Sensores Sociais em Detecção de Eventos Sociais

Augusto Zangrandi, Luis A. Rivera

Laboratório de Ciências Matemáticas - LCMAT
Universidade Estadual do Norte Fluminense - UENF
Av. Alberto Lamego, 2000; CEP 28015-620, Campos dos Goytacazes – RJ– Brasil

zangrandii@gmail.com, rivera@uenf.br

Abstract. *Event detection consists in the identification of relevant patterns of change in a system. The proposed model detects events that occur in some part of the territory from the publications in the social network Twitter. The publications are obtained from Twitter, extracted their characteristics and classified with support vector machine for their interpretation as events. Detected events are grouped and displayed in time series and marker maps in certain regions of the territory under analysis.*

Resumo. *Detecção de eventos consiste no processo de identificação de padrões de mudança relevantes em um sistema. O modelo propuesto detecta eventos que ocorre em alguma parte do território a partir das publicações na rede social Twitter. As publicações são obtidas do Twitter, são extraídas suas características e classificadas com máquina de vetores de suporte para sua interpretação como eventos. Os eventos detectados são agrupados e exibidos em gráficos no formato de série-temporal e mapas de marcadores em determinadas regiões do território em análise.*

1. Introdução

Os eventos são acontecimentos que mostram câmbios significativos ou ocorrências anômalas em relação ao comportamento geral de um sistema. A **detecção de eventos** decorre do processo de identificação de acontecimentos que fogem das regras normais de funcionamento de um sistema, ou de padrões de mudanças relevantes dentro dos mesmos. No sistema de monitoramento de tópicos em documentos de texto, por exemplo, um evento pode ser o repentino surgimento de documentos contendo um termo específico, ou um novo tópico. A frequência maior ou menor desses eventos pode ser fundamental para o sistema.

Um sistema captura as informações do ambiente através de *sensores* – temperatura, velocidade, som, documentos de texto, outras mídias –, servindo de dados para detecção de eventos (Sakaki et al, 2010). Um *usuário* de um serviço de uma rede social (SNS: *Social Networking Services*) também é considerado um sensor social e cada mensagem publicada uma informação sensorial. As promulgações diárias de milhões de publicações, sobre os mais variados temas, tem despertado interesse acadêmico nas áreas de mineração de dados e detecção de eventos. O Twitter é um dos SNS mais populares com maior número de usuários e publicações no mundo que trocam informações em tempo real.

Métodos estatísticos, métodos probabilíticos e métodos de aprendizado de máquina são utilizados para a análise das informações. Neste trabalho, apresenta-se um sistema de detecção de eventos, utilizando serviços de Twitter. O referido sistema conta

com um aprendizado de máquina SVM (*Support Vector Machine*) e utiliza gráficos de série-temporal interativos para exibir erupções anômalas de publicações. O sistema é validado com os eventos de manifestações ocorridas no Brasil no período de 01 a 31 de agosto de 2014.

A organização do presente trabalho é feita da seguinte forma: inicialmente, na Seção 2, aborda-se a detecção de eventos sociais. A seguir, na Seção 3 formula-se o modelo de detecção de eventos através do Twitter. Na Seção 4 aborda-se os detalhes da implementação do modelo e análise de resultados. Na Seção 5, finalmente, conclui-se com a indicação de trabalhos futuros.

2. Sistema de Detecção de Eventos Sociais

O sistema deve ser capaz de transformar os dados oriundos dos sensores, bem como identificar os eventos inerentes a esses dados. Os dados dos sensores são de baixo-nível, ou seja, sem formato, geralmente incompleto, com descrições diretas acerca dos acontecimentos do mundo. A detecção deve transformá-los em dados de alto nível, de forma que seja possível a compreensão humana dos acontecimentos. O sistema deve agregar, converter e reformatar os dados recebidos em uma estrutura independente da fonte de dados (Fienberge e Shmueli, 2005).

