clustering[5]（它可以处理更多元的关系）。因子分析的矩阵类似物对参数没有随机约束。集体矩阵分解使用矩阵因子分析，带 $U^{(r)}$每行和为 1 的随机约束，在每次更新的 $U^{(r)}$上分配一个交替投影的等值约束。这种额外的等式约束

可以用拉格朗日乘子加入到牛顿法迭代的步骤中，产生无约束的优化([9])。将扩展的集体矩阵分解与上面的替代方法进行比较是未来工作的一个主题。应该注意的是，选择 $X = UV^T$ 并不是矩阵分解的唯一方法。Longet. al.[23]提出对称块模型 $X \approx C_1 A C_2^T$，其中 $C_1 \in \{0,1\}^{n1xk}$，$C_2 \in \{0,1\}^{n2xk}$ 是簇的索引矩阵，$A \in R^{kxk}$ 包含行和列簇的每个组合的预期输出。这个模型的早期研究使用了一个特定平方损失的光谱松弛(spectral relaxation)[23]，随后对常规指数家族[25]的推广使用 EM 求解。一个等价的根据普通布雷格曼散度[24]的公式，使用迭代的 majorization[22,34]作为交互迭代投影的内层循环。改进的 Bregman co-clustering 导致矩阵的系统偏差，块效应[1]。

三因子模式 $\varepsilon_1 \sim \varepsilon_2 \sim \varepsilon_3$ 也包括监督矩阵分解。在这个问题中，目标是对类型 ε_2 的实体进行分类:矩阵 $X^{(12)}$ 根据一个或多个相关概念(每行一个概念)包含类标签，而 $X^{(23)}$ 列出了每个实体的特征。一个监督矩阵分解算法的例子是支持向量分解机[29]：在 SVDMS 中，特征 $X^{(23)}$ 使用平方误差分解，而标签 $X^{(12)}$ 使用合页损失。朱等人提出了一个类似的模型[37]，使用基于合页损失的一次可微分变种。另一个例子是监督的 LSI[35]，它使用平方损失对数据和标签矩阵做因子分解，对共享的因子加上正交性约束。主成分分析，使用平方损失分解双中心矩阵，也被扩展到三因子模式[36]。

另一种有趣的模式包含两个实体类型之间的多个并行关系。这种模式的一个例子是 max-margin matrix factorization (MMMF)[30]。在 MMMF 中，目标是预测序数值，例如用户对电影的评分，范围在{1,...,R}。我们可以将这个预测任务简化为一组二元阈值问题，换句话说，预测 r≥1,r≥2,...,r≥R。如果我们使用合页损失在每个二元预测中，并把损失加起来,结果就等价于集体矩阵分解,其中 ε_1 是用户，ε_2 是电影，$\varepsilon_1 \sim_u \varepsilon_2$ 对于 u=1,...,R 是二元评分预测。为了预测 R 不同关系的不同值，我们需要允许隐因子 $U^{(1)}$ 和 $U^{(2)}$ 包含一些未解的列，即，在关系中不共享的列。例如，MMMF 的作者建议在每个级别或每个(用户、评分等级)对中添加一个偏差项。为了得到对每对(用户，评分等级)的偏差，我们可以将 R 的不解列附加到 $U^{(1)}$ 上，并且每个列都乘以 $U^{(2)}$ 中的固定列。为了得到每个评分等级的共享偏差，我们可以做相同的事情，但是要约束 $U^{(1)}$ 中的每一个未解列，以成为元素全为 1 的向量的倍数。

7. 实验

7.1 电影评分预测

实验关注两个任务：(i)预测用户是否对某部特定电影评分；(ii)预测用户对电影的评分数值。用户评分数据从 Netflix Prize 数据集[27]采样：评分数据有五个值（1-5 个星）。我们从网络电影数据库[20]中增加了两个额外的电影信息来源：每一部电影的类型，编码为二元关系，HasGenre(电影,类型)；每个电影中的演员列表，编码为二元关系，HasRole(演员,电影)。在这个模式中，ε_1 对应着用户，ε_2 对应着电影，ε_3 对应着类型，ε_4 对应着演员。评分标记为 $\varepsilon_1 \sim_1 \varepsilon_2$，是否评分任务，即二值化版本的评分标记为 $\varepsilon_1 \sim_2 \varepsilon_2$。类型关系标记为 $\varepsilon_2 \sim \varepsilon_3$，角色关系标记为 $\varepsilon_2 \sim \varepsilon_4$。

这两项任务的数据有着显著的差异。在是否评分问题中，我们知道一个用户是否有对一个电影评过分，所以这个评分矩阵是没有缺失值的。在评分数值预测问题中，我们只能观测到用户评过分的电影的关系，未观测到的用户于电影关系，数据权重设置为零。

