W-SAGE: Ferramenta Web para Análise de Dados Geoespaciais

Raul S. Ferreira[1], **Carlos E. Mello**[1,2]

[1]COPPE - Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro, Brasil

[2]Departamento de Sistemas de Informação
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Rio de Janeiro, Brasil

raulsf@cos.ufrj.br, carlos.mello@ufrj.br

Abstract. *Density estimation is a important statistical method that helps to determine, in a concise way, the probability of a phenomenom about certain types of data. Some kinds of data doesn't have a known distribution but may reveal too much about certain domains, for instance, the geographic data. This work seeks to create an intuitive interface for geographic data analysis together with non geographic data, using a non-parametric density estimator with parallelization techniques in GPU, with the objective to show important perceptions about the observed database. Are shown the first empirical results about the tool W-SAGE.*

Resumo. *Estimar densidades é uma importante técnica estatística, que ajuda a determinar, de forma mais precisa, a probabilidade de um fenômeno sobre determinados tipos de dados. Certos dados não possuem distribuição conhecida a priori mas podem revelar muito sobre um determinado domínio, como é o caso dos dados geográficos. Este trabalho busca construir um sistema veloz com uma interface intuitiva para análise desses dados geográficos ao utilizar um método estimador de densidade não-paramétrico em conjunto com técnicas de paralelização em GPU, com o intuito de disponibilizar percepções importantes sobre a base de dados observada. São mostrados os primeiros resultados empíricos desta ferramenta, entitulada W-SAGE.*

1. Introdução

A informação geográfica tem grande importância em diversas áreas como, marketing, agricultura, meio ambiente, saúde, planejamento urbano entre outros, ajudando na tomada de decisões e estratégias além de agregar valor como um meio de representação visual mais expressiva do que uma representação discreta. Várias ferramentas podem ser criadas para extrair o máximo de informações agregadas à distribuição espacial, informações essas que não poderiam ser extraídas através do modo convencional de análise de dados não espacial. A informação geográfica já existe há centenas de anos (e.g., mapas) e como em vários aspectos do nosso cotidiano esta também foi e vem sendo alterada pela modernidade tecnológica.

Para lidar com esse tipo de informação, surgiram então os sistemas de informações geográficas (SIG). Os SIG são sistemas utilizados para armazenar, analisar, manter e manipular dados geográficos de maneira automatizada [Bolstad 2005]. Os dados geográficos

utilizados pelos SIG podem ser imagens digitalizadas (e.g., fotos de satélite) ou objetos que representam uma geometria no espaço, chamados objetos espaciais. Esses dados são armazenados e gerenciados por bancos de dados espaciais (objetos geométricos espaciais) e estes se encaixam bem dentro do conjunto de tecnologias que compõe este trabalho. Assim, com o auxílio de um banco de dados geográfico, técnicas de extração, tranformação e carregamento de dados, bancos de dados NoSQL(*Not only SQL*) e a utilização de métodos estatísticos, foi construída uma ferramenta de visualização de dados geográficos entitulada W-SAGE (*Web tool for Spatial Analysis of GEographic data*).

Desta forma, este trabalho está organizado em 6 capítulos, onde este é o primeiro. No segundo capítulo são apresentados os trabalhos relacionados às ferramentas de visualização de dados geográficos. No terceiro capítulo são mostradas as motivações na escolha das técnicas e tecnologias utilizadas neste trabalho, enquanto no quarto capítulo, é explicado com maiores detalhes a metodologia que guiou a construção da ferramenta. No quinto capítulo, encontram-se os resultados dos primeiros experimentos utilizando dados reais e no sexto e último capítulo são apresentadas as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

[Mello 2008] faz uso de objetos espaciais em conjunto com técnicas de clusterização com o intuito de tentar mostrar como a acessibilidade na cidade do Rio de Janeiro pode ser melhor modelada, observada e mostra como a disponibilização dessa visualização através de uma representação geográfica torna muito mais clara essa análise para o usuário do que uma representação puramente textual baseada em números e tabelas.