2.1. Eventos nos Documentos de Texto

Os eventos relevantes são detectados através da análise dos padrões presentes nos documentos de texto. Neste caso, um evento indica uma ocorrência significativa no contexto de interesse de alguma atividade humana, tal como relacionados com shows musicais, festas, eventos políticos e de moda, entre outros. Weng e Lee (2011) classificam os métodos de detecção de eventos nos documentos de texto em dois tipos: *documento-pivô* e *recursivo-pivô*. O *documento-pivô* baseia-se na divisão de documentos em grupos, de acordo com a similaridade léxica de seus conteúdos. Alguns critérios técnicos devem ser considerados: a) **proximidade temporal** (documentos referentes ao mesmo evento costumam ser próximos temporalmente); b) **erupção de documentos similares** (eventos diferentes); c) **mudanças de frequência** (mudanças rápidas nas frequências de um termo, geralmente é sinal a um novo evento). Enquanto os métodos do tipo *recursivo-pivô* analisam a distribuição e a associação das palavras. Sakaki et al. (2010), em suas pesquisas, mostram que existem três recursos para a implementação: estatísticos (número de palavras e a posição da palavra-chave dentro do documento); palavras-chave (palavras de referência no documento); contexto de palavra (palavras antes e depois da palavra-chave). Sem embargo, segundo Aiello et al. (2013), os dois tipos possuem desvantagens. Os métodos *documento-pivô* possuem problemas com fragmentação de grupos; no contexto de aquisição de documentos em tempo real, eles dependem de limiares arbitrários para a inclusão de um documento em um grupo. Os métodos *recursivo-pivô* geralmente fazem associações errôneas entre palavras-chave. Portanto, o problema ainda não está fechado.

2.2. Sensores Sociais

Os *SNS* são plataformas online onde os usuários podem se relacionar criando perfis, compartilhando publicações de variados temas, acontecimentos, e atualizações em

formato texto, foto, áudio e vídeo. Os SNS mais populares, segundo a lista10.org[1], são: Facebook, Youtube, Qzone, Sina Weibo, WhatsApp, Google+, Tumblr, Line, Twitter, WeChat, entre outros. Os serviços necessários para os desenvolvedores é que eles possuam interfaces para aquisição das informações para a criação de serviços externos para variados fins (Dong et al., 2014).

Relacionado às SNS estão os *microblogs*, informações no formato texto curto, que permitem aos usuários fazerem rápidas atualizações e distribuições. O mais popular nesta categoria é o Twitter. No entanto, o microblog é um conceito que para outras ferramentas, como Facebook e Google+, é atualização de *status*. O usuário, oa vivenciar um evento, pode compartilhar com seu grupo de amigos e eles com outros, espalhando rapidamente os tópicos rapidamente no mundo inteiro. Dessa forma, os microblogs são uma fonte de informação sobre acontecimentos do mundo real (Mai-Hranac, 2013).

O Twitter permite manipular microblogs de publicações de texto até 140 caracteres. O dinamismo do serviço é definido por *trending topics* (assuntos do momento) e ranqueados os termos mais comentados do momento. O recurso *seguir* permite que um usuário U1, ao seguir outro usuário U2, passa a receber todas as publicações de U2 na sua página de *linha do tempo*. O trabalho de Java et al. (2007) apresenta a formação de comunidades e a motivação das pessoas ao utilizarem serviços de microblogs do Twitter. Eles observaram que a interação induz alta reciprocidade e correlação entre os usuários e a facilidade de pulverização de informação. Os serviços de Twitter, retenção de informações e formas de sumariza-los como os assuntos do momento, são importantes para a análise das informações (Matuszka et al., 2013).

2.3. Trabalhos Relacionados

Aplicados nos variados âmbitos, os trabalhos de detecção de eventos utilizam métodos probabilísticos, componentes principais, métodos de aprendizado de máquina e outros métodos heruísticos. Gupchup et al. (2009) utilizam a técnica de *Análise de Componente Principal* (ACP) para construir um modelo capaz de coletar as tendências das medidas de uma rede de sensores sem fio para detectar anomalias. Hong et al. (2014) utilizam a detecção de eventos para detectar intrusos em redes de computadores. Abou-Zleikha et al. (2014) utilizam o algoritmo *Random Forest* na detecção de eventos como risadas, momentos de silencio e outros, em dados vocais. O trabalho de Ihler et al. (2006) revela um modelo de Poisson, variável no tempo, para detectar eventos anômalos em dados de contagem de séries temporais.