7.1.1 模型和优化参数

为了保持一致性，我们在整个实验中控制了许多模型和优化参数。在是否评分任务中，

所有关系都是二值的，所以我们使用 logistic 模型：sigmoid 映射配上对数损失。为了评估测试误差，我们使用了平均绝对误差(MAE)，这是二分类预测的平均 0-1 损失。由于被评分的数据与没有被打分的是高度不平衡的，所以我们降低评分的权重。我们一直使用 l_2 正则。除非另有规定，否则正则项都为 $G(U)=10^5||U||_F^2/2$。在牛顿法步骤中，我们使用 Armijo 线性搜索，拒绝步长小于 $\eta=2^{-4}$ 的更新。在牛顿法步骤中，我们一直训练直到训练损失的变化降到 5%以下。使用随机牛顿，运行固定的迭代次数。

7.2 关系提升预测

我们对于关系数据的主张是，集体分解比单个矩阵分解有更好的预测。我们在两个相关的小型数据集上做是否评分预测任务，这允许我们能够反复实验。由于这个任务涉及到一个三因子模型，所以有一个混合因子，等式 3 的 α。我们通过使用完整的牛顿步骤，从相同的初始随机参数开始学习不同 α 的模型。在测试集上的评测，根据测试集的权重从矩阵中采样的实体，对每个 α 进行度量。每次试验重复十次，提供 1 个标准差误差。

有两个场景。第一个，对用户和电影进行随机采样;在电影中超过 1%的电影类型都保留了下来。我们只在采样的电影中使用用户的评分。第二个，我们只对最多 40 部电影的用户进行抽样，这大大降低了每个用户和每部电影的评分数量。在第一种情况下，每个用户的评分数量中位数是 60(平均值，127);在第二种情况下，每个用户的评分数量中位数是 9(平均值，10)。在第一种情况下，每部电影的被评分数量为 9(平均值，21);在第二种情况下，每部电影的被评分数量中位数是 2(平均值，8)。在第一种情况下，我们有 $n_1 = 500$ 用户和 $n_2 = 3000$ 电影，在第二种情况下，我们有 $n_1 = 750$ 用户和 $n_2 = 1000$ 电影。我们在两个矩阵中使用 $k = 20$ 的嵌入维数。

密集的评级场景(图 1)显示，集体矩阵分解提高了预测任务:用户对电影的评级，以及电影所属的类型。当 $\alpha = 1$ 时，模型只使用评级信息;当 $\alpha = 0$ 时，它只使用类型信息。

在稀疏评级场景中，图 2 中，评级矩阵中的信息要少得多。一半的电影只被一两个用户评价。因为用户之间的信息太少，所以额外的类型信息更有价值。然而，由于几乎没有用户对同一部电影进行评价，所以在类型预测方面没有明显的改进。

我们假设，除了电影类型之外，增加受欢迎演员的角色会进一步提高效果。根据对称性，演员因子的更新方程类似于电影类型因子的更新。因为我们的数据中有超过 10 万的演员，其中大多数只出现在一两部电影中，我们选择了 500 个受欢迎的演员(在十部以上的电影中出现)。在各种设置下混合参数 $\{\alpha^{(12)}, \alpha^{(23)}, \alpha^{(24)}\}$ 对是否评分和评分数值预测任务没有显著的提升。

7.3 随机近似

我们对于随机优化的看法是，它提供了一个有效的替代牛顿更新的交替投影算法。因为我们的兴趣是在大量的关系中，解决带电影类型数据的是否评分任务。$n_1 = 10000$ 个用户，$n_2 = 2000$ 部电影，$n_3 = 22$ 个数据集中最常见的电影类型。混合系数 $\alpha = 0.5$。两个因子的嵌入维度 $k = 30$。

在三个因子的问题上，我们采用牛顿和随机牛顿法的集体矩阵分解方法，每一行有 25、75 和 100 个样本。数据批次大小比电影类型的数量大，所以它们都被使用。我们的主要关注点是对较大的用户电影矩阵进行采样。利用牛顿法，进行 10 个周期的交替投影;使用随机牛顿法进行 30 个周期。在每个周期之后，我们测量训练损失(对数损失)和测试误差(平均绝对错误)，根据图 3 中所需要的 CPU 时间来绘制。这个实验重复了 5 次，产生 2 个标准差。

仅使用一小部分数据，我们在五次迭代后得到与完整的牛顿法相似的结果。在批量大小

为 100，我们采样 1%的用户和 5%的电影；然而，它在测试数据上的性能与一个完整的牛顿步骤相同，但完整的牛顿法的运行时间长了八倍。批量大小递减的回报表明，使用大批次是不必要的。即使批量大小等于 $\max\{n_1, n_2, n_3\}$ 的随机牛顿，也不会返回与完整牛顿相同的结果，因为在样例 Hessian 上的 $1/\tau$ 阻尼因子。

