DivDB: Incluindo Diversidade em Consultas por Similaridade*

Humberto L. Razente [1], Maria Camila N. Barioni[1], Marcos R. Vieira[2], Divesh Srivastava[3], Marios Hadjieleftheriou[3], Vassilis Tsotras[2], Caetano Traina Jr[4]

[1] Universidade Federal do ABC (UFABC)

[2] University of California at Riverside (UCR)

[3] AT&T Research Labs

[4] Universidade de São Paulo em São Carlos (USP)

{humberto.razente,camila.barioni}@ufabc.edu.br, {mvieira,tsotras}@cs.ucr.edu, {marioh,divesh}@research.att.com, caetano@icmc.usp.br

Resumo. *Nas últimas décadas houve um grande interesse no desenvolvimento de algoritmos para execução de consultas aos k-vizinhos mais próximos. No entanto, essas consultas não consideram o relacionamento entre os elementos do conjunto resposta. Embora a existência de elementos idênticos em grandes coleções seja rara, a existência de elementos muito similares é comum. Isso pode resultar em um grande esforço de navegação no conjunto resposta por parte do usuário, principalmente em consultas exploratórias. Uma alternativa é apresentar ao usuário um conjunto de elementos similares à consulta e também diversos entre si. Nessa demonstração será apresentado um sistema de busca que implementa uma série de algoritmos de diversificação de resultados em consultas por similaridade, permitindo ao usuário avaliar a diversidade de suas consultas.*

Abstract. *In the last decades there was considerable interest in optimizing nearest neighbor queries. However, these queries do not consider the relationship among the returned elements. While the existence of identical elements in large collections is rare, the existence of very similar is frequent, resulting in a great navigation effort by the user. A way to solve this issue is to present the user with a set of elements at the same time similar to the query and diverse among themselves. In this demonstration we will present a query system that implements a series of algorithms for result diversification, allowing the user to evaluate the diversity answers for her queries.*

1. Introdução

Nas últimas décadas inúmeros trabalhos de pesquisa focaram no desenvolvimento e otimização de métodos para processamento de consultas aos k-vizinhos mais próximos (k-NN). Dado um elemento de consulta s_q e um valor $k > 0$, $k \in \mathcal{N}$, uma consulta aos k-NN retorna k elementos do conjunto de dados que possuem as menores distâncias ao elemento de consulta s_q. Existe uma grande quantidade de trabalhos que buscam otimizar a eficiência computacional dessas consultas (por exemplo [Hjaltason and Samet 1999]). No entanto, uma questão muito importante está relacionada ao aumento da satisfação do

*Este trabalho recebeu auxílios da Fapesp (processos 2006/00336-5 e 2010/18101-0), *National Science Foundation* (NSF IIS grants 0705916, 0803410 and 0910859) e CAPES/Fulbright.

usuário com os resultados da consulta. Embora a existência de elementos idênticos em grandes coleções seja rara (por exemplo, em banco de imagens médicas), a existência de elementos muito similares é comum (por exemplo, várias imagens de tomografia do crânio de diferentes pessoas). Resultados que possuem elementos que trazem ao usuário informações "novas" (ou diversas), e também relevantes à consulta, podem ser mais interessantes em certos domínios de aplicação (por exemplo, em consultas exploratórias). Outra vantagem desta abordagem é facilitar a análise e exploração dos resultados por parte do usuário. Uma possível abordagem para esse problema é apresentar ao usuário um conjunto de elementos que sejam relevantes à consulta e, ao mesmo tempo, diversos entre si.

Existem várias aplicações que podem se beneficiar da diversidade no conjunto resposta. Por exemplo, na pesquisa pelo termo "*apple*" em uma máquina de busca, ao invés da consulta exploratória inicial retornar os primeiros 100 resultados relacionados apenas ao fabricante de computadores, seria interessante que ela retornasse alguns resultados referentes à computadores, outros referentes ao cantor "Apple Gabriel", outros referentes à cidade conhecida como "*big apple*", outros referentes à fruta, etc. Para permitir a realização desse tipo de consulta, é necessário incorporar ao cálculo dos vizinhos mais próximos um componente de diversidade. Portanto, é necessário definir uma consulta que recupere um conjunto de elementos próximos ao elemento de consulta e que também satisfaça a restrição de apresentar diversidade de características entre seus elementos.