Sakaki et al. (2010) desenvolvem uma técnica para detectar terremotos e tufões no Japão com publicações coletadas do Twitter. As palavras “terremoto” e “tremendo” são palavras-chave, utiliza SVM com kernel linear. Takahashi et al. (2011) desenvolvem um sistema que monitora publicações e detecta ocorrências de rinite alérgica no Japão com publicções do Twitter. Vinceller e Laki (2013) analisam o ciclo de vida de cada palavra chave comúm nos eventos, enfatizam que os aparecimentos de eventos específicos em redes sociais podem surgir antecipadamente aos outros meios de comunicação. Mai e Hranac (2013) analisam as publicações do Twitter relacionadas a acidentes de tránsito com palavras-chave “acidente”, “batida”, “rodovia”. Apenas as publicações contendo a localização geográfica do usuário foram selecionadas. Wang et

[1]http://lista10.org/tech-web/as-10-maiores-redes-sociais-do-mundo/

al. (2013) desenvolvem um algoritmo, baseado em modelo probabilístico de mistura Gaussiana, para a detecção de palavras que erupcionam repentinamente no Twitter. As palavras são obtidas através da interface "Streaming API".

3. Publicações de Twitter em Detecção de Eventos

Neste trabalho, o modelo de detector de eventos utiliza as publicações disponibilizadas pelo Twitter, com o intuito de obter as informações relevantes como, por exemplo, o local e o horário. O modelo busca publicações que contém uma palavra-chave e faz uma cópia para um ambiente de trabalho local, onde as frases são purificadas por termos relavantes, caracterizadas e alimentadas para uma máquina de aprendizado e sua peculiar classificação. Finalmente, os resultados positivos são preparados para as respectivas visualisações. Na Figura 1 se ilustra a operação descrita.

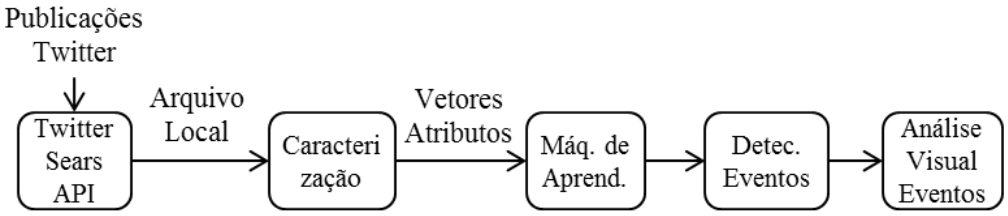

Figura 1: Processo de detecção de eventos.

3.1. Publicações como Fonte de Dados

No Twitter são criadas diariamente cerca de 500 milhões de publicações[2], também diversos dados úteis são gravados como horário de criação, idioma, dados do usuário que publicou (nome, localização, descrição e foto), geolocalização (para envios de smartphones, tablets e notebooks), referências a links e hashtags.

Para o sucesso da análise, grande parte das publicações é descartada como os erros de grafias, gírias, e conteúdos irrelevantes entre outros. Cerca de uns 40% das publicações são irrelevantes (Kelly, 2009). Para Sakaki et al (2010) os fatores que possibilitam a existência de un evento são: *escala*, *influência* e *região*. O *fator escala* se refere às vivencias de muitas pessoas que geram maior quantidade de publicações; a *influência* esta relacionada ao impacto do evento na vida das pessoas para compartilhar experiências; enquanto a *região* se refere ao espaço e tempo que permite a realização de estimativas e localização de eventos. Eventos com grande escala, alto grau de influência e região e tempo são os mais propícios para a aplicação da detecção de eventos.

3.2. Interfaces para Obtenção dos Dados

A notação JSON (Java Script Object Notation) é utilizada pelo serviço de interface de Twitter para troca de dados. Ela é uma formatação leve, de dado e valor, para facilitar transferência das publicações para externos. O Código 1 ilustra a estrutura JSON de uma publicação. Search API, da interface de Twitter, busca publicações por palavras chave, e devolve uma quantidade relevante de publicações, logo, é gravada localmente em um arquivo CSV (*comma separated values*), onde os dados textuais são

[2] www.internetlivestats.com/twiter-statistics/

representados entre aspas, separados por vírgulas definindo colunas, e as quebras em linhas de coluna.

