Um Algoritmo Eficiente para Detecção de Exceções em Bases Reais de Alta Dimensionalidade

Carlos H. C. Teixeira[1], **Gustavo H. Orair**[1], **Wagner Meira Jr.**[1]

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos 6627 – Prédio do ICEx – sala 4010 - Pampulha
31.270-010 – Belo Horizonte – MG – Brasil

{carlos, orair, meira}@dcc.ufmg.br

Abstract. *The outlier detection problem has been a research topic with interesting applications in different domains, such as data cleaning and fraud detection. In this work, we propose an efficient and scalable distance-based algorithm for detecting outliers in large high dimensional databases. Our algorithm partitions the database and sorts the objects that are candidates to be an exception, reducing significantly the number of comparisons among objects. We evaluate the different sorting heuristics in a comprehensive set of real and synthetic databases. The results show that our algorithm outperforms by 52% the state of the art algorithm.*

Resumo. *A Mineração de Exceções tem sido uma área de pesquisa que possui interessantes aplicações em diferentes domínios, variando desde a limpeza de dados à detecção de fraudes. Neste trabalho, propomos um algoritmo eficiente e escalável baseado em distância para a mineração de exceções em grandes bases de dados de alta dimensionalidade. Nosso algoritmo realiza um particionamento dos dados e ordena os objetos candidatos a exceção reduzindo significativamente o número de comparações entre objetos. Avaliamos diferentes heurísticas de ordenação em um conjunto abrangente de bases de dados reais e sintéticas. Os resultados mostram que nosso algoritmo obtém um ganho de até 52% em relação ao estado da arte.*

1. Introdução

O rápido desenvolvimento das técnicas de armazenamento de dados em meio digital fez com que os tempos recentes fossem chamados de "Era da Informação". Sistemas poderosos de banco de dados para gerência e coleta de informações são usados em diversos campos, como finanças (empresas de grande e médio porte), astronomia, bioinformática, dentre outros. A cada dia, novas transações e registros são gerados e armazenados para futura análise e auxílio em tomada de decisões. Contudo, analisar essa grande quantidade de dados torna-se uma tarefa não-trivial, mesmo para um especialista da área. A *Mineração de Dados* é a área que busca extrair conhecimentos e encontrar informações úteis em um conjunto de dados de forma automática.

A maioria dos problemas da *Mineração de Dados* tem como objetivo extrair padrões existentes em um conjunto de dados. Com isso, a tendência é que objetos[1] raros,

[1]Registros de uma base dados

que não se enquadram nos padrões minerados, sejam ignorados ou até mesmo eliminados. Esses objetos raros, porém, podem conter informações muito úteis. A *Mineração de Exceções* se propõe a obter informações por meio da identificação e análise de objetos raros existentes em uma base de dados. Barnett e Lewis[Ord 1996] definem exceção como um exemplo (ou um sub-conjunto de exemplos) que é inconsistente com o restante do conjunto de dados.

A *Mineração de Exceções* é aplicada e usada em vários cenários como detecção de fraudes, análise de desempenho de atletas, combate à sonegação de impostos, detecção de invasões em uma rede, procura de nichos de mercado, auxílio na aprovação de crédito nas instituições financeiras, dentre várias outras.

Um dos grandes desafios da *Mineração de Dados* é a criação de técnicas e algoritmos eficientes que consigam lidar com o crescimento acentuado da quantidade de dados. No contexto de *Mineração de Exceções*, os algoritmos baseados em distância se destacam pela eficiência. Basicamente, eles representam os registros da base de dados através de pontos em um espaço multidimensional. Assim, a distância de um objeto aos pontos mais próximos será considerada como seu *valor de excepcionalidade*, e, portanto, os objetos raros são aqueles que estão mais distantes de seus vizinhos, segundo alguma métrica de distância. Recentemente foram propostas alternativas para a mineração de exceções baseadas em distância para grandes bases de dados de alta dimensionalidade. Essas técnicas buscam uma redução do número de cálculos de distância entre os pontos aplicando uma simples regra de "poda", também conhecida como *busca aproximada de vizinhos próximos*. Em outras palavras, a regra de "poda" permite que identifiquemos objetos comuns (não-exceções) sem a necessidade de compará-los com todos os demais registros da base de dados. O desempenho desses algoritmos está fortemente relacionado com a eficiência dessa regra de poda.