Em [Patrol 2012] podemos ver o trabalho entitulado *Active Missing Person Map*, realizado no Missouri, um estado norte americano, onde foi desenvolvido um sistema de visualização com um viés geográfico ao mostrar em um mapa do território do estado, as quantidades de pessoas desaparecidas separadas por cidade e suas respectivas informações.

[Spina 2016] desenvolveu um trabalho de visualização de dados geográficos voltado para auxiliar o combate ao tráfico humano fazendo uma análise sobre o problema de lavagem de dinheiro. Já em [Zuo et al. 2016] o foco é em desenvolver ferramentas de visualização para auxiliar o descobrimento de padrões geoquímicos na área de geologia, como anomalias e determinadas restrições dentro de uma região.

Outro trabalho recente na área de visualização é descrito em [Guo et al. 2015], onde é proposto um novo *workflow* para tentar lidar com problemas geográficos que possuem a necessidade de computação intensa, usando métodos de paralelização tanto na parte de visualização quanto na parte de processamento, algo parecido com a proposta deste trabalho.

3. Visualização e processamento de dados geográficos

Como pode-se ver, vários trabalhos recentes procuraram lidar com alguma dificuldade no processamento de informações geográficas tanto no lado cliente quanto no lado servidor. Várias técnicas podem ser combinadas para proporcionar *insights* aos especialistas. No caso deste trabalho, iremos agregar em um sistema de visualização geográfico um algoritmo de inferência estatística, rasterização e clusterização de pontos, além de usarmos bancos de dados específicos para tarefas específicas dentro do fluxo da ferramenta.

3.1. Estimando Densidades

Estimar densidades sobre uma população é um trabalho importante e geralmente usamos uma função de densidade para isso. A densidade de uma população pode ser estimada com várias técnicas estatísticas, porém estatisticamente, alguns dados ou populações não possuem estruturas ou parâmetros característicos, no caso, estes dados são conhecidos como não paramétricos. Dados não paramétricos não dependem de dados pertencentes a nenhuma distribuição particular. Tipicamente, o modelo não-paramétrico cresce no sentido de acomodar a complexidade dos dados. Como métodos não paramétricos fazem menos suposições, a aplicabilidade deles é mais larga que os correspondentes métodos paramétricos. Em particular, eles podem ser aplicados em situações em que menos se sabe sobre o problema em questão. Devido a menor dependência de hipóteses, métodos não paramétricos são mais robustos. Exemplo de dado não-paramétrico: distribuição tem a forma normal, tanto a média quanto a variância não foram especificadas.

A função de probabilidade é um conceito fundamental em estatística e existem diversas técnicas que podem ser empregadas para estimar dados não paramétricos e um dos métodos mais conhecidos, é o estimador de densidade de kernel ou KDE(*Kernel Density Estimation*), também conhecido como Janela de Parzen[Duda et al. 2012]. Neste método são utilizadas funções não-lineares como Gaussianas e Sigmóides para se computar a densidade local de cada instância, logo, este trabalho lança mão desta técnica devido à simplicidade e à eficiência perante a literatura para tratar esses tipos de dados.

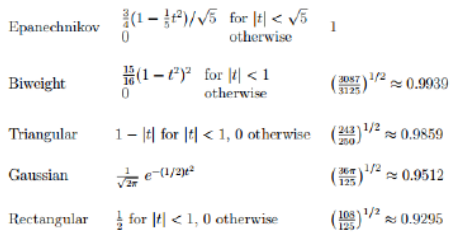
3.2. Kernel Density Estimation

O KDE é uma das técnicas de estimativa de densidade mais comuns e é bastante usada para normalizar e suavizar a distribuição de um determinado conjunto de dados. O KDE pode ser pensado como uma generalização do histograma. Possui duas variações: Univariante, cuja a entrada são dados de uma única dimensão, no caso um vetor; e Multivariante, cuja a natureza dos dados de entrada é de duas ou mais dimensões, usando uma matriz para armazenamento dos dados. É esta versão multivariante que usaremos para estimar as densidades dos pontos geográficos (latitude, longitude).