Código 1: Estrutura JSON de uma publicação.

```
1  {
2  :created_at=> “ThuJul 31 15:14:27 +0000 2014”,
3  :id=> 494863667100258304,
4  :text=> “Manifestacao deixa transito congestionado em São Cristovao: Funcionarios do
   transporte alternativo protestam...”,
5  (...)
6  :user=>{
7  :id=>2340427167,
8  :name=> “Rodrigo”,
9  :location=> “Sao Paulo”,
10 (...)
11 },
12 :geo=>nil,
13 :coordinates=>nil,
14 :place=>nil,
15 (...),
16 :lang=> “pt”
17 }
```

3.3. Caracterização

Os dados dos arquivos CSV são convertidos em entidades numéricas para as respectivas operações de máquina de aprendizado. As informações de tamanhos variados são convertidas as entidades vetoriais de tamanho uniforme. Cada vetor representa uma publicação, as publicações selecionadas definem um espaço vetorial.

O recurso escolhido para este modelo é a presença ou ausência de palavras no *dicionário de termos*. Um termo é constituído de cadeia de caracteres limitada por espaços, gerado na tokenização. Estes termos passam por pré-processamento, que agrupa termos com mesmo significado, considerando palavras seguidas de pontuação e maiúsculas ou minúsculas. Todos os termos são agrupados e ordenados em ordem alfabética, criando assim o dicionário de termos. No dicionário, cada termo possui um único identificador definido por sua ordem. Então, a dimensão do vetor é definida pelo número de palavras no dicionário. Um vetor característico de uma publicação inicialmente é zero, indicando que todos os termos são ausentes, passando ser presente (dígito 1) a posição do vetor indicado pelo índice da posição do termo no dicionário. A sequência de caracterização descrita aqui é ilustrada pela Figura 2.

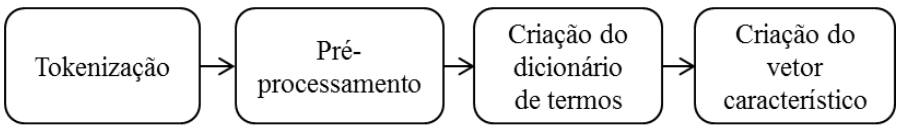

Figura 2: Processo de conversão de caracterização.

Para Turney e Pantel (2010), os tokenizadores devem ser capazes de reconhecer termos de mais de uma palavra, como “Lula da Silva”, contendo hífen e pontuações, ignorando pronomes e preposições. Neste trabalho um termo é delimitado por espaços, utilizando uma lista de palavras recorrentes para ignorá-las na criação de termos, assim como preposições “em”, “no”, “de” e outros. Também são ignorados os links começando com “http://” ou “www.”. Os termos são em minúsculas, sem caracteres especiais e pontuações, para que os termos iguais sejam essencialmente iguais. Este fato

permite diminuir a complexidade de análise e reduzir a dimensão do vetor característico.

O índice no dicionário, que corresponde a um termo da publicação, indica a posição desse termo no vetor, registrado com dígito 1 indicando a presença do termo, como mostra o exemplo: dado um dicionário de quatro termos, sequência de pares (índice, termo): (1,“acontecendo”); (2,“campos”); (3,“gosto”); (4,“manifestação”), e as mensagens A = (“manifestação”, “acontecendo”, “campos”) e B = (“gosto”, “manifestação”). Os respetivos vetores são Va = (1, 1, 0, 1) e Vb = (0, 0, 1, 1).

4. Máquina de Aprendizado

O método SVM foi desenvolvido por Vapnik nos anos 90 para tratar problemas de classificação de elementos distribuídos no espaço *d*-dimensional em classes relevantes e irrelevantes (Grigorik, 2008). O problema de otimização é formulado como, dado um conjunto de treinamento de *n* vetores característicos x_i de dimensão d e suas respectivas classes y_i, $D = \{(x_i, y_i)|x_i \in R^d, y_i \in \{-1,1\}\}_{i=1,\dots,n}$, achar um hiperplano P que divide os pontos que possuem $y_i = 1$, dos que possuem $y_i = -1$, da forma $w^T x + b = 0$. Sendo w o vetor normal do P, e b o deslocamento em relação à origem do sistema. O modelo é Minimizar $\|w\|$, Sujeito a $y_i(w^T x_i + b) \geq 1$, para $1 \leq i \leq n$. Podem existir casos em que os dados não sejam linearmente separáveis, porém existe uma margem muito pequena entre os vetores de suporte, que para margem um pouco maior deva se ignorado certos pontos como sendo de uma ou outra classe. O modelo é:

$$\text{Minimizar } \|w\| + C \sum \varepsilon_i \qquad (1)$$
$$\text{Sujeito a } y_i(w^T x_i + b) \geq 1 - \varepsilon_i, \text{ para } 1 \leq i \leq n.$$

