# Análise de Similaridade no Sistema HiperOriginal

## I. Estrutura do Sistema HiperOriginal

O sistema HiperOriginal (HO) mantém três tipos de informação úteis para efeito de cálculo de similaridade: usuários, produtos e ratings.

*Usuários* representam as pessoas que têm interesse em produtos.

*Produtos* são descritos por tags e agrupados em categorias previamente definidas no sistema. Um produto pode ter zero ou mais tags e uma tag pode ser utilizada por um ou mais produtos. O conjunto de categorias tende a ser muito menor do que o conjunto de tags e permite a navegação em árvores de produtos. Um produto pode estar em mais de uma categoria e uma categoria pode ter zero ou mais produtos. As categorias e tags associadas a um produto são escolhidas pelo usuário no momento do cadastramento.

*Ratings* indicam a percepção dos usuários sobre os produtos. Um rating sempre relaciona um usuário a um produto, indicando se é uma avaliação positiva ou negativa.

## II. Cálculo de Similaridade

Para calcular a similaridade entre dois elementos eles devem ser descritos por um mesmo conjunto de propriedades e, opcionalmente, classificados em grupos (com ou sem interseções entre os grupos). Por exemplo, os produtos do HO são associados a tags (propriedades que os descrevem) e classificados em categorias (com interseções entre as categorias).

Se as propriedades forem binárias (ou seja, cada elemento possui ou não possui cada uma das propriedades), podemos calcular a distância entre dois elementos pela função *Jaccard*. Este é o caso dos produtos do HO: considerando que cada tag (do conjunto de tags de todos os produtos do sistema) é uma propriedade, cada produto pode ter ou não uma propriedade. *Jaccard* calcula o número de propriedades compartilhadas por dois elementos P' e P'' e divide pelo número de propriedades em pelo menos um elemento. Esta função é representada pela equação abaixo e varia entre 0.0 (elementos idênticos) e 1.0 (elementos muito diferentes).

Se é possível definir a intensidade com que um elemento possui uma propriedade, podemos calcular a distância *Euclidiana* entre dois elementos P' e P'' conforme a equação abaixo. Nesta equação consideramos a existência de *m* propriedades, cada qual mensurada como Pi no intervalo [0, 1]. Como na função *Jaccard*, esta função varia entre 0.0 (produtos idênticos) e 1.0 (produtos muito diferentes).

Dada uma lista de elementos (produtos, por exemplo), podemos calcular os elementos mais próximos a cada elemento P’ pertencente à lista. Como este cálculo tem complexidade *O(n2)*, onde *n* é o número de elementos, uma alternativa é realizar este cálculo apenas dentro dos grupos em que os elementos são classificados. Assim, ao invés de comparar cada elemento com todos os outros elementos disponíveis, faríamos a comparação apenas com elementos dos mesmos grupos.

\*\*\*\* calcular a diferença das duas análises no HO !!!

## III. Oportunidades para Cálculo de Similaridade no HO

Identificamos que os seguintes tipos de cálculo de similaridade podem ser realizados com base nos dados do HO:

* *Similaridade entre produtos*: com base nas informações disponíveis sobre cada produto podemos calcular a similaridade entre eles. Um uso prático desta similaridade é a indicação de produtos para usuários de acordo com os produtos que eles gostaram. Dado um produto, poderíamos calcular os N produtos mais próximos e apresentar para o usuário como sendo do seu interesse;
  + Dados dois produtos P' e P'', podemos usar a equação *Jaccard* para calcular a distância entre eles com base em suas tags. Dado um produto P e a distância entre P e cada produto do sistema, podemos gerar uma lista de produtos próximos a P;
  + ...
* *Similaridade entre usuários*: um dos objetivos do HO é formar uma rede social de usuários com interesses em seus produtos. Como o sistema mantém um histórico dos produtos que um usuário gostou ou não gostou, podemos usar esta informação para indicar potenciais amigos na rede social. Existem diversas alternativas para calcular esta similaridade:
  + Dado um usuário U, podemos calcular o conjunto de *tags* deste usuário como a união das tags dos produtos que ele gostou. Dados dois usuários U' e U'', podemos usar a equação *Jaccard* para calcular a distância entre eles. Dado um usuário U e a distância entre U e cada usuário do sistema, podemos gerar uma lista de usuários próximos a U, indicando estes usuários como potenciais amigos de U;
  + Diferente dos produtos (que têm uma relação binária com as tags), usuários podem ter uma relação de intensidade com as tags. Por exemplo, se uma tag pertencer a dois produtos que o usuário tenha gostado, ela pode receber peso 2. Por outro lado, se a tag pertencer a um produto que o usuário não tenha gostado, ela pode receber peso -1. Neste caso, podemos usar a distância *Euclidiana* para calcular a distância entre dois usuários.

## III. Similaridade entre dois produtos

Dado um produto, queremos identificar os *N* produtos mais próximos a ele. Podemos considerar N como infinito, mas não faz sentido ter uma lista muito grande de produtos próximos e isto aumentaria o volume de memória necessário para processamento nas consultas realizadas por cada cliente. Considerando este aspecto, trataremos *N=10*.

Para calcular a distância entre dois produtos, P1 e P2, levaremos em consideração as tags destes produtos. Consideraremos somente produtos que possuam título e pelo menos duas tags. Com relação à primeira restrição, identificamos que alguns produtos enviados pelo HO não possuíam título indicado no campo *lnTitle*. Estes produtos foram descartados por acreditarmos se tratar de testes inseridos no banco de dados.

Com relação à segunda restrição, precisamos de pelo menos duas tags para realizar uma classificação crível: produtos com apenas uma tag poderão se parecer com centenas de produtos, tornando impossível diferenciar entre eles. Sugerimos que a HO realize testes mais detalhados visando avaliar se esta restrição deve ser mais severa (por exemplo, calcular a distância apenas para produtos com mais de quatro tags).

PELO MENOS UMA CATEGORIA ...

Com base nestas restrições, calculamos a distância entre dois produtos *P'* e *P''* usando a função Jaccard. Esta função calcula o número de tags compartilhadas pelos dois produtos e divide pelo número de tags únicas nos dois produtos. Esta função pode ser representada pela equação abaixo e varia entre 0.0 (produtos idênticos) e 1.0 (produtos muito diferentes).

Usamos um limite mínimo de 0.5 para montar a lista de produtos similares

Inicialmente, montamos uma lista de produtos e seus similares. Para cada produto, calculamos a distância com cada outro produto do sistema

## III. Hadoop

Classificação por categoria simplifica a parada! Um arquivo por categoria ...

Usuário deu like em X produtos

seleciona os X.10 mais próximos

remove duplicatas, pegando a distância mínima entre as duplicatas de um produto

remove produtos unliked

ordena por distância e apresenta