# 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 (as categorias de produtos, no caso do HO). Assim, ao invés de comparar cada elemento com todos os outros elementos disponíveis, faríamos a comparação apenas com elementos dos mesmos grupos.

O uso de categorias também tornaria possível paralelizar o cálculo de similaridade, pois o conjunto de produtos poderia ser dividido por categoria e submetido a um processo de Map-Reduce (Hadoop).

## 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. Guardar apenas os 10 produtos mais próximos a P, junto com a sua distância a P;
  + Dado um usuário U, identificar todos os produtos que ele gostou. Se o histórico do usuário for muito extenso, capturar apenas as avaliações positivas mais recentes deste usuário;
  + Com base nas distâncias entre os produtos, selecionar os 10 mais próximos a cada produto que o usuário gostou, indicando a intensidade da recomendação. Quanto menor a distância de um produto P a um produto que o usuário tenha gostado, maior é a intensidade de recomendação do produto P;
  + Com base na lista de produtos recomendados, remover aqueles para os quais o usuário tenha feito uma avaliação negativa. Em seguida, remover duplicatas, assumindo o maior índice de recomendação entre os produtos duplicados;
  + Por fim, ordena os produtos pelo índice de recomendação e apresenta para o usuário.
* *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.

## IV. Similaridade entre Produtos

No contexto deste projeto, vamos implementar a similaridade entre produtos, conforme especificado no primeiro item da seção III.

Consideraremos somente produtos que possuam título, pelo menos duas tags e pelo menos uma categoria. 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, produtos com apenas uma tag poderão se parecer com centenas de produtos (todos os que possuem a mesma tag), tornando impossível diferenciar entre eles. Para o futuro, sugerimos que a HO realize testes para avaliar se esta restrição deve ser mais severa (por exemplo, considerar um mínimo de quatro tags).

Com relação à terceira restrição, assumimos que produtos não classificados são testes registrados no banco de dados e não representam produtos reais. Por conta disso, não consideramos estes produtos em nossas análises.

Por fim, consideraremos um limite máximo de 0.5 na função *Jaccard* para montar a lista de produtos similares, que será composta por (até) 10 produtos mais próximos a cada produto P.

## V. Paralelizando o Cálculo de Similaridade

Fizemos um teste na base de produtos do HO e observamos que apenas 6% dos produtos próximos identificados para todos os produtos da base são alterados quando consideramos comparações apenas dentro das categorias dos produtos.

Dado o ganho de tempo obtido quando consideramos as categorias (vindo da redução no número de comparações e da capacidade de paralelizar o cálculo), recomendamos o uso de categorias, especialmente quando o número de produtos aumentar consideravelmente.

Próximos passos: Classificação por categoria simplifica a parada! Um arquivo de entrada para o Hadoop por categoria ... implementar o cálculo em Hadoop.