O processamento da consulta aos k*-vizinhos diversos mais próximos* deve empregar uma função de ordenação que considere a similaridade ao elemento de referência e, ao mesmo tempo, a diversidade entre seus elementos – ambas computadas por funções de distância (não necessariamente idênticas). Os objetivos da demonstração descrita neste trabalho são: apresentar um sistema de consultas que implementa os principais algoritmos de diversidade em uma plataforma comum para o usuário avaliar os resultados e algoritmos; e fornecer uma interface gráfica onde o usuário possa alterar o valor de diversidade na consulta e, assim, avaliar seu impacto no resultado do conjunto resposta. Nesta demonstração serão apresentados os resultados com dois conjuntos de dados reais e diversas consultas.

2. Trabalhos correlatos

O tratamento da diversidade em resultados de consultas tem atraído o interesse em domínios de aplicação que retornam grandes conjuntos de respostas. Exemplos incluem buscas exploratórias e ambíguas por palavras-chave [Gollapudi and Sharma 2009, Agrawal et al. 2009, Rafiei et al. 2010] e a diversificação de bancos de dados estruturados [Demidova et al. 2010, Liu et al. 2009]. Em geral, o resultado é computado em duas fases: uma ordenação dos elementos quanto à similaridade à consulta gera um conjunto candidato S; e em seguida, um resultado $R \subseteq S$ é computado retornando os elementos mais diversos entre os relevantes baseado na diversidade definida pelo usuário.

Recentemente, várias técnicas foram propostas para a diversificação de resultados de consultas, sendo a maioria baseada na exploração de soluções gulosas (*greedy*) que constrõem os resultados de modo incremental [Carbonell and Goldstein 1998, Vee et al. 2008, Yu et al. 2009]. Isso se deve ao fato de o problema de retornar o "melhor" conjunto de elementos diversos ser *NP-hard* [Agrawal et al. 2009].

Os conjuntos de atributos para os quais a similaridade e diversidade são computados podem ter intersecção. Portanto, as técnicas existentes procuram encontrar uma relação (*tradeoff* λ) entre os componentes da similaridade e da diversidade. Nesse modelo, o resultado da consulta diversificada é um problema de otimização com dois critérios opostos, onde o objetivo é encontrar um conjunto resultado R, $|R| = k$, que *minimize* a função objetivo. Uma vantagem de ter o parâmetro *tradeoff* é que o usuário pode ajustar a relação entre similaridade e diversidade desejada no resultado final. Uma descrição mais detalhada desses algoritmos pode ser encontrada em [Vieira et al. 2011b] enquanto em [Vieira et al. 2011a] é apresentado uma demonstração que foca em uma interface estilo SQL na qual os usuários podem escrever suas próprias consultas.

3. Diversificação de Resultados: *k-vizinhos diversos mais próximos*

Uma consulta aos k-vizinhos diversos mais próximos deve encontrar um conjunto de k elementos que *minimize* sua distância ao centro de consulta s_q e que, ao mesmo tempo, *maximize* sua diversidade entre os elementos do conjunto resposta. A consulta aos *k-vizinhos diversos mais próximos* requer duas funções de distância: uma função δ_{sim} para avaliar a *similaridade* de cada elemento do conjunto de dados ao elemento s_q, e outra função δ_{div} para avaliar a *diversidade* entre os elementos da resposta. Embora seja possível empregar a mesma função em ambos os componentes similaridade e diversidade, definir a consulta baseada em duas funções a torna mais flexível e mais voltada a situações reais. Formalmente, essa consulta é definida sobre um domínio de dados $\mathbb{D}$ que é compartilhado por dois espaços métricos $\mathcal{M}_{sim} = \langle \mathbb{D}, \delta_{sim}() \rangle$ e $\mathcal{M}_{div} = \langle \mathbb{D}, \delta_{div}() \rangle$.