Estratégias simples para a mineração de exceções utilizam algoritmos de agrupamento[Ester et al. 1996, Zhang et al. 1996]. Nesses algoritmos consideram-se exceções os objetos que perturbam o processo de agrupamento seja porque não conseguem ser agregados a nenhum agrupamento, ou são os únicos objetos atribuídos a um agrupamento. A definição de exceções dada por esses algoritmos não é precisa matematicamente, mas possui um apelo intuitivo forte da excepcionalidade dos objetos. Nesse trabalho, primeiramente apresentamos um método para localização de objetos com um alto valor de excepcionalidade (esses objetos não são necessariamente as exceções da base de dados). Nossa primeira hipótese é que podemos encontrar "boas" exceções através de heurísticas de ordenação simples que utilizem as características dos agrupamentos dos objetos. Além disso, estudamos e avaliamos os compromissos da regra de "poda". Baseado nesse estudo, formulamos a segunda hipótese: podemos melhorar consideravelmente a eficiência da regra de "poda" e reduzir o número de comparações entre objetos, analisando primeiramente os objetos com um alto valor de excepcionalidade. Assim, propomos a *busca ordenada por exceções*.

Para validar nossas hipóteses, desenvolvemos e avaliamos, segundo métodos experimentais estatísticos, quatro heurísticas de ordenação. Os resultados mostram que podemos encontrar objetos com um alto valor de excepcionalidade utilizando heurísticas simples baseadas nas características dos agrupamentos e isso torna a regra de "poda" muito eficiente. Além disso, apresentamos um algoritmo para a mineração de exceções

Símbolo	Descrição
n	Número de exceções a serem detectadas em uma base de dados
k	Número de vizinhos mais próximos considerado
$D^k(p)$	Distância entre um ponto p e seu k-ésimo vizinho mais próximo
D^k_{min}	Distância entre a "pior" exceção encontrada e seu k-ésimo vizinho mais próximo
$\|P\|$	Número de objetos em uma partição P
$R(P)$	Valor da diagonal MBR de uma partição P

Tabela 1. Notações

em grandes bases de dados de alta dimensionalidade e mostramos que nossa abordagem é eficiente e escalável em relação ao número de objetos. Os experimentos demonstram que o algoritmo proposto obtém um ganho de até 52% em relação estado da arte, o algoritmo RBRP[Ghoting et al. 2005].

A proposta, elaboração e execução deste trabalho foram realizados pelo aluno de iniciação científica Carlos H. C. Teixeira, sob orientação do professor Wagner Meira Jr. e Co-orientação do aluno de mestrado Gustavo H. Orair.

O restante deste artigo está organizado como descrito a seguir. Primeiramente, examinamos os trabalhos relacionados, os algoritmos baseados em distânca existentes, na seção 2. Na seção 3, apresentamos nosso algoritmo para a mineração de exceções. Avaliamos os resultados na seção 4 e finalmente concluímos nosso trabalho na seção 5.

2. Trabalhos Relacionados

Nesta seção, discutimos os trabalhos relacionados, mais especificamente as técnicas de detecção de exceções baseadas em distância. As notações usadas nas próximas seções estão na Tabela 1.

O emprego de técnicas não-paramétricas para detecção de exceções foi primeiramente proposto por [Knorr and Ng 1999], onde considera-se como exceção um objeto que não possui "vizinhos" suficientes. Desde então, várias técnicas de definição de exceções não-paramétricas, como as técnicas baseadas em distância e as técnicas baseadas em densidade foram propostas. Enquanto as técnicas baseadas em densidade consideram a densidade local da vizinhança do objeto para a identificação das exceções, as técnicas baseadas em distância utilizam um conceito bem definido de distância para considerar uma exceção como um objeto que está afastado de seus "vizinhos". Dada uma medida de distância entre objetos (por exemplo, distância Euclidiana), algumas definições de exceções em bases de dados são:

- Exceções são pontos que possuem um número menor que k objetos vizinhos na base de dados a uma distância menor ou igual a d [Knorr and Ng 1999].
- Exceções são os n pontos que possuem os maiores valores de distância para seus respectivos k-ésimos vizinhos mais próximos[Ramaswamy et al. 2000, Bay and Schwabacher 2003, Ghoting et al. 2005].

