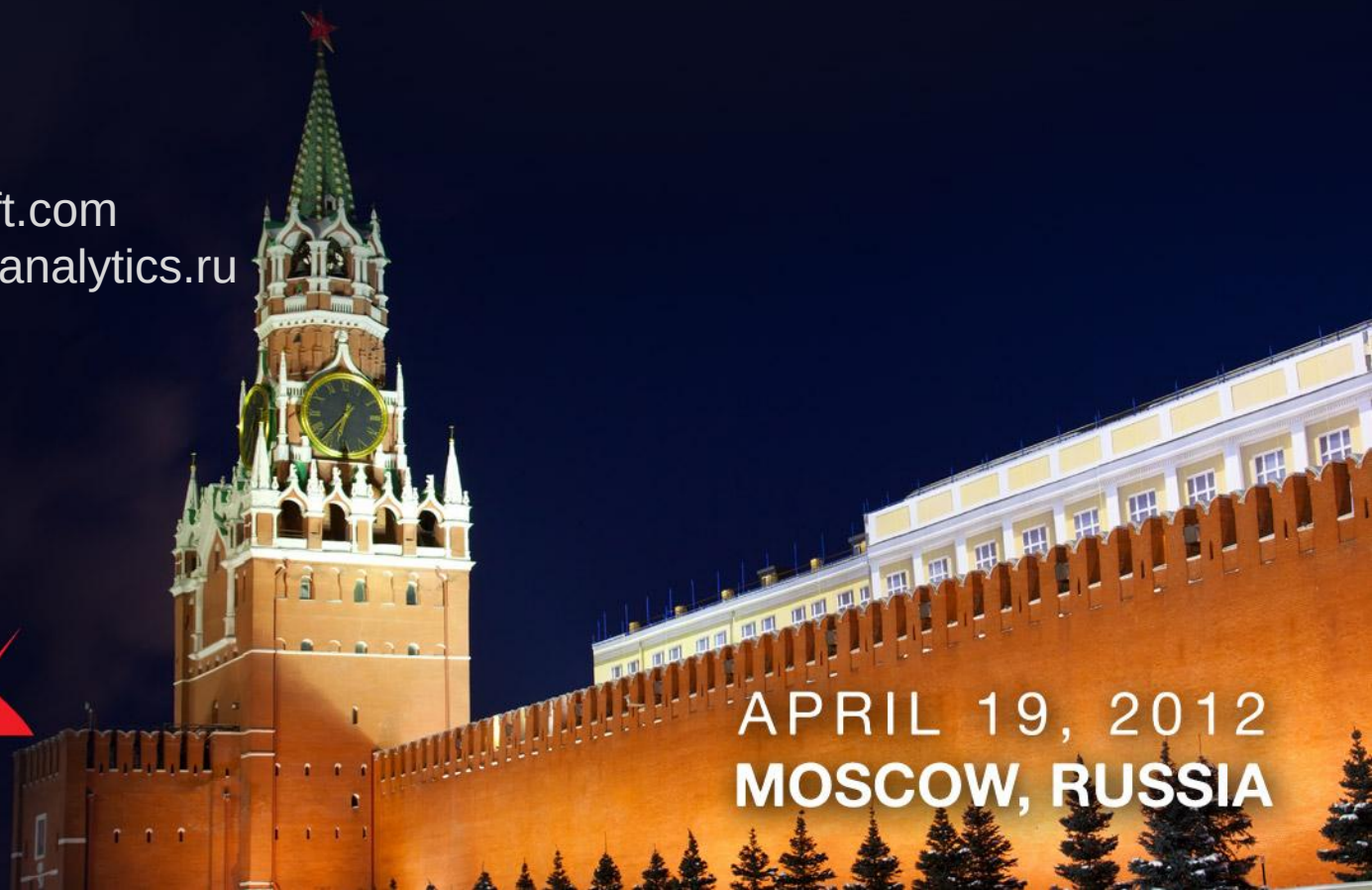
DATA MINING: РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ

Collaboration Filtering

- Максим Гончаров
  maxgon@microsoft.com
  www.businessdataanalytics.ru



PASS SQLRally RUSSIA 2012

Рекомендательные системы

- Подходы к формированию рекомендаций
  - Задачи
  - Подходы
    - Сбор данных: Explicit/Implicit
    - Content-based/Transaction-based
    - Model-based/Memory-based
    - User-based/Item-based
- Алгоритмы
  - Ассоциативные правила
  - Транзитивные ассоциативные сети
  - Collaboration Filtering
    - User-based
    - Item-based
  - Другие алгоритмы
- Big Data, Hadoop, Mahout etc

Рекомендательные системы: задачи и подходы

Задачи рекомендательных систем

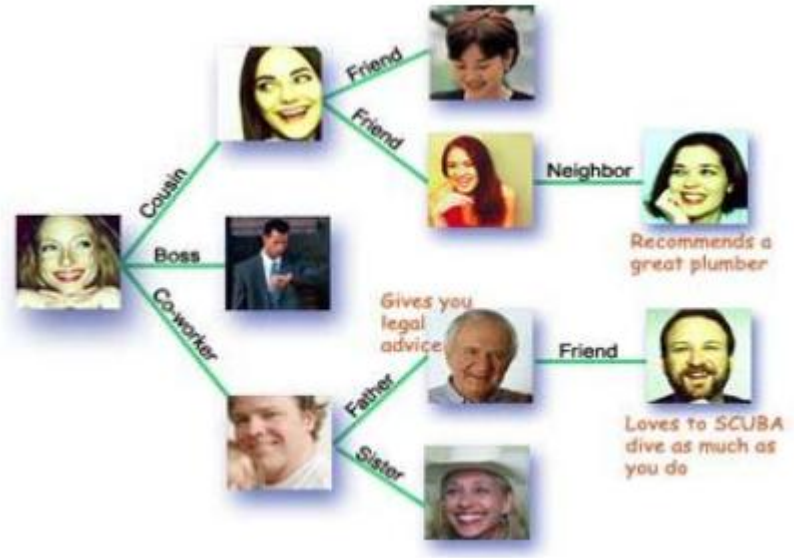
Рекомендательные системы (РС) применяются для предложения клиенту в реальном времени продуктов (фильмов, книг, одежды) или услуг, которые его с большой вероятностью заинтересуют.