Nessa formulação suave, são calculados os parâmetros w e b, tendo como entrada $\{(x_i, y_i)\}_{i=1,\dots,n}$ de treino, variando os possíveis valores de C. A verificação é feita com os dados de teste (x_i, y_i). Uma vez confirmado, possivelmente com uma margem de aproximação aceitável, o modelo será utilizado para a classificação dos eventos no formato vetor característico.

5. Implementação e Análise dos Resultados

A implementação é feita usando classes nativas da linguagem Ruby[3] e interfaces de comunicação com outros serviços. As publicações são obtidas por Search API do Twitter com palavra chave “manifestação”, entre 01 e 31 de agosto de 2014, em buscas semanais devido às políticas do Twitter na liberação das publicações. Para manipular os arquivos é utilizada a biblioteca CSV nativa do Ruby. Nas publicações de treino são adicionados os valores de classe “1” e “-1” considerando o formato (id, horário, texto, latitude, longitude, localização-perfil, classe). As classes String e Array do Ruby são aplicadas para as operações de tokenização, pré-processamento, criação do dicionário de termos e vetores de características.

[3] http://ruby-lang.org

5.1. Treino e Testes

A interface "rb-libsvm"[4] permite treinar, testar e usar um modelo de SVM. No treino, utilizam-se funções *Libsvm::Problem* e *Libsvm::Parameter*. A função *Libsvm::Problem* é encarregada de receber o conjunto de n descritores de treino $\{(x_i, y_i)\}_{i=1,\dots,n}$. Enquanto a função *Libsvm::Parameter* encapsula os ajustes dos parâmetros do SVM. Para o modelo desejado, apenas o parâmetro de custo C de (1) é ajustado, porém também é necessário inicializar os valores dos parâmetros ε_i (eps) e cache_size, como:

```
@parametro = Libsvm::SvmParameter.new
@parametro.cache_sezi = 1
@parametro.eps = 0.001
@parametro.c = 0.1  #cost
```

Para o treino, as publicações são lidas a partir do arquivo CSV como a seguir:

```
def carregador_publicacoes_treino caminho
 CSV.open(caminho) do |csv|
  csv.each do |linha|
    @publicacoes_treino<<Publicacao.new(linha,self)
  end
 end
end
```

A inserção dos descritores, os parâmetros no LIBSVM e o treino do classificador são realizados pelos métodos *Libsvm::problem.set_examples* e *Libsvm::Model.train*. O primeiro recebe apenas os descritores e os une na estrutura da interna da biblioteca, o segundo recebe esses dados em conjunto com os parâmetros e treino buscando a solução para (1), e obtendo os valores de w e b. Para saber se o modelo possui uma boa taxa de acertos ou não, deve ser testado com os respectivos valores de teste, utilizando o método *Libsvm::Problem.predict*. Caso a taxa de acertos der menor, deve-se treinar novamente variando os valores iniciais de C, tal como ilustrada pela Tabela 1, onde C = 0.1 deu um acerto de 90.5% com uma aceitável performance em relação aos valores.

Tabela 1: Taxa de acerto SVM.

C	Taxa de acerto	Performance
0.001	50%	9.4s
0.01	78.5%	9.3s
0.1	90.5%	8.0s
1	89.5%	7.9s
10	86.5%	7.6s
100	86.5%	7.8s

5.2. Resultados

O horário é extraído diretamente a partir da informação presente no arquivo CSV. Para a criação do ambiente foi utilizada a ferramenta para desenvolvimento de aplicações web Ruby on Rails4. No ambiente são exibidos gráficos de série-temporal das publicações, divididas primeiramente por dias, e posteriormente por horas.