Dado um domínio $\mathbb{D}$ e um conjunto $D \subseteq \mathbb{D}$, o objetivo de uma consulta aos *k-vizinhos diversos mais próximos* é encontrar um conjunto $R \subseteq D$, $|R| = k$, que *minimize* suas distâncias δ_{sim} em relação a $s_q \in \mathbb{D}$ (Eq. 1), e também *maximize* sua diversidade (Eq. 2). Esta última é medida pela soma das distâncias δ_{div} entre todos os pares de elementos do conjunto resposta R. As funções de distância δ_{sim} e δ_{div} podem ser distintas e aplicadas a conjuntos distintos de atributos de cada elemento de um conjunto de dados. É necessário sejam definidas a medida de similaridade de um conjunto de elementos R a s_q e a medida de diversidade de R.

Similaridade: sendo $s_q \in \mathbb{D}$ um elemento de consulta e $R \subseteq D$ um conjunto de k elementos, a medida $similaridade : \mathbb{D} \times \mathbb{D}^k \rightarrow \mathbb{R}^+$ de R com relação à s_q é dada por:

$$similaridade(s_q, R) = \sum_{i=1}^{k} \delta_{sim}(s_q, R_i) \tag{1}$$

Diversidade: sendo $R \subseteq D$, a medida $diversidade : \mathbb{D}^k \rightarrow \mathbb{R}^+$ de R é dada por:

$$diversidade(R) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \delta_{div}(R_i, R_j) \tag{2}$$

As Eqs. 1 e 2 podem, respectivamente, ser normalizadas pelo número de elementos do conjunto resposta (k) e pelo número de distâncias entre todos os pares de elementos do conjunto resposta ($k * (k-1)/2$).

Resolver a consulta aos *k-vizinhos diversos mais próximos* corresponde a encontrar o conjunto de elementos que minimize a função $\mathcal{F}$ apresentada na Eq. 3. O objetivo

é encontrar um conjunto resultado diverso que "balanceie" a relação entre a minimização das similaridades em relação ao elemento de consulta s_q e a maximização da diversidade entre os elementos da resposta. O compromisso entre a diversidade e a similaridade é dado pelo peso da diversidade λ ($0 \leq \lambda \leq 1$), o que garante flexibilidade ao usuário para determinar a prioridade de similaridade ou da diversidade. A variável λ é um fator muito importante no sistema DivDB, dado que a diversidade na consulta depende de diversos fatores (i.e., o objeto de consulta, o conjunto de dados, a distribuição de distâncias, a expectativa do usuário, a semântica da consulta).

A consulta aos k-*vizinhos diversos mais próximos* com elemento de consulta s_q sobre um conjunto de dados $S \subseteq \mathbb{D}$ é definida como:

Consulta aos *k-vizinhos diversos mais próximos*: dado s_q e $D \subseteq \mathbb{D}$, a consulta aos k-*vizinhos diversos mais próximos* retorna um conjunto $R \subseteq D, |R| = k$ que minimiza a função $\mathcal{F} : \mathbb{D} \times \mathbb{D}^k \rightarrow \mathbb{R}^+$ com relação à s_q.

$$\mathcal{F}(s_q, R) = (1 - \lambda) \cdot similaridade(s_q, R) - \lambda \cdot diversidade(R) \tag{3}$$

Quando $\lambda = 0$, o problema equivale à consulta tradicional aos k-vizinhos mais próximos. Por outro lado, quando $\lambda = 1$, o problema equivale ao problema da diversidade máxima, que consiste em identificar um conjunto R com os k elementos mais diversos.

4. Demonstração do Sistema DivDB

Diversos algoritmos foram adaptados e implementados no sistema DivDB e são apresentados na Tabela 1. Dois novos métodos (*GMC* e *GNE*) estão também implementados no DivDB. Com base na heurística empregada, alguns algoritmos focam em tempo de execução, enquanto outros na qualidade dos resultados.

Tabela 1. Algoritmos de diversificação implementados no DivDB.