Применения: **Cross-Sale, Up-Sale**

- электронная коммерция,
- он-лайн банкинг,
- розничная торговля,
- справочные центры,
- поиск фильмов, музыки, ПО, научных статей

Принцип рекомендательных систем

Рекомендации предоставляются на основании уже совершенных покупок, посещений сайтов, а также приема обратной связи: выставленных рейтингов, заполненных анкет.



Примеры:

- Amazon.com (Item-based CF),
- Музыка на Yahoo!,
- Cinemax.com,
- Moviecritic,
- TV Recommender,
- Video Guide,
- CDnow.com и проч.

Критерии оценки качества РС

- **Точность.** Разделяем транзакции на обучающее и тестовое множество. Транзакции в обучающем множестве служат для оценки рейтингов товаров. Товары из каждой тестовой транзакции случайно разделяются на две группы: «известные» и «неизвестные». На основании рейтингов группы «известных» товаров строятся рейтинги для группы «неизвестных».
- **Покрытие.** Для какой части товаров РС может формировать прогнозы рейтнига или выдавать рекомендации?
- **Скорость обучения.**
- **Степень новизны.** Можно представить РС, дающую точные прогнозы с большим покрытием, но бесполезные. Например рекомендации по самым популярным товарам.

Как собирать данные?

Используется **явный** или **неявный** сбор данных.

- При **явном** сборе от пользователя требуется заполнять анкеты для выявления предпочтений.
- При **неявном** сборе действия пользователя протоколируются для выявления предпочтений, и составления рейтингов происходит автоматически.

Иногда оба подхода комбинируются: когда транзакционной истории пока нет – используются опросы. Затем начинают учитывать транзакции.

Как вычислять похожесть?

Основной принцип: **похожим пользователям рекомендуют похожие товары.** Как определять похожесть?

- **Content-based**. Похожесть определяется на основании характеристик товаров и пользователей. Для товаров: сюжет, режиссер, киношкола; общее музыкальное направление, стиль; функциональное назначение, категория, ценовая группа. Для клиентов: демографические данные, предпочтения из заполненных анкет.
- **Transaction-based**. Товары считаются похожими, если часто входят вместе в одну транзакцию, а пользователи – если совершают схожие покупки.

Используется ли модель предметной области?

- **Memory-based.** Модели данных нет. Рекомендации формируются на основании вычисления некой меры похожести по всем накопленным данным. Этот подход проще, обладает высокой точностью, может использовать инкрементальный учет новых данных. Но ресурсоемок, не может предоставить описательный анализ существующих закономерностей, дать большее понимание имеющихся данных и объяснить прогноз.
- **Model-based.** Сначала формируется описательная модель предпочтений пользователей, товаров и взаимосвязи между ними, а затем формируются рекомендации на основании полученной модели. Преимущества: наличие модели, дающей б**о**льшее понимание формируемых рекомендаций, а также то, что процесс формирования рекомендаций разбит на два этапа: ресурсоемкое обучение модели в отложенном режиме и достаточно простое вычисление рекомендаций на основе существующей модели в реальном времени. Недостатки: не поддерживают инкрементального обучения, меньшая точность прогноза.

Collaboration Filtering

Collaboration Filtering – способ формирования рекомендаций: implicit (пользователи не опрашиваются), memory-based (без составления модели), transaction-based (на основе пользовательского поведения).

- **User-based.** Предложить товары, приобретаемые *похожими пользователями*: усреднить рейтинги товара, проставленные другими пользователями, с весами по степени похожести пользователей.
- **Item-based.** Предложить *товары*, *похожие* на уже приобретенные: усреднить рейтинги уже оцененных товаров с весами по степени похожести на неоцененный товар.

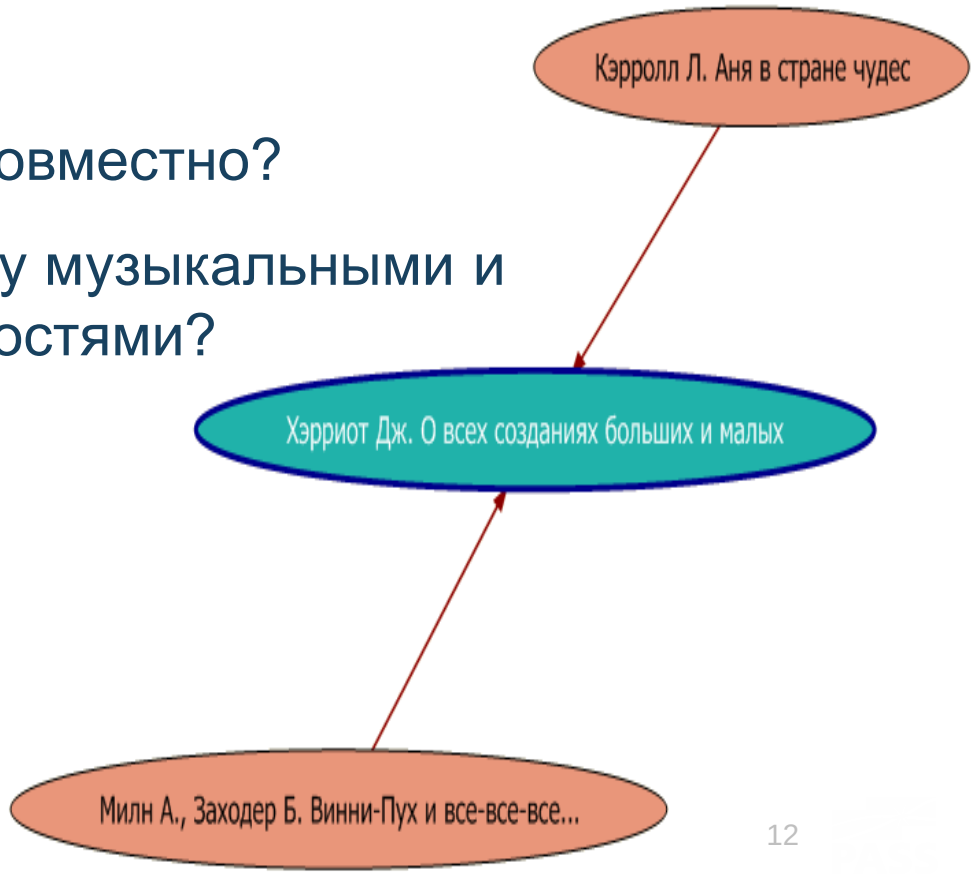
Рекомендательные системы: Алгоритмы

SQL Server Analysis Services Data Mining Microsoft Association Rules

Правила вида «если A и B, то с вероятностью x также C», описывающие, например, совместно приобретаемые услуги.