值得注意的是，评分数值预测是一个计算上更简单的问题。在 $n_1 = 100000$ 用户，$n_2 = 5000$ 电影和 n3 = 21 电影类型的三因子问题上，在单个 1.6 GHz 的 CPU 上，32 分钟内，用完整的牛顿步骤进行的交替投影会在 32 分钟内达到收敛。我们使用一个小的嵌入维度，k = 20，但是对于大型的 Hessian 矩阵，可以利用一些常见的技巧。我们使用了泊松映射来进行评分预测，对于电影类型使用逻辑回归映射；在恒等映射下，收敛速度通常更快。

7.4 与 pLSI-pHITS 比较

在本节中，我们提供了一个例子，说明集体矩阵分解的额外灵活性会带来更好的结果；另一种模式是，协同聚类模型，即 pLSI-pHITS，也具有这个优点。

我们选取了两个是否评级任务的实例，控制每个电影的评级。在密集的数据集中，每部电影(用户)评分数量的中位数为 11(76)；在稀疏数据集中，每部电影(用户)评分数量的中位数为 2(4)。在这两种情况下，都有 1000 个随机选择的用户，以及 4975 个随机选择的电影，并且所有的电影都在稠密的数据集中。

由于 pLSI-pHITS 是一种协同聚类方法，而我们的集体矩阵分解模型是一种链路预测方法，因此我们选择了一种合适的度量方法，排序。我们为每个用户引入电影排名，评价指标使用平均准确率(MAP)[18]：查询对应于用户对评分的要求，“相关”条目是被链接的电影，我们在每个排名中只使用前 200 个电影，平均是针对用户的。大多数电影都不被任何给定的用户评分，因此相关只适用于一小部分内容：MAP 的绝对值很小，但是相对的差异很大。我们比较了四种不同的为用户提供电影排名的模型：

CMF-Identity：使用恒等映射的集体矩阵分解，$f_1(\theta) = f_2(\theta) = \theta$，使用平方损失。使用完整的牛顿步骤。正则化和优化参数与 7.1.1 小节的描述一样，除了最小步长 $\eta = 2^{-5}$。用户 i 的电影排序是由 $f(U_i V^T)$ 引起的。

CMF-Logistic：类似 CMF-Identity，除了匹配映射和损失对应着伯努利分布，就像逻辑斯蒂回归：$f_1(\theta) = f_2(\theta) = 1/(1+\exp^{-\theta})$

pLSI-pHITS：对每个矩阵进行多项式假设，这对于评分预测任务来说是不自然的——五分制的评分不意味着用户和电影参与一个评分关系五次。因此我们使用是否评分任务。我们给 pLSI-pHITS 添加正则项。正则 $\beta \in [0, 1]$，在每次迭代时使用 tempered EM 来选值。β 越小，参数平滑的均匀分布越强。我们对 β 的设置也比 Cohn 等人[13]小心，我们使用 0.95 的衰减率并且最小值为 0.7。为了对该方法和 CMF 之间的迭代进行一致的解释，我们使用退火法来选择正规化的数量，然后选择一个随机的初始状态与最优的 β 去拟合参数。电影排序通过使用 p(movie|user)生成。

Pop：忽略电影类型的基准方法。它为所有用户生成一个排序的电影，按照频率排序。

在每种情况下的模型，保存流行度排名、嵌入维数 k = 30，并在最多 10 次迭代中运行。我们比较了 α 的各种值，但我们不认为混合信息提高了排名的质量。由于 b 是一个自由参数，我们需要在几个值中确定这些方法的相对效果。图 4，在稠密的数据集上，集体矩阵分解显著优于 pLSI-pHITS；反过来，在稀疏数据集中，后者优于前者。在任何方法、数据集中，评分数值预测都不会从混合信息中受益。虽然集体矩阵分解的灵活性有其优点，特别是计算性的优点，但我们并没有提出这个模型比基于矩阵协同聚类的关系模型有着绝对优势。

8. 贡献

我们提出了矩阵分解的统一观点，并以此为基础，提供了集体矩阵分解作为一种pairwise关系数据的模型。实验证据表明，从多个关系中混合信息会使我们的方法得到更好的预测，这与在关系共聚类[23]中所做的相同观察是相辅相成的。在一个可分解的、二次可微的损失的共同假设下，我们在一个交替投影的框架中得到一个完整的牛顿步骤。这在涉及数十万个实体和数以百万计的观测样本的关系型领域是可行的。我们提出了一种新的随机逼近在集体矩阵分解上的应用，它使得集体矩阵分解处理更大的矩阵，通过对梯度和Hessian矩阵近似采样，在实践中可证明收敛且有快速的收敛速率。

感谢

作者感谢Jon Ostlund在合并Netflix和IMDB数据方面的帮助。这项研究部分是由DARPA的RADAR项目资助的。意见和结论是作者单独提出的。

引用

[1] D. Agarwal and S. Merugu. Predictive discrete latent factor models for large scale dyadic data. In KDD, pages 26-35, 2007.

