Introduction to a Recommender System for Apache OFBiz™

Anahita Goljahani, PhD
Information Engineering
Apache OFBiz™ is an open source product for the automation of enterprise processes that includes framework components and business applications for ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), E-Business / E-Commerce, SCM (Supply Chain Management), MRP (Manufacturing Resource Planning), MMS/EAM (Maintenance Management System/Enterprise Asset Management), POS (Point Of Sale).”

– Currently OFBiz doesn’t include any service that generates Recommendations for online shoppers.
Outline

• **Introduction**
  – Recommendations
  – Product Association Recommendations

• **A Product Association Recommender for OFBiz**
  – Out of the box support for Product Association
  – Recommender based approach to Product Association
  – Prototype of a Product Association Recommender
  – Demo

• **From the prototype to a production-ready component**
  – Summary of the main prototype features
  – Improvements and automation strategies
Introduction
Introduction
Recommendations (1/2)

• Context

  – **System** (e-commerce site, web-radio, etc.) that **provides a service** (selling, streaming, etc.) **on some products** (books, music, films, etc.)
  – **Set of users’ that interact with the system** (browse, rate, click, etc.) and **utilize the service** (buy a book, listen to a song, etc.)

• Recommendation

  – Information given to the users to support their choices (suggestions about products to buy, etc.)

  – Based on
    – Users’ preferences (preferred literary genre, preferred music genre, etc.)
    – Historical data (about past purchases, products’ ratings, etc.)
Examples of Recommendations

- **Personalized**: takes into account historical data or specific preferences of the user who is receiving the recommendation
  - Collaborative filtering (User-user, Item-item, Dimensionality reduction)
  - Content-based filtering
- **Non personalized**: does not take into account historical data or preferences of the user who is receiving the recommendation
  - Summary statistics (aggregated opinion recommenders)
  - Non personalized Product Association Recommendation (*Ephemerally personalized*: takes into account the current activity of the user, e.g., the product that is being checked out or is being viewed, etc.)
Introduction

Non personalized Product Association Recommender (1/4)

• Key concept
  – People who did X (bought some product, listened to a song, etc.) also did Y

People who bought X...

... also bought Y

• Key question
  • Given X, what are the products Y whose purchase is related (‘associated’) to the purchase of X?
Non personalized Product Association Recommender (2/4)

• Naïve algorithm
  – $X$: product for which we want Product Association Recommendations
  – Recommend $Y$ if most of the people who bought $X$ also bought $Y$
    - **Maximization of $P[Y|X]$** (read: probability of $Y$ given $X$)
  – **Drawback**: recommend popular products that are not related to $X$
  – **Example**
    - $X$: ice-cream
    - $Y$: popular barbecue sauce
    - Most of the people who bought the ice-cream also bought the barbecue sauce $\Rightarrow$ **high $P[Y\mid X]$** $\Rightarrow$ **recommend the barbecue sauce**
Introduction

Non personalized Product Association Recommender (3/4)

• Proper algorithm

  – X: product for which we want Product Association Recommendations

  – Recommend Y if the purchase of X increases the probability of buying Y

    - $P[Y|X] > P[Y|\sim X]$ (read: probability of Y given not X)
Introduction

Non personalized Product Association Recommender (4/4)

• Proper approach
  – Example
    - X: ice-cream
    - Y₁: popular barbecue sauce; Y₂: caramel topping for ice-creams
    - Y₁: most of the people who bought the ice-cream also bought the barbecue sauce → **high** \( P[Y_1 | X] \), but also most of the people who have never bought an ice-cream bought barbecue sauce → **same high** \( P[Y_1 | \neg X] \)
      - \( P[Y_1 | X]=P[Y_1 | \neg X] \) → \( Y_1 \) **NOT** recommended
    - Y₂: most of the people who bought the ice-cream also bought the caramel topping → **high** \( P[Y_2 | X] \), but very few people among those who have never bought an ice-cream bought a caramel topping → **low** \( P[Y_2 | \neg X] \)
      - \( P[Y_2 | X]> P[Y_2 | \neg X] \) → \( Y_2 \) **recommended!**
A Product Association Recommender for Apache OFBiz
A Product Association Recommender for OFBiz
Out of the box support for Product Association (1/2)

- Demo ecommerce site (http://localhost:8080/ecommerce/)
- Product detail page
A Product Association Recommender for OFBiz
Out of the box support for Product Association (2/2)

– Manual insertion of a record in the ProductAssoc entity with productAssocTypeId set to “ALSO_BOUGHT”

<entity-engine-xml>

<ProductAssoc
  productId="GZ-1000"
  productIdTo="WG-1111"
  productAssocTypeId="ALSO_BOUGHT"
  fromDate="2014-09-02 12:00:00.000"/>

</entity-engine-xml>

• Drawbacks
  – Requires manual maintenance
  – Not based on any Product Association recommender algorithm
A Product Association Recommender for OFBiz