Поиск ассоциаций:

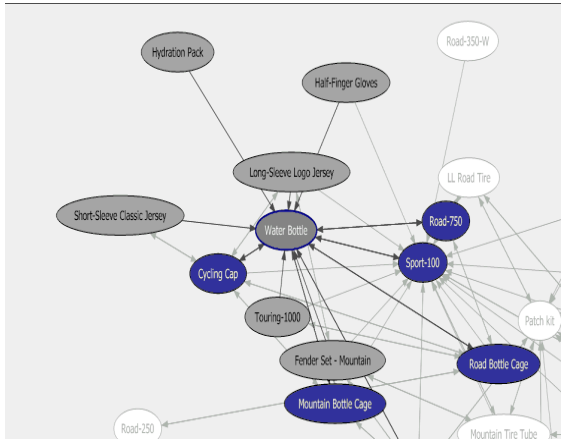
- Какие товары продаются совместно?
- Есть ли взаимосвязь между музыкальными и математическими способностями?

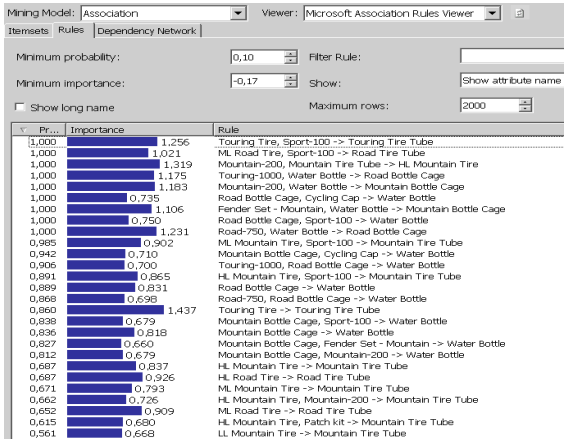


SQL Server Analysis Services Data Mining Microsoft Association Rules

Алгоритм a-priori

Алгоритм основан на итеративном вычислении частых наборов длиной i, т.е. наборов из i товаров, входящих более, чем в N чеков. Далее, наборы длиной i объединяются друг с другом для получения частых наборов длиной i+1. Как только все частые наборы найдены, они разбиваются на левую и правую часть для получения правил вхождения товаров из правой части набора при условии вхождения левой части. Оставляются только состоятельные правила (с большой вероятностью) и интересные (те, в которых левая часть сильно влияет на правую часть правила).



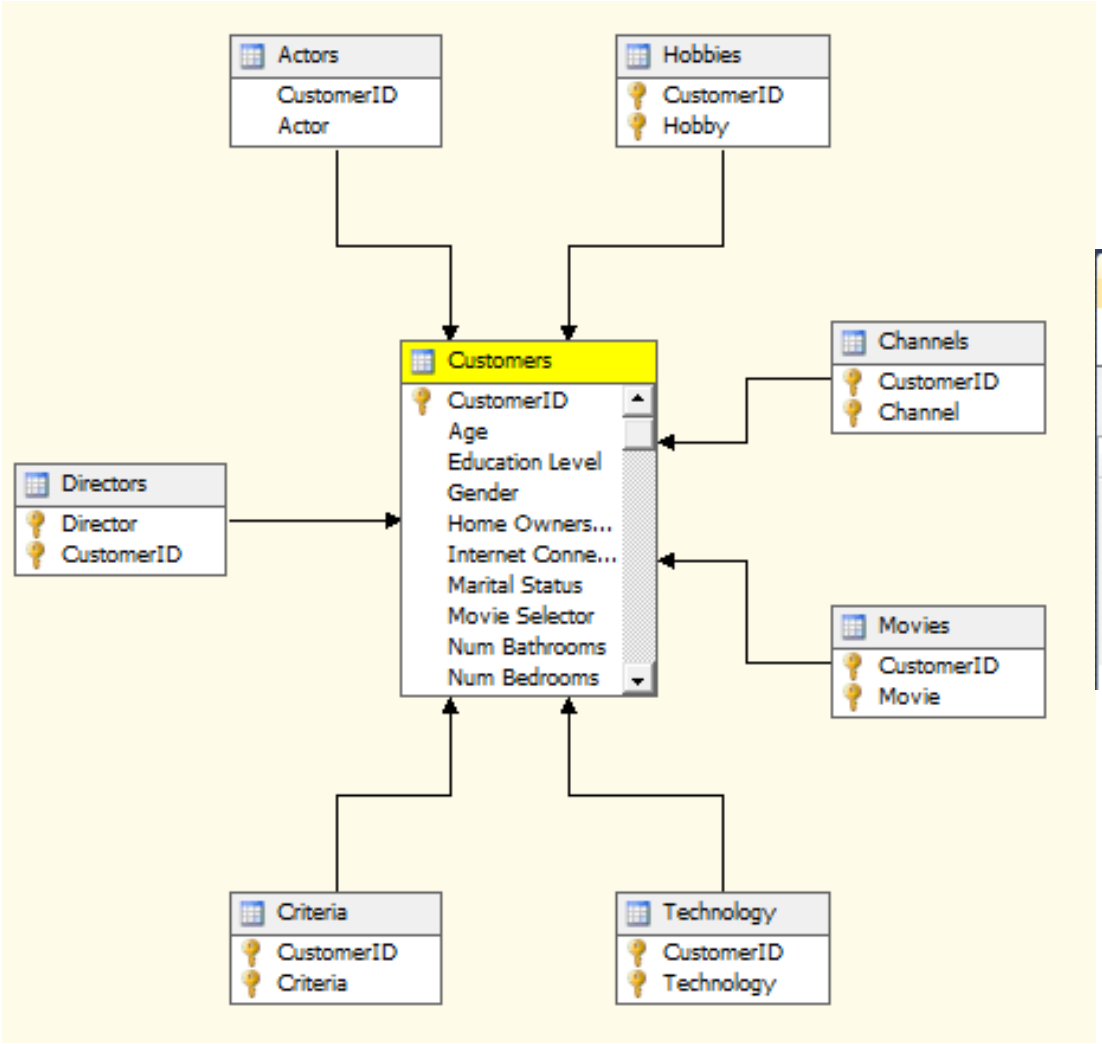


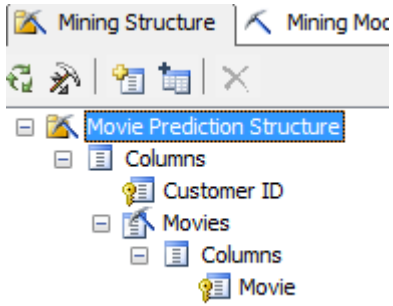
SQL Server Analysis Services Data Mining Microsoft Association Rules