[2] D. J. Aldous. Representations for partially exchangeable arrays of random variables. J. Multi. Anal., 11(4):581-598, 1981.

[3] D. J. Aldous. Exchangeability and related topics, chapter 1. Springer, 1985.

[4] K. S. Azoury and M. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. Mach. Learn., 43:211-246, 2001.

[5] A. Banerjee, S. Basu, and S. Merugu. Multi-way clustering on relation graphs. In SDM. SIAM, 2007.

[6] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. J. Mach. Learn. Res., 6:1705-1749, 2005.

[7] L. Bottou. Online algorithms and stochastic approximations. In Online Learning and Neural Networks. Cambridge UP, 1998.

[8] L. Bottou and Y. LeCun. Large scale online learning. In NIPS, 2003.

[9] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge UP, 2004.

[10] L. Bregman. The relaxation method of nding the common points of convex sets and its application to the solution of problems in convex programming. USSR Comp. Math and Math. Phys., 7:200-217, 1967.

[11] Y. Censor and S. A. Zenios. Parallel Optimization: Theory, Algorithms, and Applications. Oxford UP, 1997.

[12] P. P. Chen. The entity-relationship model: Toward a unied view of data. ACM Trans. Data. Sys., 1(1):9-36, 1976.

[13] D. Cohn and T. Hofmann. The missing link{a probabilistic model of document content and hypertext connectivity. In NIPS, 2000.

[14] M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal component analysis to the exponential family. In NIPS, 2001.

[15] J. Forster and M. K. Warmuth. Relative expected instantaneous loss bounds. In

COLT, pages 90-99, 2000.
[16] G. H. Golub and C. F. V. Loan. Matrix Computions. John Hopkins UP, 3rd edition, 1996.
[17] G. J. Gordon. Generalized2 linear2 models. In NIPS, 2002.
[18] D. Harman. Overview of the 2nd text retrieval conference(TREC-2). Inf. Process. Manag., 31(3):271{289, 1995.
[19] T. Hofmann. Probabilistic latent semantic indexing. In SIGIR, pages 50-57, 999.
[20] Internet Movie Database Inc. IMDB interfaces. http://www.imdb.com/interfaces, Jan. 2007.
[21] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In NIPS, 2001.
[22] J. D. Leeuw. Block relaxation algorithms in statistics, 1994.
[23] B. Long, Z. M. Zhang, X. W u;, and P. S. Yu. Spectral clustering for multi-type relational data. In ICML, pages 585-592, 2006.
[24] B. Long, Z. M. Zhang, X. Wu, and P. S. Yu. Relational clustering by symmetric convex coding. In ICML, pages 569-576, 2007.
[25] B. Long, Z. M. Zhang, and P. S. Yu. A probabilistic framework for relational clustering. In KDD, pages 470-479, 2007.
[26] P. McCullagh and J. Nelder. Generalized Linear Models. Chapman and Hall: London.,1989.
[27] Netflix. Netflix prize dataset. http://www.netflixprize.com, Jan. 2007.
[28] J. Nocedal and S. J. Wright. Numerical Optimization. Springer, 1999.
[29] F. Pereira and G. Gordon. The support vector decomposition machine. In ICML, pages 689-696, 2006.
[30] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In ICML, pages 713-719, 2005.
[31] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. Technical Report CMU-ML-08-109, Machine Learning Department, Carnegie Mellon University, 2008.
[32] N. Srebro and T. Jaakola. Weighted low-rank approximations. In ICML, 2003.
[33] N. Srebro, J. D. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization.In NIPS, 2004.
[34] P. Stoica and Y. Selen. Cyclic minimizers, majorization techniques, and the expectation-maximization algorithm: a refresher. Sig. Process. Mag., IEEE, 21(1):112-114, 2004.
[35] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In SIGIR, pages 258-265, 2005.
[36] S. Yu, K. Yu, V. Tresp, H.-P. Kriegel, and M. Wu. Supervised probabilistic principal component analysis. In KDD, pages 464-473, 2006.
[37] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classication using matrix factorization. In SIGIR, pages 487-494, 2007.