Em [Ramaswamy et al. 2000], propõe-se que a distância ao *k-ésimo vizinho mais próximo* seja utilizada como uma medida da "excepcionalidade" dos objetos. A introdução desse conceito permitiu que se apresentasse uma classificação entre os objetos da base de dados considerando os valores de excepcionalidade. Dessa forma, objetos com alto valor de excepcionalidade seriam mais bem classificados. Neste contexto, a abordagem mais simples é um laço aninhado (algoritmo LA) que calcula as

distâncias entre todos objetos da base de dados para encontrar o k-ésimo vizinho mais próximo de cada objeto, o que resulta em complexidade quadrática – $O(N^2)$. Surgiram, então, várias estratégias para tornar mais eficiente a procura pelo k-ésimo vizinho mais próximo, desde propostas baseadas em índices espaciais por meio de estruturas, em geral árvores (como KD-trees[Bentley 1975], R*-trees[Ramaswamy et al. 2000]) até o particionamento do espaço em células uniformes. Porém, essas abordagens são afetadas pela *maldição da dimensionalidade* não escalando em relação ao número de dimensões [Knorr and Ng 1999, Ramaswamy et al. 2000].

Recentemente, surgiram trabalhos que propõem a mineração de exceções em bases de alta dimensionalidade. Bay e Schwabacher apresentaram o ORCA [Bay and Schwabacher 2003], um algoritmo baseado no algoritmo do LA combinado à *busca aproximada de vizinhos próximos* (definição 1). Além disso, para garantir que a base de dados não possua uma disposição dos objetos que leve à ineficiência da poda, os autores realizam um pré-processamento de ordenar aleatoreamente os objetos da base de dados. Note que, à medida que processamos os objetos da base de dados, o valor do limite de poda mínimo D^k_{min} cresce monotonicamente, obtendo assim uma poda mais eficiente.

Definição 1 *Se durante a busca pelos k vizinhos mais próximos de um ponto p, ou seja, durante a computação de $D^k(p)$ encontrarmos um valor inferior a D^k_{min}, podemos seguramente descartar o ponto p como uma exceção.*

Contudo, em [Ghoting et al. 2005] mostra-se que em bases de dados que possuem um número de exceções muito pequeno, a busca aproximada de vizinhos ainda é muito ineficiente, resultando em uma complexidade de $O(N^2)$. Foi proposto, então, o RBRP, um algoritmo de duas fases que otimiza a regra de "poda" através da *busca ordenada por vizinhos*. Na primeira fase, realiza-se um particionamento dos dados como fase de pré-processamento. Os agrupamentos obtidos são, então, utilizados para ordenar o espaço de busca por vizinhos próximos. Assim, a procura por vizinhos deve acontecer primeiramente no próprio agrupamento do objeto e posteriormente prosseguir a busca das partições mais próximas às mais distantes. A proposta fundamental do algoritmo RBRP é encontrar vizinhos próximos de maneira eficiente. De acordo com a definição 1, podemos dizer que o algoritmo RBRP tem como objetivo fazer com que o valor de $D^k(p)$ fique menor que D^k_{min} rapidamente, através da diminuição de $D^k(p)$. Dessa forma, consegue-se melhorar a eficiência da *busca aproximada de vizinhos próximos* obtendo um desempenho superior ao algoritmo ORCA.

3. O Algoritmo Proposto

Baseado no estudo dos compromissos da regra de "poda", podemos apontar duas formas de otimizarmos a *busca aproximada de vizinhos próximos*:

- Diminuir rapidamente o valor de $D^k(p)$, ou seja, encontrar vizinhos próximos eficientemente para um objeto p;
- Aumentar a taxa de elevação do valor de D^k_{min}, em outras palavras, encontrar exceções rapidamente.