Abreviatura	Algoritmo	construção de R
Swap	*Swap* [Yu et al. 2009]	troca
BSwap	*BSwap* [Yu et al. 2009]	troca
MMR	*Maximal Marginal Relevance* [Carbonell and Goldstein 1998]	incremental
Motley	*Motley* [Jain et al. 2004]	incremental
MSD	*Max-Sum Dispersion* [Gollapudi and Sharma 2009]	incremental
CLT	*Clustering* [Leuken et al. 2009]	troca
GMC	*Greedy Marginal Contribution* [Vieira et al. 2011b]	incremental
GNE	*GRASP with Neighbor Expansion* [Vieira et al. 2011b]	meta-heurística

Nessa demonstração são empregados os conjuntos de dados DBLP[1] e um conjunto de imagens contendo 20k imagens de exames médicos. Considere o seguinte exemplo de consulta baseado no conjunto DBLP: a consulta *"nearest neighbor algorithm"*, com $k = 5$ e $\lambda = 0.25$ para a consulta aos k-*vizinhos diversos mais próximos* resulta nas tuplas apresentadas na Figura 1. O resultado contém tuplas que exploram a diversidade entre os elementos mais próximos, retornando as 5 primeiras publicações distintas mais próximas do elemento de consulta. Note que todos os resultados são muito similares à consulta (i.e., contém os termos da consulta), porém relacionados a diferentes domínios/aplicações.

[1] http://dblp.uni-trier.de/xml/

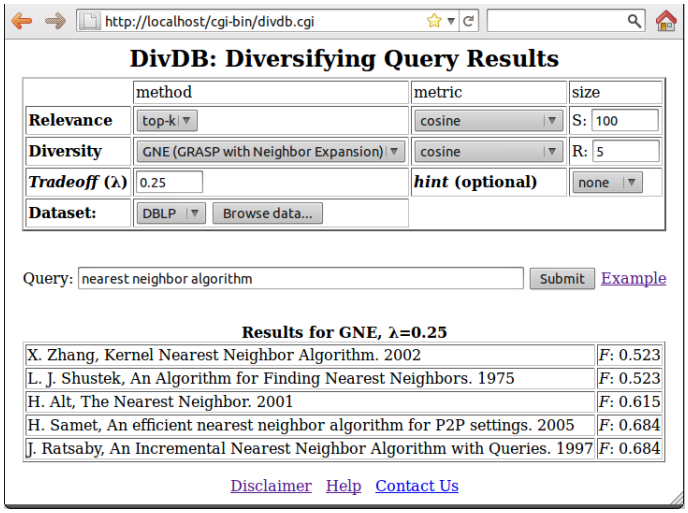

Figura 1. Interface do DivDB: consulta a "nearest neighbor algorithm".

Após a escolha do algoritmo e da execução da consulta, o DivDB permite que o usuário atualize o parâmetro de diversidade λ. Esta é uma característica importante do protótipo uma vez que o usuário pode aumentar ou diminuir a diversidade do resultado. A Figura 2 apresenta uma sequência de resultados para um conjunto de imagens de exames médicos onde foram selecionados 5 imagens por meio do algoritmo *Greedy Marginal Contribution* (GMC). Os parâmetros selecionados são apresentados em (a). No primeiro resultado (b), com $\lambda = 0$, o resultado contém os k-vizinhos mais próximos. Nas figuras seguintes (c)-(f), o usuário atribui valores de 0.2 (baixa diversidade) a 0.8 (alta diversidade) para λ. Como há muitas imagens muito similares entre si, o resultado com $\lambda = 0$ contém muitas imagens quase idênticas (possivelmente do mesmo exame/paciente). Ao aumentar o λ, imagens mais diversas, porém similares a s_q, são incluídas no resultado.

