UMA ABORDAGEM PERTURBATIVA APLICADA A MODELOS DE REGRESSÃO PARA LOCALIZAÇÃO EM REDES CELULARES

Por

TATIANA VIANA PADRÃO

Trabalho de Graduação

Universidade Federal de Pernambuco
graduacao@cin.ufpe.br
www.cin.ufpe.br/~graduacao

RECIFE/2018

Tatiana Viana Padrão

UMA ABORDAGEM PERTURBATIVA APLICADA A MODELOS DE REGRESSÃO PARA LOCALIZAÇÃO EM REDES CELULARES

Monografia apresentada ao Programa de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Orientador: Daniel Carvalho de Cunha

RECIFE
2018

Monografia submetida por **Tatiana Viana Padrão** ao programa de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **Uma abordagem perturbativa aplicada a modelos de regressão para localização em redes celulares**, orientada pelo **Prof. Daniel Carvalho de Cunha** e aprovada pela banca examinadora formada pelos professores:

Prof. Daniel Carvalho de Cunha
Centro de Informática/UFPE

Prof. Paulo Salgado Gomes de Mattos Neto
Centro de Informática/UFPE

RECIFE
2018

Agradecimentos

Primeiramente, eu agradeço à minha família, principalmente aos meus pais, pelo o apoio, paciência e incentivo durante toda a minha vida acadêmica. Agradeço a todos os meus amigos do CIn que me aguentaram durante o curso e fizeram tudo valer mais a pena e ser mais divertido. Agradeço aos meus amigos de fora da faculdade também, principalmente minhas amigas do Equipe, que sempre estiveram ao meu lado e me deram apoio em todos os aspectos desde a infância. Além disso, agradeço a todos os professores que ao longo da minha vida acadêmica me ensinaram não só como ser uma boa profissional mas também a crescer como ser humano. Por fim, devo um agradecimento especial a Robson Timoteo, Marcel Santana, Walber Macedo e o meu orientador Daniel Cunha por todo o apoio, seja emocional ou técnico, que foi imprescindível para a finalização desse trabalho.

Resumo

Com a disseminação de redes e aparelhos móveis, além de aplicações baseadas em localização, há uma demanda por métodos cada vez melhores de localização de dispositivos dentro dessas redes. Além disso, o paradigma da Internet das Coisas trouxe mais uma demanda para essa área, exigindo métodos eficientes energeticamente, o que se alinha bem com o uso de sinais de radiofrequência. Nesse contexto, um dos possíveis métodos de localização se baseia no uso de regressores para encontrar as coordenadas geográficas do dispositivo móvel de forma direta. Considerando que esses modelos de regressão são sub-ótimos, esse projeto propõe que o erro desses modelos seja aprendido, e então os resultados sejam combinados para gerar resultados de maior acurácia, baseando-se na ideia da teoria da perturbação. O objetivo do trabalho é, então, implementar a técnica mencionada anteriormente e compará-la com a técnica baseada em regressão direta simples em termos de acurácia e custo computacional. Apesar dos resultados do método proposto terem apresentado pequenas melhoras em acurácia, foi demonstrado que as técnicas comparadas não são diferentes de uma forma estatisticamente relevante e que a proposta tem um maior custo computacional. Assim, a técnica desenvolvida nesse trabalho não parece apresentar nenhuma vantagem significativa.

Palavras-chave: Comunicações móveis, sistema de posicionamento, teoria da perturbação, aprendizagem de máquina

Abstract

The advance of mobile networks and devices, in addition to location-based applications, imply on a demand for increasingly better methods of locating devices within these networks. Besides that, the Internet of Things paradigm has brought more demand to this area, requiring energy-efficient methods, which aligns well with the use of radio frequency signals. In this context, one of the possible localization methods is based on the use of regressors to find the geographical coordinates of the mobile device directly. Considering that these regression models are sub-optimal, this project proposes that these models' errors can be learned, and then the results might be combined to produce more accurate results, based on the idea of perturbation theory. The goal of this work is then to implement the previously mentioned technique and compare it with the technique based on simple direct regression in terms of accuracy and computational cost. Although the results of the proposed method presented small improvements in accuracy, it was demonstrated that the comparative techniques are not different in a statistically relevant way and that the proposal has a higher computational cost. Thus, the technique developed in this work does not seem to present any significant advantage.