A primeira alternativa foi estudada em [Ghoting et al. 2005], onde foi proposto o algoritmo estado da arte, RBRP e a *busca ordenada de vizinhos*. Nossa proposta de otimização da regra de "poda" baseia-se na segunda alternativa. Nossa hipótese é que

podemos encontrar objetos com um alto valor de excepcionalidade rapidamente através de heurísticas de ordenação simples que utilizem as características dos agrupamentos. Assim, elevaremos o valor do limite de poda mínimo D^k_{min} rapidamente tornando a *busca aproximada de vizinhos próximos* ainda mais eficiente. Nosso algoritmo possui três etapas principais: (1) Particionamento da base de dados, (2) Ordenação das partições e (3) Busca por exceções.

3.1. Particionamento da base de Dados

Em nosso trabalho, utilizaremos uma extensão do algoritmo de particionamento proposto em [Ghoting et al. 2005]. Nesse trabalho, Ghoting utiliza uma técnica de agrupamento hierárquico divisivo baseada no popular algoritmo K-médias[2] [Hartigan and Wong 1979]. É importante ressaltar que o algoritmo de agrupamento é utilizado apenas como fase de pré-processamento para que o algoritmo de busca de exceções consiga ser eficiente, assim, pode-se utilizar qualquer técnica de particionamento de dados.

3.2. Ordenação das partições

Apontamos como uma boa heurística aquela que seja capaz de abstrair, a partir das informações provindas das partições, quais são os objetos, de fato, raros. Propomos, então, as seguintes heurísticas de ordenação: (1) Aleatório, (2) Número de objetos nas partições, (3) Tamanho espacial ocupado pela partição e (4) Densidade da partição.

A primeira heurística, aleatória, será usada como linha de base. Na segunda heurística, usamos uma idéia aplicada pelos algoritmos de agrupamento de que objetos que estejam sozinhos em partições indicam uma intuição de excepcionalidade, e, assim, realizamos a ordenação das partições pelo número de objetos pertencentes às partições em ordem crescente. A terceira heurística ordena as partições considerando o tamanho espacial ocupado pelos agrupamentos. Partições maiores em termos de espaço ocupado serão consideradas primeiro, pois tendem a possuir uma área maior para disposição dos pontos. Com relação à métrica de densidade, acreditamos que em uma partição com baixa densidade os objetos estão mais afastados uns dos outros sendo fortes candidatos à exceção.

As heurísticas aleatória e número de objetos são de implementação direta. Para estimar o tamanho espacial das partições, na terceira heurística, utilizamos o valor da diagonal do MBR[Roussopoulos et al. 1995]. Na última heurística proposta, utilizamos como medida de densidade $\frac{|P|-k}{R(P)}$, onde $|P|$ é o número de objetos dentro de uma partição P, $R(P)$ é o valor da diagonal do MBR da partição P e k o número de vizinhos considerados.

3.3. Busca por exceções

Nessa fase, recebemos um conjunto ordenado de partições, $\{P_1, P_2, P_3, ..., P_l\}$, onde P_1 é o agrupamento com maior chance de possuir um objeto raro de acordo a heurística utilizada, enquanto o agrupamento P_l contém objetos com menores chances de serem exceções. A busca por exceções segue a ordem da lista de partições, ou seja, inicia-se nos objetos de P_1 e termina nos objetos do agrupamento P_l.

[2] A abordagem K-protótipos [Huang 1998] foi utilizada para a mineração de atributos numéricos e categóricos.

Suponha, então, que estamos iniciando a busca por exceções na partição P_1. Selecionamos um objeto p qualquer dessa partição e iniciamos a busca pelos vizinhos próximos de p. A busca pelos vizinhos próximos de p inicia-se no próprio agrupamento de p. Contudo, se necessitarmos comparar p com outros objetos da base, prosseguimos para os agrupamentos seguintes do conjunto ordenado, $\{P_2, P_3, ..., P_l\}$ nessa ordem, mesmo que essa não seja a classificação das partições mais próximas de P_1. Note que, no algoritmo RBRP, a busca por vizinhos fora da própria partição prossegue, de forma ordenada, das partições mais próximas às mais distantes da partição em questão.