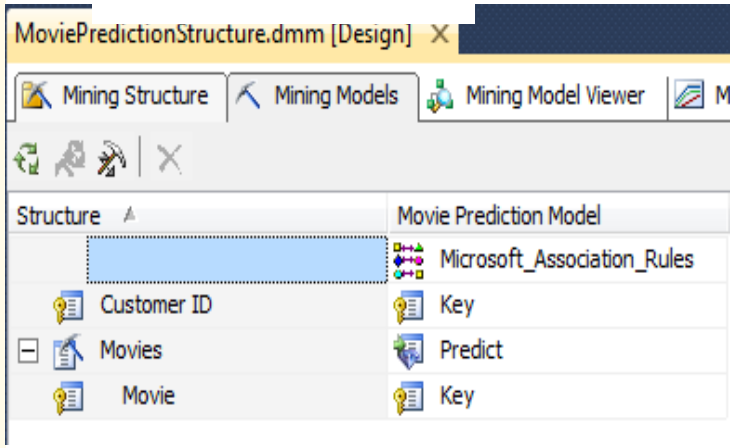
Параметры алгоритма a-priori

- Support – насколько часто корзина товаров встречается совместно в транзакциях.
- Confidence – насколько велика условная вероятность правой части правила при условии соблюдения левой части.
- Lift – насколько выполнение левой части правила увеличивает вероятность выполнения правой части.

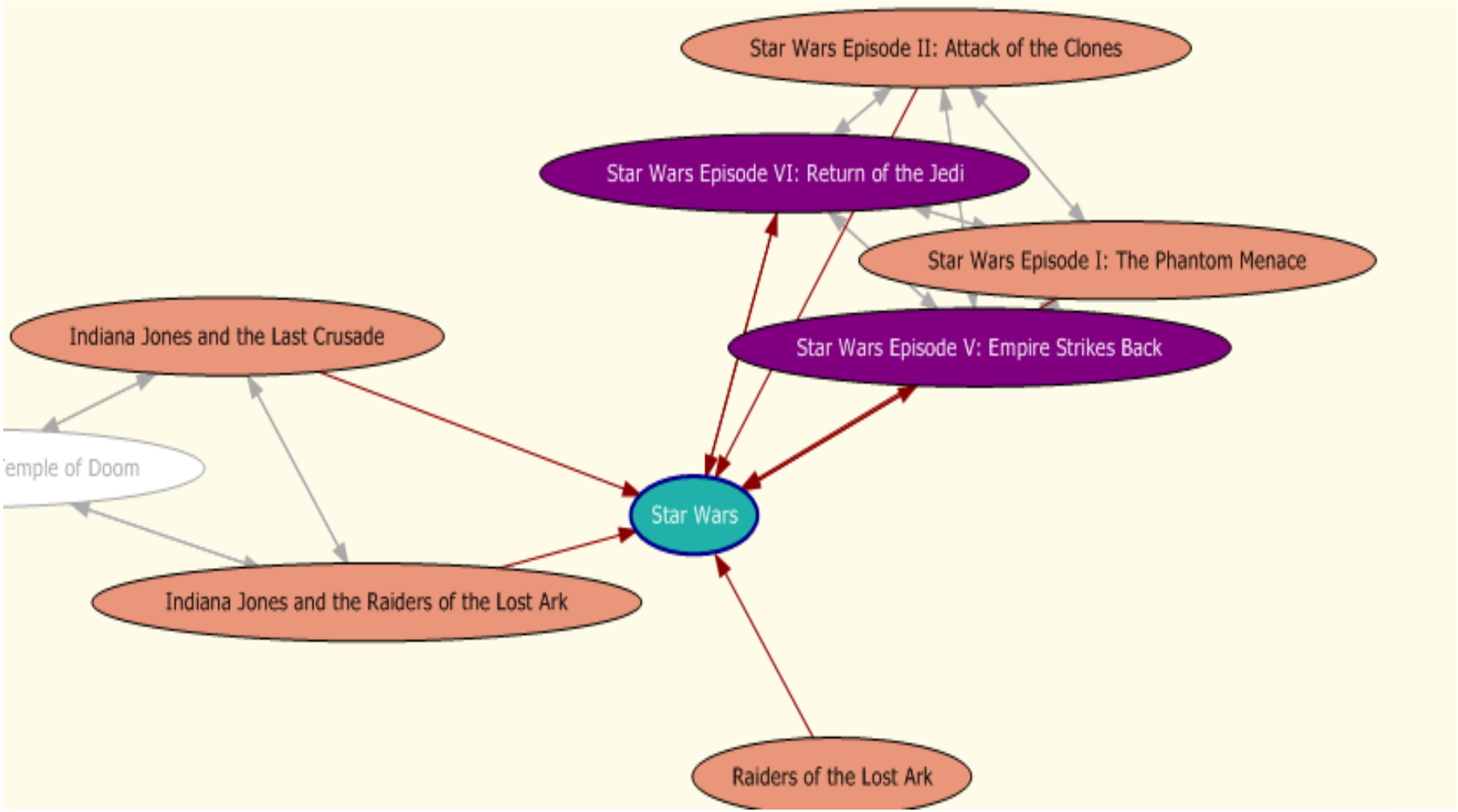
SQL Server Analysis Services Data Mining Microsoft Association Rules



