Restricted Boltzmann Machines for Collaborative Filtering - 论文阅读笔记

标题：Restricted Boltzmann Machines for Collaborative Filtering

作者：Ruslan Salakhutdinov，Andriy Mnih，Geo reyHinton

单位：University of Toronto

发表于：ICML 2007

引自：http://blog.csdn.net/xceman1997/article/details/9901395

主要内容：

这篇文章尝试把 RBM 应用到协同过滤中，在 netflix 上的数据集做实验，RBM 方法与 SVD 方法线性插值相结合，能提高系统性能 6%左右。

问题描述：对电影的推荐，用户对电影集合中的某个电影进行打分，分值区间为(0, k]。

基于 user-based 的协同过滤算法，用用户对电影的打分形成一个整数向量，用来表征用户特征，进一步计算用户之间的相似度。这篇文章引入 RBM，输入向量为用户对电影的打分的整数向量，从而学习隐含层，并用隐含层的数值来表示用户特征。进一步地，类似协同过滤，也可以用用户向量来计算用户之间的相似关系。不过作者没有按照这个思路做，而是直接用 RBM 的输入向量，从隐含层向显示层 reconstruction 的结果，来预测当前用户与未知电影之间的关系。针对每一个用户，训练一个 RBM。节选文章中的原话：

Every RBM has the same number of hidden units, but an RBMonly has visible softmax units for the movies rated by that user,soan RBM hasfew connections if that userrated few movies. Each RBM only has a single training case,but all of the corresponding weights and biases are tied together, so if two users have rated the same movie, their two RBM's must use the same weights betweenthe softmax visible unit for that movie and the hidden units.

RBM 的结构如下：

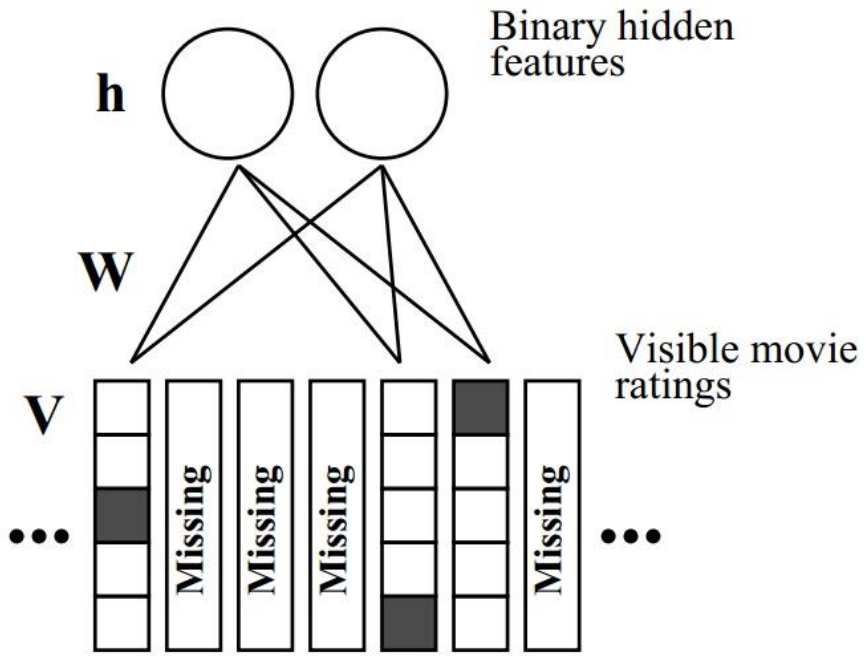

Figure 1. A restricted Boltzmann machine with binary hidden units and softmax visible units. For each user, the RBM only includes softmax units for the movies that user has rated. In addition to the symmetric weights between each hidden unit and each of the $K = 5$ values of a softmax unit, there are 5 biases for each softmax unit and one for each hidden unit. When modeling user ratings with an RBM that has Gaussian hidden units, the top layer is composed of linear units with Gaussian noise.

标准 RBM 的输入时 0-1 向量，上面的 RBM 的输入节点是个(0, k]的向量，表示用户对当前电影的打分。其时，将(0, k]向量向水平方向展开，就是标准 RBM 了。

用户的输入向量是个稀疏向量，包含了所有电影，但是很多电影用户并没有评分，就是上面的 missing 的部分。这部分的输入如何取值？作者的解决方法是不考虑这部分节点，在模型训练的时候，只考虑用户已经有评分的电影、以及相关联的权重。用 CD-k 的算法，来更新当前用户的 RBM 的权重。更新之后，输入向量转为下一个用户的输入向量，这时候 missing 的节点电影，就有可能被那个用户评分。这个用户，模型的训练过程同上，也是只针对这个用户所评分的电影，进行 RBM 权重更新——文章中说对每个用户单独建立 RBM 是不准确的，其实就是一个 RBM，只不过更新权重的时候，1. 不是所有的权重都要更新，2. 每个用户的输入作为单独一个样本，一个样本一个样本的训练。最终，所有样本