Note que a otimização implementada no algoritmo RBRP e a nossa abordagem de ordenação podem ser utilizadas conjuntamente, melhorando ainda mais a eficiência da busca por exceções. No entanto, isso não foi feito, pois pretendemos mostrar a diferença de impacto entre as duas propostas no desempenho dos algoritmos.

4. Resultados Experimentais

Nesta seção, faremos uma análise experimental de nosso algoritmo. Por falta de espaço, selecionamos quatro bases representativas do conjunto total de 10 bases de dados. Apresentamos os resultados obtidos a partir de duas bases de dados reais[3] e duas sintéticas[4]. O teste-t foi utilizado para comprovar se os resultados demonstram de forma significativa a diferença entre as diferentes heurísticas e algoritmos.

- *CoverType* : Base de dados que representa tipos de floresta que cobrem regiões de 30x30 metros na região de Rocky Mountain [Bay et al. 2000].
- *Itens Pregão* : A base de dados Itens Pregão possui registros que contém informações referentes a Compras Governamentais de diversas instituições do Brasil [Projeto Tamandua 2006].
- *Agrupamentos* : Esta base sintética é formada por agrupamentos uniformes e gaussianos muito bem definidos no espaço de (2, -2).
- *Agrupamentos com Ruído* : Formada pela base de dados Agrupamentos adicionando poucos ruídos de objetos seguindo uma distribuição uniforme (2, -2).

Base de dados	Objetos	Atrib. Reais	Atrib. Categóricos
Itens Pregão	268.170	6	7
Forest Covertype	581.012	10	45
Agrupamentos	500.000	30	0
Agrupamentos com Ruído	500.500	30	0