Para analisar os eventos, no ambiente interativo (disponível em: http://deteccao.zangrandi.me/) são exibidos gráficos de séries-temporais das

[4] https://github.com/febeling/rb-libsvm

publicações, criados a partir da ferramenta *Highcharts*[5] e mapas de marcadores criados a partir de *TileMill*[6]. Na Figura 3(a) observa-se uma maior concentração de publicações no dia 20, e nos dias 23 e 24 houve uma baixa significativa, possivelmente corresponde ao sábado e domingo respectivamente, período que no Brasil ocorreram convocações dos últimos protestos sociais. No dia 20 (Figura 3 (b)), ao longo das horas, é possível ver os eventos de protesto enfatizados entre as 7 a 10 de manhã, tentando subir no final da tarde, sinal de que foi no início do dia aconteceu a convocatória.

A consulta no ponto de 6h-7h no dia 20 de agosto exibe os eventos distribuídos no território brasileiro (Figura 4). As publicações que não contém a geolocalização são aproximadas através do mapeamento das cidades para a sua geolocalização, de acordo com a informação do IBGE[7].

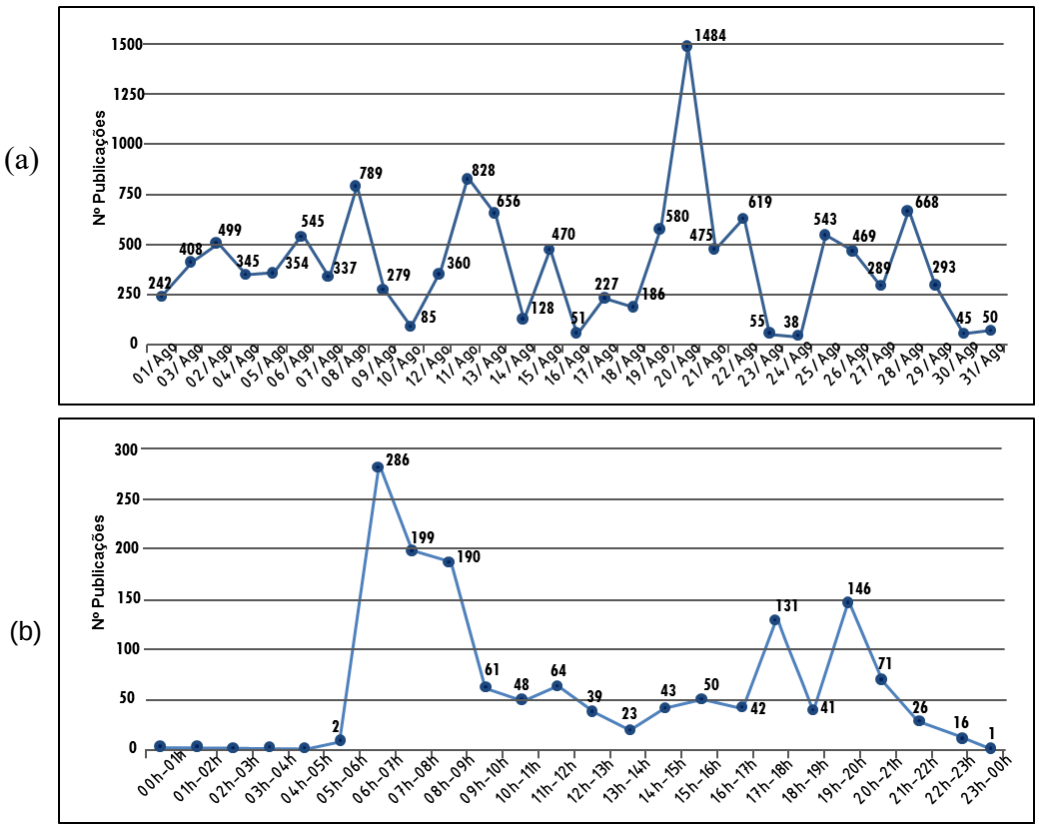

Figura 3: Mês de agosto de 2014 (a) e dia 20 por horas (b).

6. Conclusões e Trabalhos Futuros

A dinâmica de compartilhamento rápida de informações através de Twitter permite prever eventos de qualquer tipo em lugares específicos. Assím como o de Sakaki et al. (2010) que detecta terremotos em japão, Mai et al. (2013) que detecta acidentes de trânsito em locais de um território específico, o modelo implementado neste trabalho apresenta a possibilidade de detecção de manifestações no Brasil no período de convulsão social de 2014. Manifestação é um acontecimento social com certa escala, de

[5] http://www.highcharts.com
[6] https://www.mapbox.com/tilemill/
[7] http://ibge.gov.br

importância na vida das pessoas, com especificação de local e horário. É notório que os usuários ao tomarem conhecimento de uma manifestação, rapidamente publicam sobre ela, gerando assim picos de publicações que podem ser observados através dos gráficos de série-temporal.