O algoritmo KDE multivariante é descrito na figura 2 e seu processamento é feito levando-se em consideração cada indivíduo em relação a sua população e a sua complexidade é $O(n^2k)$, pois é implementado como um somatório de um produtório de matrizes e portanto, dependendo do número de dimensões k na entrada, o algoritmo pode-se tornar um tanto quanto lento para apresentar a estimativa de densidade final ou mais conhecido como PDF (*Probability Density Function*), que é calculado para cada indivíduo em relação a sua população.

A partir de um dado número de observações n, calculamos curvas de densidade delas em relação à distância de um valor central, o núcleo, para cada um desses pontos e obtemos a Estimativa de Densidade final somando esses valores. Um kernel é uma função de ponderação padronizada, ou seja, o núcleo determina a suavização do PDF. Esta técnica é amplamente usada em vários algoritmos de aprendizado de máquina, principalmente em SVM (*Support Vector Machines*)[Hearst et al. 1998]. Esta função Kernel precisa ser cuidadosamente escolhida pois pode provocar um super ajustamento (*overfitting*) ou o contrário (*underfitting*) nos valores dos PDFs [Duda et al. 2012][Bishop et al. 2006], no caso deste trabalho será utilizado o Kernel Gaussiano, pois produz uma estimativa mais

suave, porém há outros tipos de funções, como descrito na figura 1.



Epanechnikov	$\frac{3}{4}(1-\frac{1}{5}t^2)/\sqrt{5}$ for $\lvert t\rvert < \sqrt{5}$, otherwise 0	1
Biweight	$\frac{15}{16}(1-t^2)^2$ for $\lvert t\rvert < 1$, otherwise 0	$\left(\frac{3087}{3125}\right)^{1/2} \approx 0.9939$
Triangular	$1-\lvert t\rvert$ for $\lvert t\rvert < 1$, 0 otherwise	$\left(\frac{243}{250}\right)^{1/2} \approx 0.9859$
Gaussian	$\frac{1}{\sqrt{2\pi}}\, e^{-(1/2)t^2}$	$\left(\frac{36\pi}{125}\right)^{1/2} \approx 0.9512$
Rectangular	$\frac{1}{2}$ for $\lvert t\rvert < 1$, 0 otherwise	$\left(\frac{108}{125}\right)^{1/2} \approx 0.9295$

Figure 1. Tipos de Kernel.

```
for i ← 0 to n do
    soma_kernel ← 0.0
    for j ← 0 to n do
        prod_kernel ← 1.0
        for k ← 0 to xLen do
            prod_kernel *K((x[i][k] − x[j][k])/h)/h
        end
        soma_kernel ← soma_kernel + prod_kernel
    end
    pdf[i] ← soma_kernel / n
end
```

Figure 2. KDE Multivariante.

3.3. Processamento paralelo do KDE em GPU

Devido ao custo de processamento do KDE foi preciso pensar em como melhorar o desempenho do algoritmo. Devido ao fato do algoritmo ser paralelizável em certas partes de seu processamento então foi implementada uma versão que faz uso da paralelização massiva dos passos da aplicação. Utilizamos então, placas gráficas GPU (*Graphics Processor Unit*) através da linguagem CUDA(*Compute Unified Device Architecture*), que devido a sua natureza SIMD (*Single Instruction Multiple Data*) fornece quantidades superiores de processamento e desempenho muito maior do que processadores convencionais [Sanders and Kandrot 2010].

3.4. Sistema de cache, clusterização e rasterização de dados

Para evitar que toda a vez que uma nova consulta fosse feita, uma nova requisição ao banco de dados e um novo processamento do KDE fosse realizado, foi utilizada uma estratégia de cache dos dados de consulta. Para tal, foi utilizado um banco de dados NoSQL, baseado em chave-valor, amplamente utilizado para problemas com alta latência de dados, conhecido como Redis[Han et al. 2011]. Desta forma, este banco funciona como um cache da consulta e assim, melhoramos ainda mais o gargalo de processamento dos PDFs resultantes do KDE e do acesso ao banco de dados geográfico, o que melhora ainda mais a velocidade de resposta da aplicação.

Além disso, devido à grande quantidade de pontos no navegador foi preciso pensar em uma estratégia para diminuir o carregamento da página. A saída foi implementar a clusterização dos pontos por distância, ou seja, pontos que estão muito próximos entre si ou que possuam a mesma coordenada não precisam aparecer individualmente mas apenas um único ponto representando o total de pontos contidos naquela região.