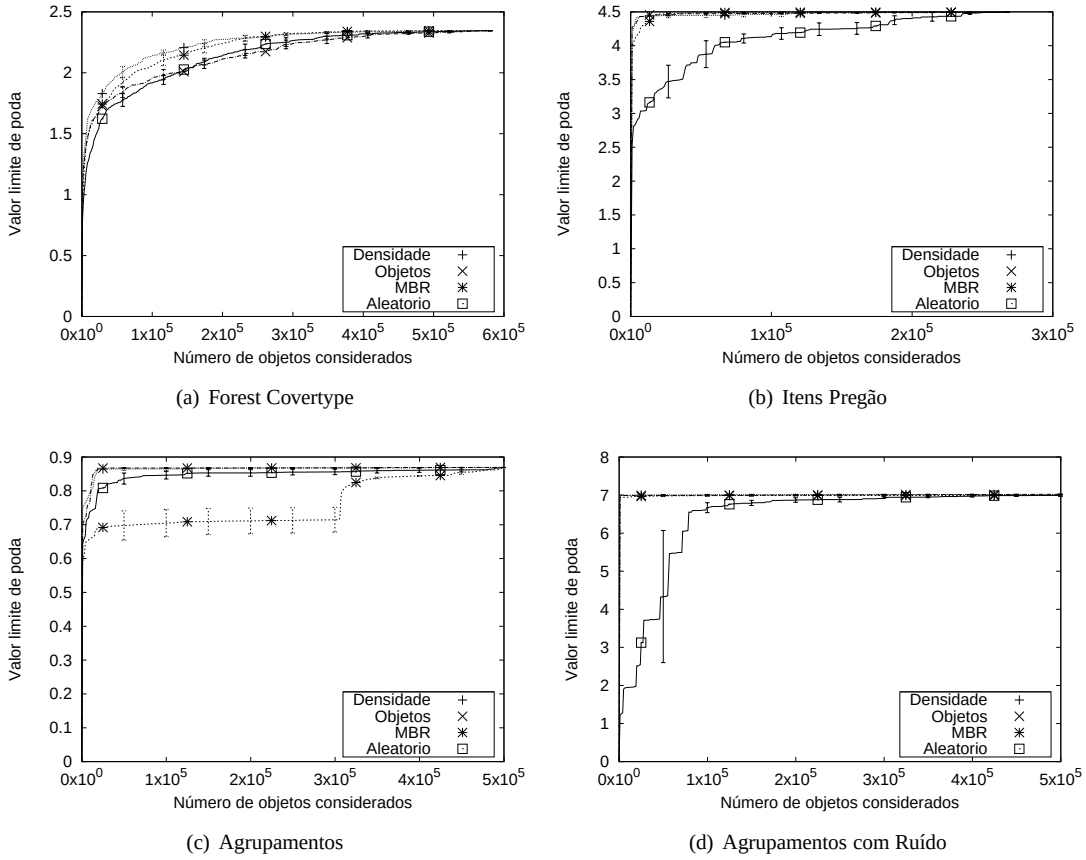
Tabela 2. Descrição das bases de dados

4.1. Eficácia das heurísticas de ordenação

Para estudar a eficácia das heurísticas de ordenação quanto ao fato de encontrarem "boas" exceções, medimos o valor D^k_{min} (limite de "poda" mínimo) em 10 execuções utilizando cada heurística de ordenação. Note que o valor de D^k_{min} mensura a excepcionalidade dos objetos raros encontrados até o dado momento. Os gráficos com o intervalo de confiança de 90% são apresentados na Figura 1 considerando as bases de dados reais e sintéticas.

[3]Os atributos numéricos foram normalizados segundo a distribuição normal e os valores categóricos passados para uma representação inteira.

[4]Bases com partições bem definidas e usadas para mostrar o impacto causado pela inserção de ruído no desempenho do algoritmo.



(a) Forest Covertype (b) Itens Pregão

(c) Agrupamentos (d) Agrupamentos com Ruído

Figura 1. Convergência de D^k_{min} para as bases de dados com 90% de confiança

Podemos ver nos gráficos (Fig. 1) que as heurísticas conseguem elevar o valor do limite de "poda" mínimo percorrendo uma quantidade menor de objetos. Comprovamos que as heurísticas de ordenação densidade, objetos e MBR foram estatisticamente superiores à estratégia aleatória quanto à convergência de D^k_{min}. Nas bases reais, as heurísticas densidade e objetos claramente se destacaram. Na base de dados *Agrupamentos* (Figura 1(c)) vemos que a heurística MBR estima de forma muito ruim onde estão as exceções causando uma convergência retardada do valor do limite de "poda". Isso ocorreu porque, nessa base, os agrupamentos foram gerados com tamanho espacial muito próximos, o que fez o tamanho das partições não ajudasse a encontrar onde estão as exceções. Contudo, na presença de ruído, essa mesma heurística conseguiu melhorar extraordinariamente a convergência do valor D^k_{min} (Fig. 1(d)). Note que o valor das exceções de 0.87 na base *Agrupamentos* saltou para 8 com a adição de ruídos. Portanto, esperamos que o tempo de execução para a base *Agrupamentos com ruído* seja menor, pois o valor de D^k_{min} convergiu muito rapidamente.

Com esses resultados, validamos a hipótese de que podemos encontrar objetos com um alto valor de excepcionalidade rapidamente, com heurísticas simples, baseando-se nas características dos agrupamentos dos dados.

4.2. Eficiência das heurísticas de ordenação

Para demonstrar o impacto das heurísticas de ordenação na eficiência do algoritmo, realizamos comparações pareadas com as diferentes heurísticas propostas para todas as ba-

ses de dados citadas, executando 10 replicações de cada experimento[5]. Os tempos de execução podem ser vistos na tabela 3, o intervalo de confiança de dois lados foi calculado com um nível de significância de 10% (confiança de 90%). Note que o ganho médio foi calculado considerando a heurística que obteve o melhor desempenho em relação à heurística aleatória, a linha de base utilizada.