都训练完之后，形成 RBM 最终的权重，此时，将用户的输入向量输入进去，在隐含层得到的就是用户在这些电影上的兴趣表示了。

RBM 的训练过程仍然用的 CD-k，与普通 RBM 没什么不同；不同之处在于某些权重（用户没有看过电影的节点对应的权重）没有参与当时的训练。

模型预测：

得到了 RBM，且得到了用户在 RBM 隐含层表示的用户兴趣向量，如何给用户推荐电影呢？两种思路：1. 采用协同过滤的框架，用用户表示向量来计算用户之间的相似度，然后用“相似用户看过的电影给当前用户进行推荐”。不过这篇文站采用的是另外一种方法——更加直接的方法。

本文的方法，是在得到 RBM 模型之后，用用户的电影评分向量作为输入向量，然后用 RBM reconstruct 用户的输入向量，包括那些 missing input（表示用户没看过的电影）。此时，这些 missing input 的值就是表示了用户对这个电影的喜好程度。电影有 k 个评分，RBM reconstruct 对这 k 个值（所对应的节点）都有输出。选择用户对这个电影的预测评分就有两种方法：1. 选择 k 个节点中 RBM 输出权值最大的那个节点作为用户对当前电影的评分；2. 对这 k 个节点的权重加权平均，平均值离 k 个数字哪个最近，就是评分是多少。在实践中，这两种方法都可以尝试一下。

模型改进：

1. conditional RBM

上面的模型有一个缺点，忽略了一个信息：用户可能看过这个电影，只不过没有评分而已。这个信息上面的模型没有利用。用户看过电影，也是表示对这个电影的一个喜好。conditional RBM 就是解决这个问题。它引入了一个辅助输入 0-1 向量，表示用户是否看过这个电影。如果看过，但是没有评分，那么对应的电影的用户输入（missing 状态，是随机值）以及对应的权重，也参与训练。作者认为这样会有助于提升模型性能。此时 RBM 的结构如下：

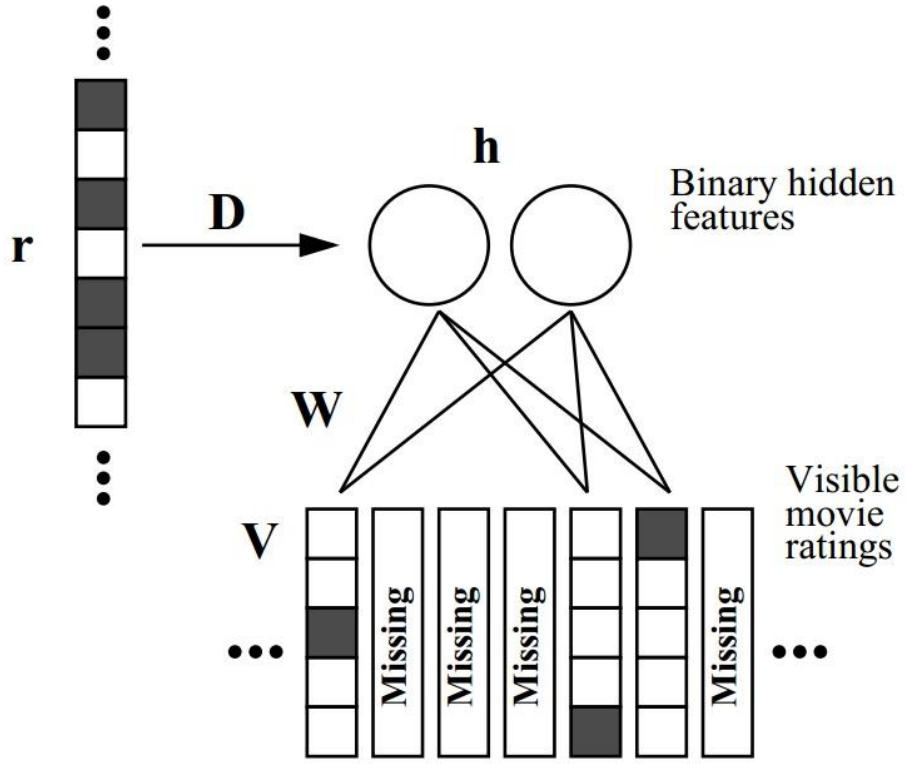

Figure 2. Conditional RBM. The binary vector **r**, indicating rated/unrated movies, affects binary states of the hidden units.

2. Conditional Factored RBM

完全训练 conditional RBM 会因为参数过多而变慢。Factored RBM 将 RBM 中的权重矩阵 W 分解为两个小矩阵 A 和 B 的乘积，CD-k 算法分别对这两个小矩阵进行权重更新（非常类似对 W 的更新），从而用减少参数的方式来加快训练速度。文章中称同 conditional RBM 相比，Conditional Factored RBM 的效果略差，但是速度很快。不过文章中的实验效果是 Conditional Factored RBM 的效果也更好。

模型试验：

结果自然是好，否则文章也发不了。conditional RBM 比最初提出的 RBM 要好。和 SVD 相比，Conditional Factored RBM 性能也要略好。

未来工作：

1. 用 autoencoder，这样隐含层就可以是实数向量。

2. 用深层神经网络（stack RBM）

完