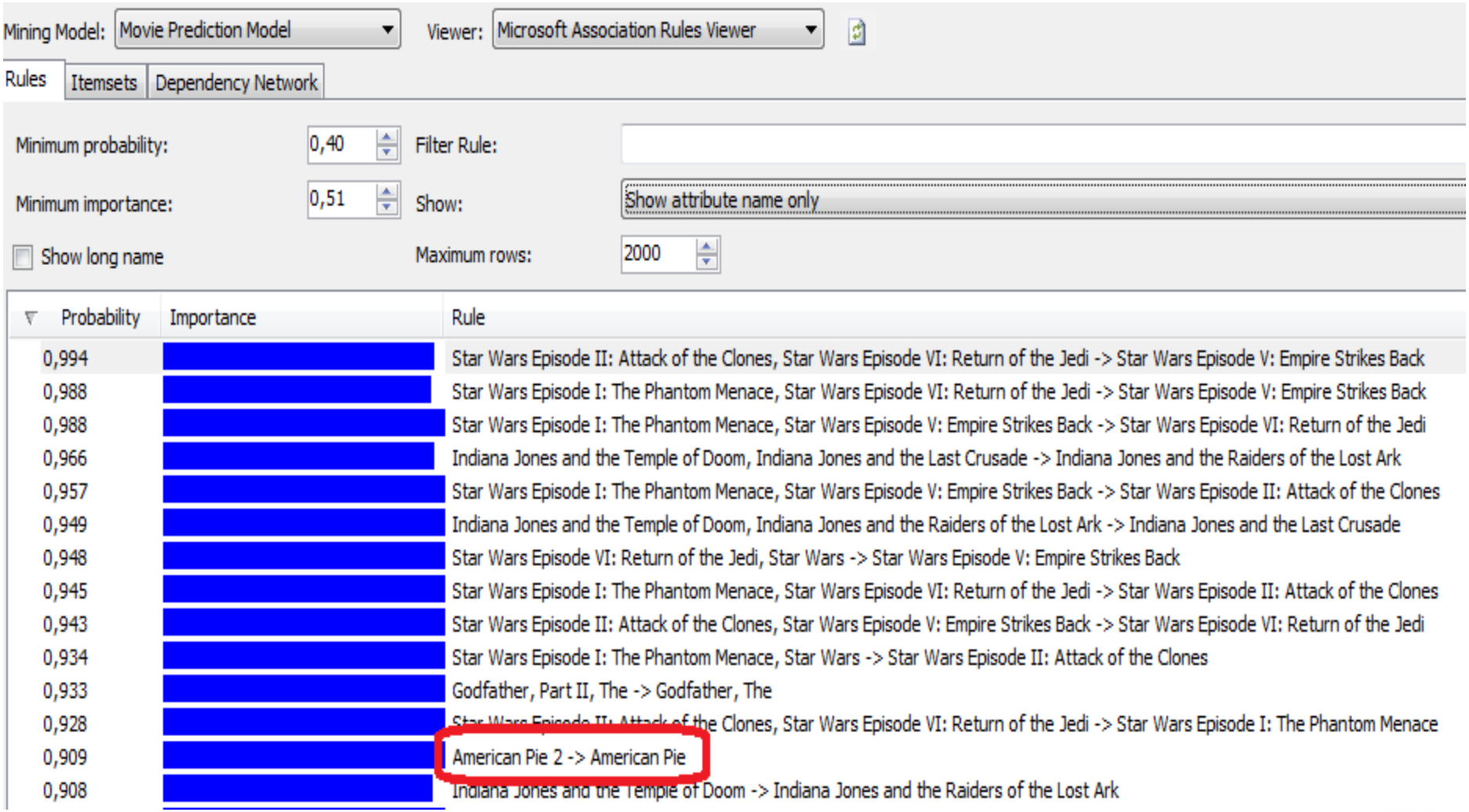




SQL Server Analysis Services Data Mining Microsoft Association Rules



SQL Server Analysis Services Data Mining Microsoft Association Rules



SQL Server Analysis Services Data Mining Microsoft Association Rules

Ассоциативные правила более предназначены для **описательного анализа**, т.е. для определения какие товары продаются совместно, а не для формирования рекомендаций. Это связано с тем, что у алгоритма низкое покрытие: правила существуют только для небольшого набора товаров.

Если у нас 1000 товаров и мы хотим для каждой пары товаров иметь возможность рекомендовать третий товар, то необходимо, чтобы транзакции, включающие все пары товаров + произвольный третий товар, были «частыми», т.е. встречались не менее 100 раз, получаем: минимум ~50 000 000 записей.

SQL Server Analysis Services Data Mining Microsoft Association Rules

DMX-запросы дают одинаковые результаты, т.к. не выявлены правила с участием левой части, поэтому рекомендуются наиболее популярные товары.

```
SELECT FLATTENED
(
    SELECT
        [Movie],
        $NODEID
    FROM PredictAssociation([Movie Prediction
Model].[Movies], 5, INCLUDE_NODE_ID)
) AS [Movie Prediction]
FROM [Movie Prediction Model]
NATURAL PREDICTION JOIN
(
    SELECT
    (
        SELECT '28 Days' AS [Movie]
    ) AS [Movies]
) AS t
```

Microsoft Association Rules

```
SELECT FLATTENED
(
    SELECT [Movie]
    FROM PredictAssociation([Movie Prediction Model].[Movies], 5)
) AS [Movie Prediction]
FROM [Movie Prediction Model]
NATURAL PREDICTION JOIN
(
    SELECT
    (
        SELECT '28 Days' AS [Movie]
    ) AS [Movies]
) AS t
```

00 %

Messages | Results

Movie Prediction.Movie
Star Wars
Matrix, The
Lord of the Rings: The Fellowship of the Ring, The
A beautiful mind
Star Wars Episode V: Empire Strikes Back

Microsoft Association Rules

```
SELECT FLATTENED
(
    SELECT [Movie]
    FROM PredictAssociation([Movie Prediction Model].[Movies], 5)
) AS [Movie Prediction]
FROM [Movie Prediction Model]
NATURAL PREDICTION JOIN
(
    SELECT
    (
        SELECT '28 Days' AS [Movie]
        UNION SELECT 'What''s New, Pussycat' AS [Movie]
    ) AS [Movies]
) AS t
```

100 %

Messages | Results

Movie Prediction.Movie
Star Wars
Matrix, The
Lord of the Rings: The Fellowship of the Ring, The
A beautiful mind
Star Wars Episode V: Empire Strikes Back

Microsoft Association Rules

```
SELECT FLATTENED
(
    SELECT [Movie]
    FROM PredictAssociation([Movie Prediction Model].[Movies], 5)
) AS [Movie Prediction]
FROM [Movie Prediction Model]
```

100 %

Messages | Results

Movie Prediction.Movie
Star Wars
Matrix, The
Lord of the Rings: The Fellowship of the Ring, The
A beautiful mind
Star Wars Episode V: Empire Strikes Back