Como podemos ver na tabela 3, os resultados mostram que a utilização de heurísticas de ordenação impactam significativamente no desempenho do algoritmo. O ganho em eficiência na base *Forest Covertype* foi cerca de 15% enquanto na *Itens Pregão* obteve uma surpreendente melhora de 82%. Nossas heurísticas continuam sendo superiores com ganho de 36% e 71% para as bases *Agrupamentos* e *Agrupamentos com ruído*, respectivamente. Note que, após a inserção do ruído na base *Agrupamentos* nossas heurísticas se distanciaram ainda mais da estratégia aleatória quanto à eficiência, principalmente a heurística MBR. Além disso, o tempo de execução da base de dados com ruído foi muito menor comparado à base *Agrupamentos*, como esperávamos.

	Heurísticas				
Database	Densidade	Objetos	MBR	Aleatória	Ganho Médio
Forest Covertype	**274,38**	302,3	310,71	321,81	14,74%
Itens Pregão	**12,82**	**10,53**	15,27	57,56	81,70%
Agrupamentos	**111,75**	**109,29**	198,02	170,27	35,81%
Agrupamentos com ruído	36,09	38,31	**30,56**	106,05	71,18%

Tabela 3. Comparação do tempo de execução das heurísticas de ordenação com 90% de confiança

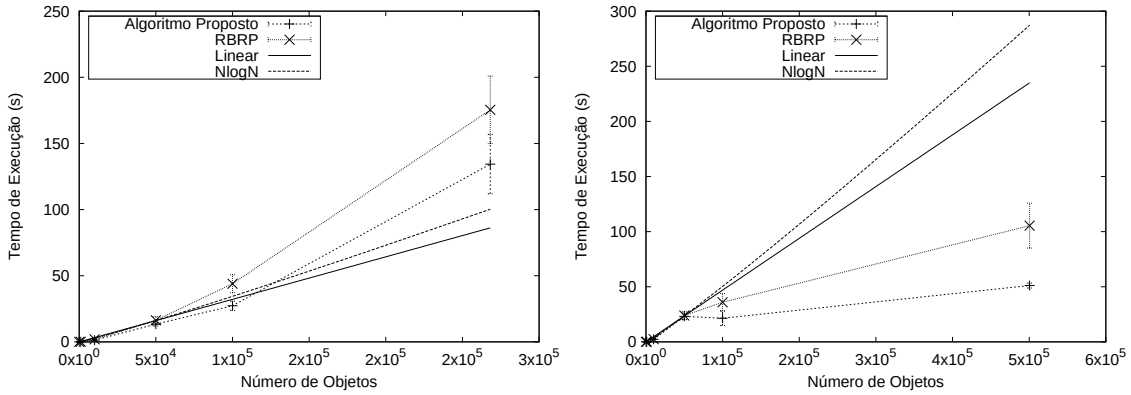
Analisando mais profundamente os resultados obtidos, podemos ver que há uma correlação entre a convergência do valor D^k_{min} e o desempenho das heurísticas. Isso explica o desempenho inferior da heurística MBR, na base *Agrupamentos*, comparado às demais heurísticas, inclusive comparado à ordenação aleatória. Ainda, observamos nitidamente que as heurísticas densidade e objetos conseguiram excelentes resultados em todas as bases de dados, confirmando mais uma vez a dependência da função de convergência no desempenho do algoritmo. Com isso, validamos nossa hipótese de que encontrar objetos raros rapidamente é um fator determinante para um bom desempenho do algoritmo.

4.3. Comparação entre os algoritmos

Avaliaremos agora, a escalabilidade de nosso algoritmo[6] em relação ao número de objetos das bases de dados comparando com o algoritmo estado da arte RBRP[7]. Por falta de espaço, os gráficos mostram os resultados (com 90% de confiança) apenas para a base *Itens pregão* e a base sintética *Agrupamentos com ruído*.

Verificamos na Figura 2 que nossa abordagem é consistentemente superior ao algoritmo RBRP em todas as bases testadas. O gráfico 2(a) mostra o ganho de 23% de nosso algoritmo utilizando os registros de compras governamentais do Brasil (Itens pregão). Para a base *Agrupamentos com ruído*, nossa estratégia foi 52% mais eficiente que o estado da arte, ambos apresentando uma característica sub-linear neste caso. Isso mostra que, analisar primeiramente os objetos com um alto valor de excepcionalidade

[5]Os tempos de preparação da base de dados não foram considerados durante as análises das heurísticas.
[6]A heurística baseada na densidade dos agrupamentos foi utilizada em nossos experimentos.
[7]Foi implementada uma versão estendida do algoritmo RBRP utilizando a abordagem *k-protótipos*.