Para isso, foi utilizada a clusterização de pontos no lado cliente da aplicação. Desta forma, ao utilizar a visualização de pontos no mapa da aplicação, caso o usuário esteja em um nível mais distante de zoom então círculos contendo apenas uma contagem total de pontos referentes a uma área aparece e caso o usuário dê o zoom na página então essa distância relativa tenderá a aumentar e consequentemente os pontos, que antes estavam clusterizados, aparecerão em outros *clusters* menores e assim sucessivamente até aparecer o próprio ponto. Dessa forma, evitamos que vários pontos fiquem agrupados em

um espaço pequeno ou até mesmo um em cima do outro, dificultando a visualização e deixando a página desnecessariamente carregada.

Outro ponto importante na implementação foi o uso de rasterização dos pontos, onde convertemos uma imagem vetorial em uma imagem raster (pixels ou pontos) e colocamos em *buffer*. Dessa forma os pontos deixam de ser objetos e caso uma fique sobre a outra a imagem acima tende a sobrescrever o que estiver abaixo dela, evitando que múltiplas imagens empilhadas sejam criadas na tela.

4. W-SAGE: *Web tool for Spatial Analysis of GEographic data*

Foram realizados dois estudos de caso com a utilização do W-SAGE e este foi construído em 2 partes: Lado servidor ou mais comumente chamado de back-end, responsável pelo processamento do KDE e pelo cache das consultas ao banco de dados geográfico; Lado cliente ou simplesmente front-end, responsável por intermediar a interface do usuário com o servidor além de garantir a visualização das consultas em formato de cluster, mapa de calor e gráficos. No primeiro estudo de caso, foi usada uma base de dados contendo a localização das agências do IBGE pelo país. Já no segundo caso de teste, foi utilizada uma base anonimizada referente à alguns dados de matrícula dos alunos da Universidade Federal Rural do Rio de Janeiro (UFRRJ).

4.1. Coleta dos dados

Os dados do IBGE foram coletados no próprio site, nesta base aproveitamos apenas os atributos de identificação numérica e as coordenadas geográficas, totalizando 570 pontos representando as agências do IBGE pelo país. Para o segundo experimento, foram coletados junto à diretoria da UFRRJ, em planilha excel, alguns dados de matrícula de todos os alunos que se matricularam na universidade de 2000 até 2013, totalizando 14027 registros. Os dados fornecidos foram: Cep, situacao da matrícula, status de bolsista, sexo, nascimento, naturalidade, forma de ingresso, periodo real, período cronológico, campus, código do curso, cr acumulado e percentual integralizado. Não houve coleta de nomes ou qualquer tipo de dado sócio-econômico.

4.2. Modelo da aplicação

Conforme pode ser visto na figura 3, o sistema foi desenvolvido com uma arquitetura própria para coleta de dados heterogêneos que podem ser facilmente guardados como documentos no banco de dados NoSQL orientado à documentos, no caso deste trabalho, o banco escolhido foi o MongoDB[Han et al. 2011].

O modelo da aplicação foi desenvolvido pensando-se em uma arquitetura simples de automação, onde num primeiro passo coleta-se os dados de diferentes fontes, colocando-os em um banco de dados orientado à documentos. Em seguida, processos de ETL(*Extract, Transform and Load*) podem ser definidos de acordo com os dados coletados para em seguida serem jogados no banco de dados relacional, a partir daí, a aplicação pode ou não processar uma consulta, baseada no cache salvo pelo banco de dados chave-valor, Redis.

O cache na aplicação foi implementado da seguinte forma: Assim que uma consulta é feita, gera-se um *hash* dos parâmetros desta consulta e salva-se este *hash* como chave e o resultado da consulta como o valor. Caso o usuário faça uma nova consulta, é

feito o mesmo processo descrito e caso a chave seja igual ao que o Redis já possui, este traz o valor sem precisar ir ao banco e caso seja uma nova chave, o Redis guarda esta nova chave e o resultado da consulta como valor, e assim por diante.