Figura 4: Mapa de marcadores para horário entre 6h a 7h.

O modelo implementado ainda não consegue estimar de forma automática se, em determinado momento no tempo, está ocorrendo um evento ou não, ou seja, se a quantidade de publicações em determinada faixa de horário é normal ao funcionamento do sistema ou de fato anômala. Isso seria possível através de consideração de um modelo probabilístico, que se encarregaria de analisar a qual tipo de distribuição probabilística os dados se enquadram. Desse modo, poderia ser feito uma comparação entre a distribuição e os dados reais obtidos.

Referências

Abou-Zleikaha, M.; Tan, Z.; Christense, M. (2014) "Non-linguistic vocal event detection using online random forest", Information and Communication Technology, Electronics and Microelectronics, 37th International Convention on, p. 1326-1330.

Aiello, L.; Petkos, G.; Martin, C.; Corney, D.; Papadopoulos, S.; Skraba, R.; Goker, A. (2013) "Sensing Trending Topics in Twitter", IEEE Transactions on Multimedia, V 15, N. 6, p. 1268-1282.

Dong, X.; Mavroeidis, D.; Calabrese, F.; Frossard, P. (2014) "Multiscale Event Detection in Social Media", Journal Data Mining and Knowledge Discovery, p.1374-1405.

Fienberg, S. E.; Shmueli, G. (2005) "Statistical issues and challenges associated with rapid detection of bio-terrorist attacks". John Wiley and Sons.

Grigorik, I. (2008) “Support Vetor Machines (SVM) in Ruby”, https://www.igvita.com/2008/01/07/support-vector-machines-svm-in-ruby/.

Gupchup, J.; Terzis, A.; Burns, A.; Szalay, A. (2009) “Model-based event detection in wireless sensor networks”, CoRR abs/0901.3923.

Hong, J.; Liu, C. C.; Govindarasu, M. (2014) “Detection of cyber intrusions using network-based multicast messages for substation automation”, Innovative Smart Grid Technologies Conference (ISGT), IEEE PES. p. 1-5.

Ihler, A.; Hutchins, J.; Smyth,P. (2006) “Adaptive event detection with time-varying poisson processes”, Proceedings 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 207-216.

Java, A.; Song, A.; Finin, T. (2007) “Why we twitter: Understanding microblogging usage and communities”, Procedings of the Joint 9th WEBKDD, p. 56-65.

Kelly, R. (2009) “Twitter Study Reveals Interesting Results About Usage – 40is Pointless Babble”. http://www.pearanalytics.com/blog/2009/twiter-study-reveals-interesting-results-40-percent-pointless-babble/.

Mai, E.; Hranac, R. (2013) “Twitter interactions as data source for transportation incidents”, TRB 2013 Annual Meeting.

Matuszka, T.; Vinceller, Z.; Laki, S. (2013) “On a keyword-lifecycle model for real-time event detection in social network data”. 2013 IEEE 4th International Conference on Cognitive Infocommunications, p. 453-458.

Sakaki, T.; Okazami, M.; Matsuo, Y. (2010) “Earthquake shakes twitter users: Real-time event detection by social sensors”, WWW2010, pp. 851-860.

Takahashi, T.; Abe, S.; Igata, N. (2011) “Can twitter be an alternative of real-world sensors?”, Human-Computer Interaction, Part III, Springer-Verlag, p.240-249.

Turney, P.D; Pantel, P. (2010) “From frequency to meaning: Vector space models of semantics”, Journal of Artificial Interlligence Research, p. 141-188.

Wang, X.; Zhu, F.; Jiang, J.; Li, S. (2013) “Real time event detection in twitter”, 14th International Conference, WAIM, Springer-Verlag, p. 502-513.

Weng, J. e Lee, B.S. (2011) “Event detection in twitter”, Fifth International AAAI Conference on Weblogs and Social Media, Association for the Advancement of Artificial Inteligence.