(a) Itens Pregao (b) Agrupamentos com Ruído

Figura 2. Escalabilidade com relação ao número de objetos das bases de dados com 90% de confiança

(*busca ordenada por exceções*) tem um impacto maior sobre o desempenho comparado com a técnica de encontrar vizinhos próximos rapidamente (*busca ordenada de vizinhos próximos*).

5. Conclusão e Trabalhos Futuros

Neste trabalho, formulamos e comprovamos duas hipóteses: (1) analisar primeiramente registros com um alto valor de excepcionalidade impacta diretamente no desempenho dos algoritmos para mineração de exceções baseados em distância, (2) podemos encontrar "boas" exceções, de forma simples, usando as características dos agrupamentos dos dados. Propomos a busca ordenada por exceções e avaliamos quatro heurísticas para otimização da regra de "poda". Nossos resultados provaram a efetividade das heurísticas propostas na busca por exceções e como essa efetividade afeta diretamente o desempenho do algoritmo. Demonstramos que nosso algoritmo obtém um desempenho até 52% superior em relação ao estado da arte RBRP através da redução do número de cálculos de distância entre objetos.

Como trabalhos futuros, estamos realizando um estudo de caso, a identificação e análise dos registros anormais encontrados na base de dados de compras governamentais do Brasil, a Itens Pregão. Esses objetos raros, podem representar operações ilícitas, fraudes, ou mesmo uma inserção errada de informação. Além disso, analisaremos de forma detalhada várias contribuições importantes na área dos algoritmos de mineração de exceções baseados em distância. Dentre eles podemos citar a *busca ordenada por vizinhos próximos* do algoritmo RBRP, a "poda" de partições inteiras de Ramaswamy e a *busca ordenada por exceções* apresentada neste trabalho. Com isso, pretendemos avaliar o impacto de cada uma dessas abordagens na eficiência dos algoritmos e como elas se relacionam entre si.

6. Agradecimentos

Este trabalho é parcialmente financiado por CNPq, FINEP e FAPEMIG.

Referências

Bay, S. D., Kibler, D., Pazzani, M. J., and Smyth, P. (2000). The uci kdd archive of large data sets for data mining research and experimentation. *SIGKDD Explor. Newsl.*, 2(2):81–85.

Bay, S. D. and Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *9th ACM SIGKDD Int. Conf. on Knowledge Discovery on Data Mining*.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517.

Ester, M., Kriegel, H. P.and Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial fatabases with noise. In *In Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*. AAAI Press.

Ghoting, A., Parthasarathy, S., and Otey, M. E. (2005). Fast mining of distance-based outliers in high-dimensional datasets. *6th SIAM Int. Conf. on Data Mining*.

Hartigan, J. A. and Wong, M. A. (1979). A K-means clustering algorithm. *Applied Statistics*, 28:100–108.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Min. Knowl. Discov.*, 2(3):283–304.

Knorr, E. M. and Ng, R. T. (1999). Finding intensional knowledge of distance-based outliers. In *VLDB '99: 25th Int. Conf. on Very Large Data Bases*, pages 211–222, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ord, K. (1996). Outliers in statistical data : V. barnett and t. lewis, 1994, 3rd edition, (john wiley & sons, chichester), isbn 0-471-93094. *Int. Journal of Forecasting*, 12(1):175–176.

Projeto Tamandua (2006). Projeto Tamandua. http://tamandua.speed.dcc.ufmg.br/.

Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *SIGMOD '00: Proc. ACM SIGMOD Int. Conf. on Management of data*, pages 427–438, New York, NY, USA. ACM Press.

Roussopoulos, N., Kelley, S., and Vincent, F. (1995). Nearest neighbor queries. In *SIGMOD '95: ACM SIGMOD Int. Conf. on Management of data*, pages 71–79, New York, NY, USA. ACM.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. *SIGMOD Rec.*, 25(2):103–114.