Recommender based approach

- **Procedure**

  1) Generation of associations based on Product Association algorithm
     - X: product for which we want Recommendations
     - recommend Y if the purchase of X increases the probability of buying Y
       \[ P[Y|X] > P[Y|\sim X] \]

  2) Storage of associations in ProductAssoc entity

- **Implementation of Product Association algorithm**

  - Y: items bought together with X in the same orders
  
  - **Computation of** \( P[Y|X] \) **and** \( P[Y|\sim X] \)
    
    \[
    \hat{P}[Y | X] = \frac{\text{(# of orders containing Y and X)}}{\text{(# of orders containing X)}}
    \]
    
    \[
    \hat{P}[Y|\sim X] = \frac{\text{(# of orders containing Y but not X)}}{\text{(# of orders not containing X)}}
    \]

  - **Assessment of** \( P[Y|X] > P[Y|\sim X] \)
    
    \[
    \hat{P}[Y | X] > \hat{P}[Y|\sim X] \text{ and}
    \]

    - Log-likelihood ratio hypothesis test
      
      - Null hypothesis \( P[Y|X] \) equal to \( P[Y|\sim X] \)
A Product Association Recommender for OFBiz
Prototype of a Product Association Recommender (1/2)

• “Recommender” component  (https://cwiki.apache.org/confluence/x/OYPfAg)
  – getRecommendedAssociatedProducts Java service
    - Input
      - product ID (X) for which we want to provide Recommendations
      - N max number of recommended products to return
    - Output
      - list of recommended products for X
    - Main steps:
      1) Finds the products ordered together (items in the same sales orders) with X
      2) For each ‘co-ordered’ product Y, computes the conditional probabilities and collect the products that pass the test
      3) Returns the first N (highest scores) recommended products and stores them in ProductAssoc entity (productAssocTypeId = “ALSO_BOUGHT”)
A Product Association Recommender for OFBiz
Prototype of a Product Association Recommender (2/2)

- `createTestDataForProductAssociationRecommender` Java service: creates sales orders that simulate purchases with known conditional probabilities

  - **Input**
    - N number of orders

  - **Products**
    - LED TV (X)
    - USB PEN DRIVE (Y₁) \( P[Y₁ | X] = P[Y₁ | ~X] \)
    - TV AUDIO SYSTEM (Y₂) \( P[Y₂ | X] > P[Y₂ | ~X] \)

  - **Conditional probabilities**
    - \( P[\text{LED TV}] = 0.5 \)
    - \( P[\text{TV AUDIO SYSTEM} | \text{LED TV}] = 0.7; P[\text{TV AUDIO SYSTEM} | ~\text{LED TV}] = 0.1 \)
    - \( P[\text{USB PEN DRIVE} | \text{LED TV}] = 0.9; P[\text{USB PEN DRIVE} | ~\text{LED TV}] = 0.9 \)
A Product Association Recommender for OFBiz Demo (1/2)

- Install the “Recommender” component under the hot-deploy folder
- Load demo products (ant load-demo)
- Run OFBiz (%OFBiz_HOME\tools\startofbiz.bat or %OFBiz_HOME\tools\startofbiz.sh)
- Web Tools (http://localhost:8080/webtools/control/main) -> Run Service
A Product Association Recommender for OFBiz Demo (2/2)

• Demo ecommerce site (http://localhost:8080/ecommerce/)

• Led TV detail page
From the prototype to a production-ready component
From the prototype to a production-ready component
Summary of the prototype main features

– ‘Recommender’ hot-deploy component for the generation of Product Association Recommendations (downloadable from https://cwiki.apache.org/confluence/x/OYPfAg)

– Services
  - getRecommendedAssociatedProducts
  - createTestDataForProductAssociationRecommender
  - removeRecommenderTestData

– Products ‘bought together’ ↔ in the same order

– Product Association algorithm based on computation of conditional probabilities and log-likelihood statistical hypothesis testing
From the prototype to a production-ready component
Improvements and automation strategies

– Introduction of a configurable time-window for considering products as “bought together”

– Bulk processing (all products, categories, etc.)

– Load-testing (on hundreds of products and thousands of orders) and performance improvements

– Automation strategies
  - service triggered, e.g., by the creation of an order (e.g., ECA)
  - scheduled job (every night, week, etc.)

– Integration of business rules

– Community feedback and contributions are very welcome!
References

- OFBiz website http://ofbiz.apache.org/

- Product Association Recommender for OFBiz:
  - https://cwiki.apache.org/confluence/x/OYPfAg

- Recommendations

- Log-likelihood ratio hypothesis testing
  - http://tdunning.blogspot.it/2008/03/surprise-and-coincidence.html

About me

- short bio
  http://apacheconeu2014.sched.org/speaker/anahita.goljahani#.VFosvMnYr04
- contact: anahita.goljahani@gmail.com