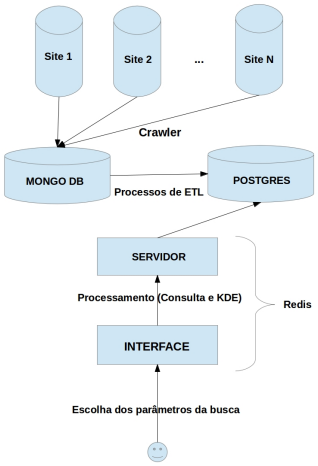


Figure 3. W-SAGE: Fluxo da aplicação

4.3. Visualização dos dados

Depois de enviada a requisição de consulta para o servidor, este recupera os dados e em seguida processa o KDE que por sua vez entrega o resultado, no caso, um vetor de pdf, para que seja criado o mapa de calor onde atribui-se aos pesos dos pontos os valores dos PDFs, gerando no mapa, colorações que variam do azul (menor peso) ao vermelho (maior peso). Quanto maior o peso maior a probabilidade de ocorrer o fenômeno estudado. Os pontos então são processados através de clusterização e rasterização e colocados no mapa juntamente com o mapa de calor e os demais gráficos.

5. Resultados

Os resultados apresentados estão balizados em três premissas:

1. Verossimilhança do resultado visual do estimador de densidade processado pelo sistema em relação ao mesmo algoritmo fornecido em um software comercial;
2. Velocidade de processamento do KDE em um software comercial comparado ao W-SAGE;
3. Facilidade de visualização e suporte a tomada de decisão através da ferramenta.

Como parâmetro para comparação, foi usado o já consagrado software comercial Matlab[Guide 1998] para rodar o KDE gaussiano com as mesmas coordenadas usadas na aplicação e depois seus tempos de processamento foram comparados. Em seguida, a imagem gerada pelo Matlab foi comparada com o mapa de calor gerado pela aplicação, para assim podermos comparar a distribuição gerada pelos dois métodos.

A figura 4 mostra o resultado do KDE gerado nos dados do IBGE pelo matlab e a figura 5 mostra o resultado do KDE gerado nos dados do IBGE pelo W-SAGE. Na figura 6 e na figura 7 são mostrados o KDE no MATLAB e no W-SAGE respectivamente, enquanto na figura 8 uma visão mais completa do mesmo resultado no W-SAGE, já com

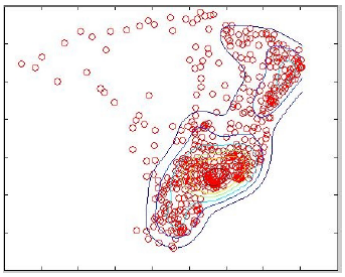


Figure 4. KDE gerado nos dados do IBGE pelo MATLAB.

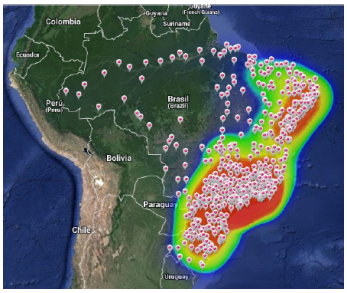


Figure 5. KDE gerado nos dados do IBGE pelo W-SAGE.

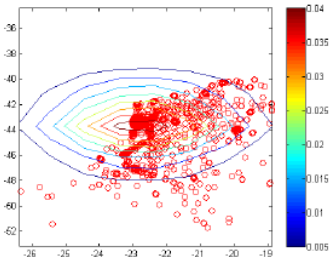


Figure 6. KDE gerado nos dados da UFRRJ pelo MATLAB.

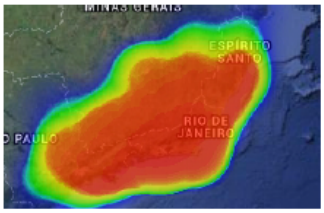


Figure 7. KDE gerado nos dados da UFRRJ pelo W-SAGE.