Microsoft Association Rules

```
SELECT FLATTENED
(
    SELECT
        [Movie],
        $NODEID
    FROM PredictAssociation([Movie Prediction Model].[Movies], 5, INCLUDE_NODE_ID)
    --WHERE $NODEID <>''
) AS [Movie Prediction]
FROM [Movie Prediction Model]
NATURAL PREDICTION JOIN
(
    SELECT
    (
        SELECT '28 Days' AS [Movie]
        UNION SELECT 'What''s New, Pussycat' AS [Movie]
        --UNION SELECT 'American Pie' AS [Movie]
    ) AS [Movies]
) AS t
```

00 %

Messages | Results

Movie Prediction.Movie	Movie Prediction...
Star Wars	
Matrix, The	
Lord of the Rings: The Fellowship of the Ring, The	
A beautiful mind	
Star Wars Episode V: Empire Strikes Back	

Microsoft Association Rules

```
SELECT FLATTENED
(
    SELECT
        [Movie],
        $NODEID
    FROM PredictAssociation([Movie Prediction Model].[Movies], 5, INCLUDE_NODE_ID)
    --WHERE $NODEID <>''
) AS [Movie Prediction]
FROM [Movie Prediction Model]
NATURAL PREDICTION JOIN
(
    SELECT
    (
        SELECT '28 Days' AS [Movie]
        UNION SELECT 'What''s New, Pussycat' AS [Movie]
        UNION SELECT 'American Pie' AS [Movie]
    ) AS [Movies]
) AS t
```

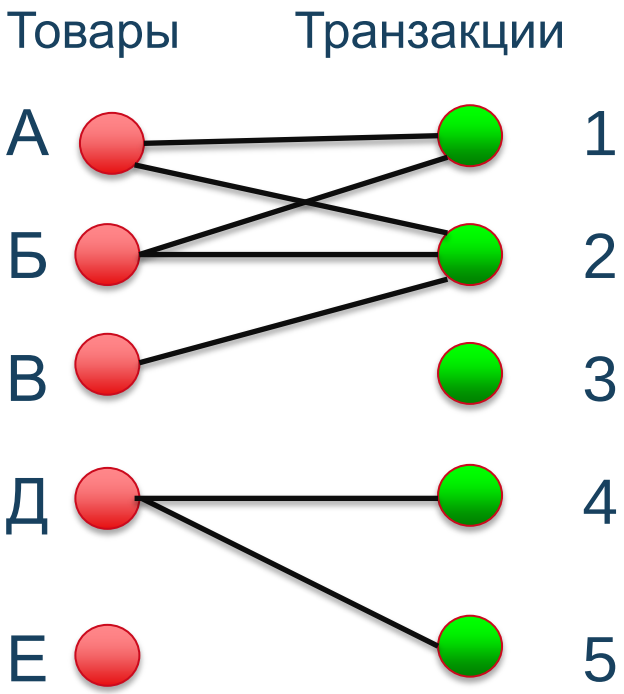
0 %

Messages | Results

Movie Prediction.Movie	Movie Prediction...
American Pie 2	71
Star Wars	
Matrix, The	
Lord of the Rings: The Fellowship of the Ring, ...	
A beautiful mind	

SQL Server Analysis Services Data Mining Microsoft Association Rules

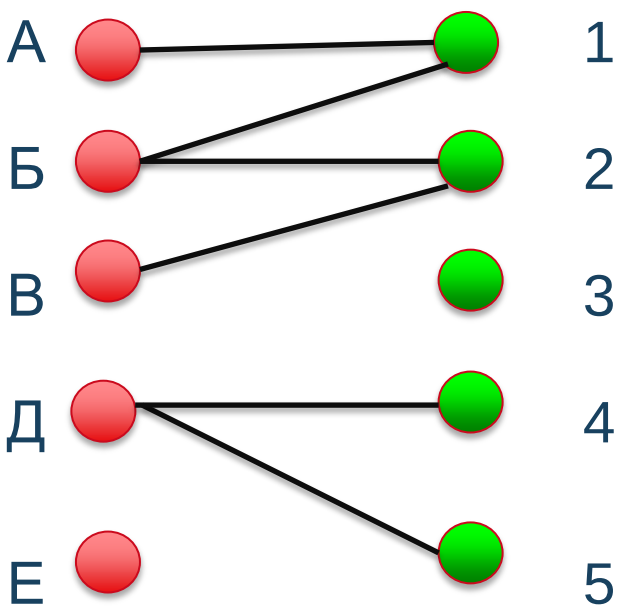
Ассоциативные правила: идея



Товары А и Б «сильно похожи», потому что вместе входят в транзакции 1 и 2.
Товары Б и В «менее похожи», потому что вместе входят в транзакцию 2.

Транзитивные ассоциативные сети

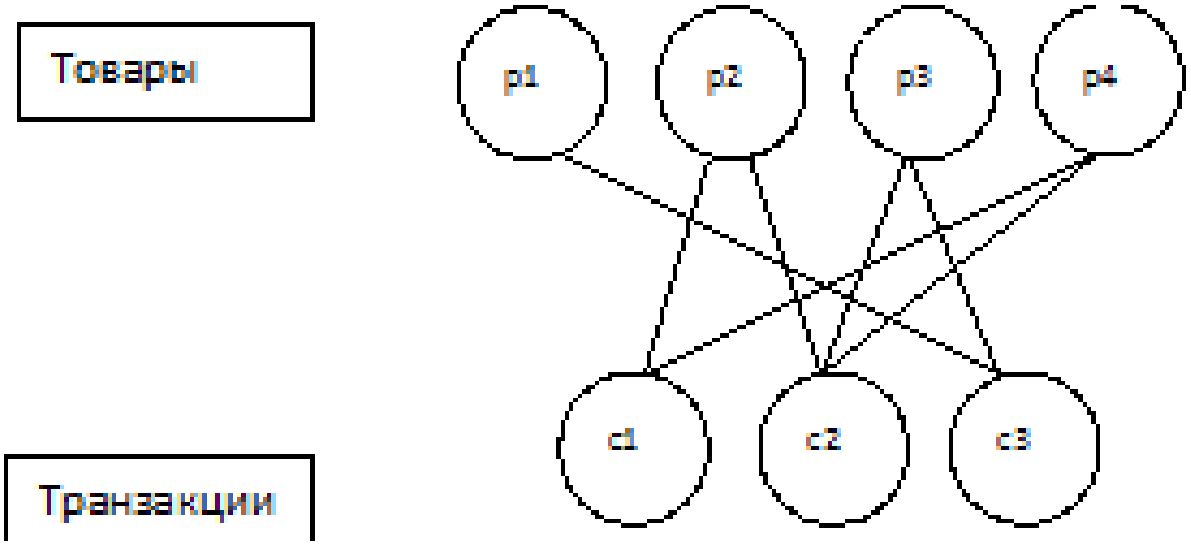
Расширение алгоритма ассоциативных правил. Мы считаем, что товары А и В также похоже, хотя они вместе не входят ни в одну транзакцию. А и В похожи потому что А ~ Б, Б ~ В.