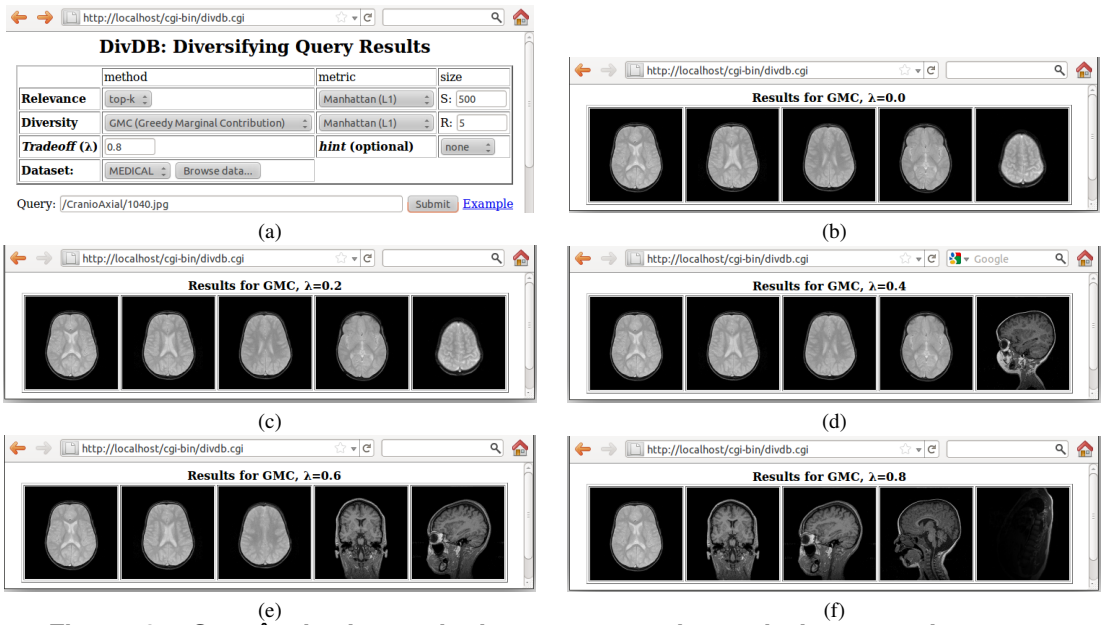

Figura 2. Sequência de resultados para o conjunto de imagens de exames médicos.

Além desses exemplos, serão apresentados resultados para diferentes algoritmos,

para avaliação da qualidade e tempo de processamento. Também serão apresentados resultados com outros conjuntos de dados.

5. Conclusões

O sistema DivDB foi implementado para prover uma plataforma para diversificação de resultados, permitindo a avaliação e validação de diversos algoritmos propostos na literatura. Adicionalmente, o DivDB permite a um analista de dados afinar suas consultas, pela seleção do algoritmo mais indicado para seu conjunto de dados bem como do valor de *tradeoff* (λ). Dentro do conhecimento dos autores, o DivDB é o primeiro protótipo a fornecer ao usuário a comparação de diferentes métodos de diversificação de resultados em uma plataforma comum. Por meio do sistema, o usuário pode fazer o "ajuste" fino dos parâmetros das consultas de modo iterativo, e com isso, fornecer resultados com diversidade dependendo da consulta e expectativa do usuário.

Referências

Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. (2009). Diversifying search results. In *ACM WSDM*, pages 5–14.

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *ACM SIGIR*, pages 335–336.

Demidova, E., Fankhauser, P., Zhou, X., and Nejdl, W. (2010). DivQ: Diversification for keyword search over structured databases. In *ACM SIGIR*, pages 331–338.

Gollapudi, S. and Sharma, A. (2009). An axiomatic approach for result diversification. In *WWW*, pages 381–390.

Hjaltason, G. R. and Samet, H. (1999). Distance browsing in spatial databases. *ACM Transactions on Database Systems (TODS)*, 24:265–318.

Jain, A., Sarda, P., and Haritsa, J. (2004). Providing diversity in k-nearest neighbor query results. In *PAKDD*.

Leuken, R., Garcia, L., Olivares, X., and Zwol, R. (2009). Visual diversification of image search results. In *Proc. WWW*.

Liu, Z., Sun, P., and Chen, Y. (2009). Structured search result differentiation. *PVLDB*, 2(1):313–324.

Rafiei, D., Bharat, K., and Shukla, A. (2010). Diversifying web search results. In *WWW*, pages 781–790.

Vee, E., Srivastava, U., Shanmugasundaram, J., Bhat, P., and Amer-Yahia, S. (2008). Efficient computation of diverse query results. In *IEEE ICDE*, pages 228–236.

Vieira, M. R., Razente, H. L., Barioni, M. C. N., Hadjieleftheriou, M., Srivastava, D., Traina-Jr, C., and Tsotras, V. (2011a). Divdb: A system for diversifying query results. In *Demo Track, VLDB*.

Vieira, M. R., Razente, H. L., Barioni, M. C. N., Hadjieleftheriou, M., Srivastava, D., Traina-Jr, C., and Tsotras, V. (2011b). On query result diversification. In *IEEE ICDE*.

Yu, C., Lakshmanan, L., and Amer-Yahia, S. (2009). It takes variety to make a world: diversification in recommender systems. In *EDBT*, pages 368–378.