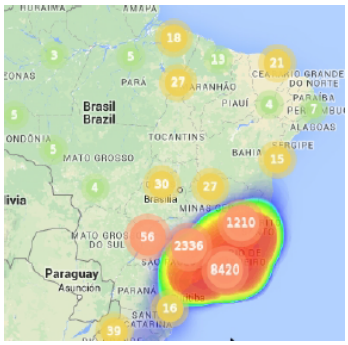


Figure 8. W-SAGE: Resultado de uma consulta.

os pontos rasterizados, clusterizados e com mapa de calor provindo dos PDFs calculados. A velocidade de processamento foi calculado com detalhes em [Ferreira et al. 2014] e um resumo pode ser visto na tabela 1 onde o ganho do KDE paralelizado com GPU foi de 3024% sobre a versão sequencial e 6% superior à versão paralelizada do MATLAB.

Table 1. Tabela comparativa

Tabela 1. Tempos medidos.

KDE Sequencial	KDE Paralelizado Matlab	KDE c/ CUDA	KDE c/ CUDA otimizado	Speed-Up (GPU x Serial)	Speed-Up (GPU x Matlab)
31.088s	6.355s	1.680s	1.028s	30,241	6,181

6. Conclusão

Através deste trabalho conseguimos integrar um método estimador de densidades, com resultados parecidos com o de softwares reconhecidos pelo mercado, à uma aplicação web de análise de dados geográficos. A ferramenta se mostra escalável para o carregamento de milhares de pontos no browser se for necessário, ao paralelizar uma parte do algoritmo usando GPU e ao utilizar técnicas de clusterização e rasterização de coordenadas geográficas. Além disso, foi utilizada uma estratégia para lidar com dados heterogêneos além da utilização de *caching* evitando reprocessamento de consultas. O sistema proporciona assim, a análise dos dados geográficos, não só limitada a estes, de forma intuitiva, com qualidade e com boa velocidade de resposta.

6.1. Trabalhos Futuros

Como trabalho futuro pretendemos utilizar bancos de dados distribuídos e uma arquitetura assíncrona preparada para receber muitas requisições (Node.js), para tentarmos escalar o sistema para *datasets* com milhões de pontos. Outro ponto importante é a questão da substituição do KDE tradicional por um modelo mais rápido do algoritmo, recentemente proposto por [O'Brien et al. 2016] onde a complexidade do KDE, que é quadrática, foi reduzida para linear.

References

Bishop, C. M. et al. (2006). Pattern recognition and machine learning, vol. 1springer. *New York*, (4):12.

Bolstad, P. (2005). *GIS Fundamentals: A First Text on Geographic Information Systems.* Eider Press.

Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Ferreira, R. S., Valenzuela, J. E. H., and Zamith, M. P. (2014). Paralelização do algoritmo de método de estimação não-paramétrico por núcleo estimador multivariado (kde) utilizando gpu/cuda. In *II Reunião Anual de Iniciação Científica da UFRRJ*.

Guide, M. U. (1998). The mathworks. *Inc., Natick, MA*, 5:333.

Guo, M., Guan, Q., Xie, Z., Wu, L., Luo, X., and Huang, Y. (2015). A spatially adaptive decomposition approach for parallel vector data visualization of polylines and polygons. *International Journal of Geographical Information Science*, 29(8):1419–1440.

Han, J., Haihong, E., Le, G., and Du, J. (2011). Survey on nosql database. In *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, pages 363–366. IEEE.

Hearst, M. A., Dumais, S. T., Osman, E., Platt, J., and Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.

Mello, C. E. R. (2008). *Agrupamento de regiões: Uma abordagem utilizando acessibilidade*. PhD thesis, UNIVERSIDADE FEDERAL DO RIO DE JANEIRO.

O'Brien, T. A., Kashinath, K., Cavanaugh, N. R., Collins, W. D., and O'Brien, J. P. (2016). A fast and objective multidimensional kernel density estimation method: fastkde. *Computational Statistics & Data Analysis*, 101:148–160.

Patrol, M. (2012). Active missing person map.

Sanders, J. and Kandrot, E. (2010). *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional.

Spina, M. (2016). *New techniques for combatting human trafficking; specifically through the analysis of anti-money laundering and geographic data visualization technology*. PhD thesis, UTICA COLLEGE.

Zuo, R., Carranza, E. J. M., and Wang, J. (2016). Spatial analysis and visualization of exploration geochemical data. *Earth-Science Reviews*, 158:9–18.