

Determinação do peso das arestas de um grafo

Uma aplicação prática

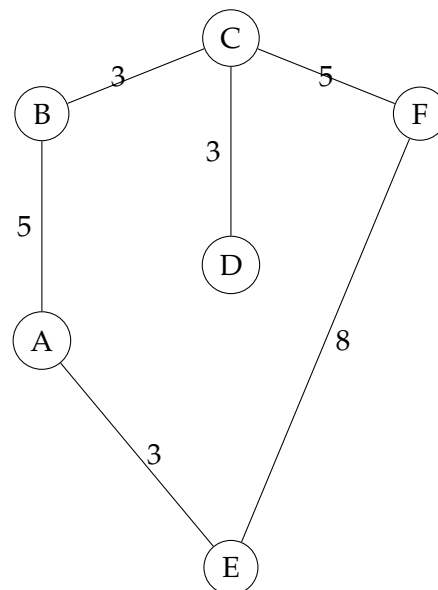
FELIPE AMORIM
Universidade Federal do Paraná
12 de março de 2019

O Contexto e Exemplo Prático

Inicialmente, o problema consiste em uma aplicação cujo o objetivo é formar relações entre pessoas, de forma que elas conheçam outras pessoas através de uma dinâmica de perguntas e respostas, possibilitando a formação de amigos ou até de casais. Para isso a aplicação inicialmente aplica um questionário simples ao usuário de forma a obter alguma informação sobre ele. Essas perguntas são do tipo:

- Você prefere verão ou inverno?
- No geral você prefere sair e se divertir com amigos ou ficar em casa vendo séries, jogando vídeo game, etc?
- Você gosta mais de gatos ou de cachorros?

Assim, quando o usuário A terminar de responder o questionário, o sistema irá comparar as respostas desse usuário com a de outros usuários da rede e tentará classificá-lo de acordo com a similaridade entre eles. Imaginamos até o momento que uma boa abstração e estrutura de dados para representar essa situação seja um grafo não direcionado com um peso k nas arestas que representa o grau de similaridade entre dois usuários (quanto maior o k maior é a similaridade entre as respostas do questionário entre dois



usuários) conforme a ilustração ao lado.

Na figura, o usuário *A* está ligado ao usuário *B* com uma similaridade igual a 5 e ligado a *E* com uma similaridade menor, igual a 3. O usuário *A* não aparece ligado a *D* aqui porque não houve nenhuma resposta comum entre *A* e *D* e portanto seu k é igual a zero, por isso não representamos uma aresta no grafo.

Nesse exemplo prático, essa seria a situação da rede de usuários no instante em que o usuário termina de responder o questionário. Porém, logo após concluir esse processo, o sistema apresenta um feed para o usuário *A* contendo uma série de perguntas que os outros usuários que estão ligados a ele estão fazendo. O nosso usuário do exemplo pode então selecionar uma das perguntas do seu feed e respondê-la. No momento em que ele responder, o usuário que criou aquela pergunta e a disponibilizou na rede poderá visualizar a resposta vinda de *A* através de uma outra tela na aplicação, ou um outro feed de respostas. O ponto é que, até esse momento nenhum usuário sabe a real identidade dos outros usuários na rede. Sendo assim, uma pessoa pode perguntar na rede quem prefere um estilo de filme ao invés de outro, quais são as opiniões delas sobre um determinado assunto e por aí vai. Conforme os outros usuários forem respondendo as questões, o criador poderá avaliar essas respostas e decidir se elas são adequadas ou não.

Uma nota importante sobre o feed de perguntas: num cenário de muitos usuários, ficaria impraticável que todos recebam perguntas em seus feeds o tempo todo e de muitas pessoas porque isso pode gerar muito tráfego na rede. Sendo assim, é possível priorizar as perguntas que aparecem no feed e dizer que elas, por exemplo, só serão enviadas a um determinado usuário se o k for suficientemente grande. Isso garante que o usuário receberá

perguntas de pessoas que potencialmente são mais parecidas com ele e diminui a quantidade de informação trafegada, tornando o feed um pouco mais seletivo.

Suponha agora que A recebeu em seu feed uma pergunta do usuário B (mas ele não sabe quem é B , apenas sabe que existe uma pergunta a ser respondida caso ele o queira). B quer saber, por exemplo, quem gosta da cor azul e essa é justamente a cor favorita de A . Então A responde dizendo que sim, gosta da cor azul. Nesse momento B visualiza a resposta de A , fica satisfeito e marca a resposta como adequada. Nesse instante o peso k da aresta entre A e B no grafo aumenta, digamos para 6. Assim, se considerarmos que o feed prioriza questões vindas de usuários onde o k é maior, agora há uma chance maior de A receber mais uma pergunta de B e vice versa. Conforme essa interação vai acontecendo, num ato contínuo, esperamos que num determinado momento A e B tenham um k suficientemente alto a ponto de considerarmos que eles deveriam se conhecer e formar uma amizade (nesse momento a aplicação pode avisar-los disso e abrir uma janela de chat onde os dois podem conversar livremente, trocar telefones e tudo mais).

O Cálculo de k

Ainda não temos definido exatamente como será o cálculo do peso da aresta do grafo de usuários, apenas que existem alguns fatores a serem considerados, como talvez o número de usuários na rede e certamente a quantidade de perguntas trocadas entre dois usuários. Pode-se assumir até o momento que teríamos algo parecido com a relação abaixo.

$$k(A, B) = 1 - \left| \frac{R_A}{P_B} - \frac{R_B}{P_A} \right|$$

Onde $k(A, B)$ representa o grau de similaridade entre A e B , P_A e R_A a quantidade de perguntas que A fez na rede e que chegaram até B e a quantidade de respostas adequadas que A forneceu para B , respectivamente. Da mesma forma, P_B representa a quantidade de perguntas que B enviou para A e R_B a quantidade de respostas que B forneceu e que A considerou como adequadas.

Então se supormos alguns valores para essa relação, como $R_A = 7$, $P_A = 10$, $R_B = 1$ e $P_B = 10$, teríamos:

$$k(A, B) = 1 - \left| \frac{7}{10} - \frac{1}{10} \right| = 0.4$$

Logo, poderíamos estabelecer um limiar, onde a partir de um k maior que 0.7, por exemplo, a relação entre A e B seria consolidada e surgiria a janela de chat para que eles possam conversar diretamente.

Considerações Finais

O cálculo de k é o trabalho em si. Uma vez que ele seja determinado o restante será simples porque apenas precisaríamos construir a aplicação.