Другими словами: между товарами А и В существует путь длины 2.

Транзитивные ассоциативные сети

Допустим, что пользователи c1 и c2 приобрели товар p1, а пользователи c2 и c3 приобрели товар p2. Стандартные алгоритмы совместной фильтрации свяжут пользователей c1 и c2, а также пользователей c2 и c3, но не свяжут c1 и c3. Для получения транзитивных связей между транзакциями используется граф, узлы которого состоят из двух частей – транзакции и товары, а дуги связывают транзакции с входящими в них товарами.



Транзитивные ассоциативные сети

Для учета транзитивных связей между товарами мы рассматриваем все пути между ними с длиной не больше заданного числа M. Стандартные алгоритмы совместной фильтрации учитывают пути длиной 1. Интуитивно понятно, что чем больше число различных путей, соединяющих два узла, тем сильнее связь между ними. Также интуитивно понятно, что чем длинней путь, связывающий два узла, тем слабее связь между ними.

Если A – матрица сопряженности (достижимости между товарами через одну транзакцию) между товарами, то A^i – матрица достижимости за i шагов (через i транзакции). Если складывать число путей одной длины между товарами с весом <1 в степени длины пути, то получим матрицу степени близости:

$$P(M) = \sum_{i=1}^{M} \rho^i A^i$$ Переходя к пределу при достаточно малых весах:

$$P(\infty) = I - \rho A^{-1} - I$$

Прогноз: товары наиболее близкие к данной транзакции (товарам транзакции).

Транзитивные ассоциативные сети

```
-- заполняем таблицу транзакций
DECLARE @CoOccurance AS input.CoOccuranceTable;
INSERT INTO @CoOccurance
(
    TransactionId,
    ItemId
)
SELECT
CustomerID,
Movie
FROM dbo.Movies

-- запрос на похожие товары:
SELECT *
FROM [cooccurance].[getItemDistance](N'28 Days', 1,
@CoOccurance)
ORDER BY similarity DESC

-- запрос на прогноз
SELECT *
FROM cooccurance.predictItemsFromStringWithoutTraining(2,
@CoOccurance, N'28 Days,') t
ORDER BY similarity DESC
```

Транзитивные ассоциативные сети

```
-- заполняем таблицу транзакций
DECLARE @CoOccurance AS input.CoOccuranceTable;
INSERT INTO @CoOccurance
(
    TransactionId,
    ItemId
)
SELECT
    CustomerID,
    Movie
FROM dbo.Movies

SELECT *
FROM cooccurance.predictItemsFromStringWithoutTrainin(1, @CoOccurance, N'28 Days|') t
ORDER BY similarity DESC
```

Results | Messages

ItemId	Similarity
Lord of the Rings: The Fellowship of the Ring, The	0,02
Star Wars	0,02
Star Wars Episode I: The Phantom Menace	0,02
Star Wars Episode II: Attack of the Clones	0,02
Star Wars Episode V: Empire Strikes Back	0,02
Star Wars Episode VI: Return of the Jedi	0,02
Sting, The	0,01
Talented Mr. Ripley, The	0,01
Tank Girl	0,01
Taxi Driver	0,01

Транзитивные ассоциативные сети

```
-- заполняем таблицу транзакций
DECLARE @CoOccurance AS input.CoOccuranceTable;
INSERT INTO @CoOccurance
(
    TransactionId,
    ItemId
)
SELECT
    CustomerID,
    Movie
FROM dbo.Movies

SELECT *
FROM cooccurance.predictItemsFromStringWithoutTraining 2 @CoOccurance, N'28 Days|') t
ORDER BY similarity DESC
```

00 %

Results | Messages

	ItemId	Similarity
1	Star Wars	0,389999999999976
2	Star Wars Episode V: Empire Strikes Back	0,279299999999986
3	Matrix, The	0,270399999999986
4	Star Wars Episode VI: Return of the Jedi	0,264899999999988
5	Lord of the Rings: The Fellowship of the Ring, The	0,255299999999989
6	Indiana Jones and the Raiders of the Lost Ark	0,24869999999999
7	Star Wars Episode II: Attack of the Clones	0,239399999999991
8	Star Wars Episode I: The Phantom Menace	0,221899999999993
9	Gladiator	0,205099999999993
10	Indiana Jones and the Last Crusade	0,203599999999995
11	Apollo 13	0,188599999999996

Collaboration Filtering

User-based Collaboration Filtering

Каждому пользователю ставится в соответствие вектор покупок, где каждому товару приписывается его рейтинг. В качестве меры близости между пользователями применяются метрики, основанные на углах между векторами покупок.

- Косинус угла между векторами покупок.
- Корреляция между векторами покупок.

Рейтинг товара данного пользователя определяется как средний рейтинг этого товара, проставленный другими пользователями с учетом веса, определяемого мерой близости к данному пользователю.

User-based Collaboration Filtering

Фильтрация по пользователям имеет высокую точность. Однако, недостатком всех вариантов приведенного алгоритма является его ресурсоемкость (требование к памяти) и сложность (количество вычислений, требуемое для получения рекомендаций).

К тому же вычисление степени близости анализируемой транзакции ко всем остальным транзакциям может производиться **только в реальном времени**, так как данные о текущей транзакции становятся доступными только в момент выработки рекомендаций

Вывод: фильтрация по пользователям может применяться только к относительно небольшим базам данных.

Item-based Collaboration Filtering

Каждому товару ставится в соответствие вектор клиентов, где каждому клиенту приписывается рейтинг, который он выставил данному товару. В качестве меры близости между товарами применяются метрики, основанные на углах между векторами клиентов.

Идея в фильтрации по товарам состоит в выставлении неизвестного рейтинга товару в анализируемой транзакции на основании взвешенных рейтингов других товаров, входящих в эту транзакцию. Т.е. мы считаем, что рейтинг товара будет близок к рейтингам похожих на него товаров из уже оцененных.

Item-based Collaboration Filtering

В алгоритме фильтрации по товарам степень близости анализируемого товара ко всем остальным товарам может быть вычислена в отложенном режиме по расписанию, так как вектора рейтингов всех товаров доступны до момента формирования рекомендации.

Мы можем разделить процесс выработки рекомендаций на отложенную стадию (вычисление степени близости товаров друг к другу) и стадию в реальном времени (вычисление рейтингов товаров).

Таким образом, алгоритм фильтрации по товарам оказывается более эффективным с точки зрения времени формирования рекомендаций, чем алгоритм фильтрации по транзакциям благодаря возможности проведения отложенной предобработки данных.

Рекомендательные системы: Другие алгоритмы

Другие алгоритмы

- **Компрессия и понижение размерности** матрицы связи между клиентами и товарами при помощи SVD.
- **Метод персональной диагностики**. Мы рассматриваем анализируемую транзакцию как будто она была выбрана случайно с равномерным распределением из генеральной совокупности транзакций с добавлением белого шума. Т.е. помимо наблюдаемой случайной величины, соответствующей рейтингам товаров в анализируемой транзакции, у нас есть еще ненаблюдаемая случайная величина , которая описывает вариантом какой транзакции является «на самом деле».
- Классификация: **Naïve Bayes**.

Другие алгоритмы

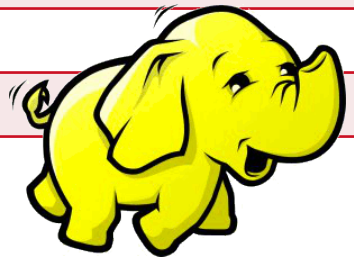
- **Гибкая смешанная модель (Flexible Mixture Model).** Идея этого метода основывается заключается в следующем:
  - Существует отдельный набор групп пользователей и отдельный набор групп товаров, вхождение в которые определяют значение рейтинга. Т.е. пользователи из группы i оценивают товары из группы j примерно одинаково.
  - Пользователь или товар могут входить в несколько групп. Т.е. фильм может быть историческим, военным и психологическим одновременно.
- **Аспектная модель (Aspect Model)**. Множество данных состоит из наблюдений, представляющих собой пары совместно встречающихся транзакций и товаров, интерпретируемые как «в i-ую транзакцию входит j-ый товар». С каждым наблюдением также связана ненаблюдаемая переменная класса, описывающая группы транзакций и товаров, «встречающихся» совместно.

Big Data, Hadoop, Mahout etc

Big Data

Large Complex Unstructured

	Traditional RDBMS	MapReduce
Data Size	Gigabytes *(Terabytes)*	Petabytes *(Hexabytes)*
Access	Interactive and Batch	Batch
Updates	Read / Write many times	Write once, Read many times
Structure	Static Schema	Dynamic Schema
Integrity	High (ACID)	Low
Scaling	Nonlinear	Linear
DBA Ratio	*1:40*	*1:3000*



Reference: Tom White's Hadoop: The Definitive Guide

Mahout

Apache Mahout™ - масштабируемая библиотека машинного обучения для Hadoop, предназначенная для обработки больших объемов данных.

- **Collaborative Filtering**
- **User and Item based recommenders**
- K-Means, Fuzzy K-Means clustering
- Mean Shift clustering
- Dirichlet process clustering
- Latent Dirichlet Allocation
- **Singular value decomposition**
- Parallel Frequent Pattern mining
- Complementary Naive Bayes classifier
- Random forest decision tree based classifier

Hadoop: ссылки

- http://hadoop.apache.org/
- http://mahout.apache.org/
- https://www.hadooponazure.com/
- Running Apache Mahout at Hadoop on Windows Azure: http://blogs.msdn.com/b/avkashchauhan/archive/2012/03/06/running-apache-mahout-at-hadoop-on-windows-azure-www-hadooponazure-com.aspx

Литература

- J. Herlocker, J. Konstan, L. Terveen, and J. Riedl.: “Evaluating collaborative filtering recommender systems”, ACM Translations on Information Systems, Vol. 22(1), 2004.
- Jun Wang, Arjen P. de Vries, Marcel J.T. Reinders, “Unifying Userbased and Itembased Collaborative Filtering Approaches by Similarity Fusion”.
- Justin Basilico, Thomas Hofmann “Unifying Collaborative and Content-Based Filtering”.
- Manolis G. Vozalis, Konstantinos G. Margaritis “Applying SVD on Generalized Item-based Filtering”.
- David M. Pennock, Eric Horvitz, Steve Lawrence and C. Lee Giles “Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach”.
- Prem Melville, Raymond J. Mooney, Ramadass Nagarajan “Content-Boosted Collaborative Filtering for Improved Recommendations”.
- Anne Yun-An Chen, Dennis McLeod “Collaborative Filtering for Information Recommendation Systems”.
- Luo Si, Rong Jin “Flexible Mixture Model for Collaborative Filtering”.
- Lyle H. Ungar, Dean P. Foster “A Formal Statistical Approach to Collaborative Filtering”.
- Rong Jin, Joyce Y. Chai, Luo Si “An Automatic Weighting Scheme for Collaborative Filtering”.
- Thomas Hofmann, Jan Puzieha “Latent Class Models for Collaborative Filtering”.
- Thomas George, Srujana Merugu “A Scalable Collaborative Filtering Framework based on Co-clustering”.
- A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. Modha “A generalized maximum entropy approach to Bregman co-clustering and matrix approximation”.
- Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu1, Zheng Chen “Scalable Collaborative Filtering Using Cluster-based Smoothing”.
- Zan Huang, Hsinchun Chen, Daniel Zeng “Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering”.
- Kai Yu, Anton Schwaighofer, Volker Tresp, Wei-Ying Ma, HongJiang Zhang “Collaborative Ensemble Learning: Combining Collaborative and Content-Based Information Filtering via Hierarchical Bayes”.
- Dmitry Y. Pavlov, David M. Pennock “A Maximum Entropy Approach To Collaborative Filtering in Dynamic, Sparse, High-Dimensional Domains”.
- Michael Leben “Applying Item-based and User-based collaborative filtering on the Netflix data”.
- Benjamin Marlin “Collaborative Filtering: A Machine Learning Perspective”.
- А.А. Барсегян, М.С. Куприянов, И.И. Холод, М.Д. Тесс “Анализ данных и процессов”, СПб.: БХВ-Петербург, 2009.