Keywords: Mobile communications, positioning systems, perturbation theory, machine learning

Sumário

1 Introdução **7**
1.1 Objetivos 8
1.2 Estrutura do Documento 8

2 Referencial Teórico **9**
2.1 Localização baseada em Radiofrequência 9
2.2 Teoria da Perturbação 10
2.3 Aprendizagem de Máquina 11
2.3.1 *k-Nearest Neighbors* 12
2.3.2 *Support Vector Machine* 12
2.3.3 Funções de Kernel 14
2.3.4 Support Vector Regression 15

3 Metodologia e Resultados **17**
3.1 Base de dados 17
3.2 Técnica proposta 18
3.2.1 Fase Offline - Obtenção dos Erros Residuais 19
3.2.2 Fase Offline - Treinamento dos Modelos Finais 22
3.3 Resultados 23

4 Conclusão **30**

Referências **32**

1

Introdução

A quantidade de aparelhos móveis, sendo eles celulares, tablets ou até sensores, além do tráfego de dados em redes sem fio vêm crescendo bastante ao longo dos últimos anos. Além disso, a Internet das Coisas vem se estabelecendo como um paradigma que reforça ainda mais a importância das tecnologias sem fio (MIORANDI; SICARI; CHLAMTAC, 2012). Nesse contexto, aplicações com serviços baseados em localização mostram-se cada vez mais importantes e presentes (BARNES; WINTERBOTTOM; DAWSON, 2011), sendo necessárias tecnologias que possam localizar dispositivos de forma eficiente e com a maior acurácia possível (KJAERGAARD, 2012).

No caso da localização *outdoor*, o GPS (sistema de posicionamento global) já é uma forma de localização bem estabelecida e com acurácia suficiente. Apesar disso, essa técnica não se alinha bem com uma das demandas mais importantes da Internet das Coisas que é a eficiência energética (WANG et al., 2014). Internet das Coisas engloba redes grandes de sensores, dispositivos pequenos que têm que ter alta performance energética e baixo custo, e para esses requisitos GPS não é a melhor opção, pelo menos não pra ser usada em todos os nós da rede (BULUSU; HEIDERMANN; ESTRIN, 2000). Assim, localização baseada em sinais de radiofrequência é uma alternativa mais apropriada, e a forma de realizar localização por esse meio é um tópico relevante de pesquisa.

Existem vários métodos de localização baseados em radiofrequência, cada um com suas vantagens e desvantagens, porém nesse trabalho focamos principalmente naqueles que utilizam métodos de aprendizagem de máquina, em particular o uso da regressão de forma direta para encontrar as coordenadas geográficas dos dispositivos.

Aprendizagem de máquina abrange algoritmos capazes de extrair padrões a partir de dados, fazendo predições com alta acurácia. Essa técnica já foi aplicada a essa área de estudos de diferentes formas (TIMOTEO et al., 2016; BRUNATO; BATTITI, 2005; WANG et al., 2016). Uma das formas em que essa técnica foi utilizada no posicionamento em redes móveis foi através da previsão direta das latitudes e longitudes, usando-se um modelo para cada, e atingindo bons resultados (OLIVEIRA et al., 2018).

Dessa forma, esse trabalho propõe uma extensão do método de regressão direta para localização, baseando-se na teoria da perturbação. Essa teoria busca explicar sistemas reais

complexos através da combinação de sistemas mais simples, que converge para uma aproximação da solução alvo. Supondo que os modelos desenvolvidos na regressão direta não já sejam a solução ótima do problema, a aplicação de termos de correção do erro preditos a partir dos modelos de predição da solução poderia levar a uma maior acurácia.

Até onde sabemos, não há outros trabalhos utilizando essa abordagem de combinação iterativa com a predição do erro para melhorar os resultados de regressão direta nesse contexto. Assim, a proposta desse projeto é investigar a abordagem perturbativa aplicada à regressão direta para melhorar resultados de localização de dispositivos móveis.

1.1 Objetivos

O principal objetivo desse trabalho é desenvolver uma técnica de localização em redes celulares utilizando-se de uma abordagem perturbativa e então, comparar essa nova forma com métodos já existentes. A técnica desenvolvida propõe estender o método já existente de regressão direta para encontrar latitudes e longitudes (OLIVEIRA et al., 2018), fazendo-se uma combinação do regressor base com um regressor treinado nos seus erros residuais, promovendo uma correção incremental. A teoria das perturbações defende que a aplicação de termos perturbativos, baseados em sistemas mais simples, a uma solução inicial podem fazê-la convergir para representar a solução ótima de um problema mais complexo (BENDER; ORSZAG, 2013). Como esses termos perturbativos tendem a influenciar sucessivamente menos na solução final do problema, esse trabalho concentrou-se no uso de uma iteração de correção.

Dessa forma, as técnicas que pretendemos avaliar são duas variações da abordagem proposta: regressão base combinada com regressão do erro usando *k-Nearest Neighbors* (*k*-NN), e regressão base usando *k*-NN combinada com regressão do erro utilizando *Support Vector Regression* (SVR). Essas duas variações serão comparadas com a regressão direta simples, fazendo-se análises de acurácia e custo computacional.

Para atingir esse objetivo, é necessário utilizar uma base de dados, que foi obtida de forma experimental, sendo medidas 2756 amostras de localização e força de sinal para um dispositivo móvel através de um *drive-test* em um ambiente urbano de Recife-PE. Além disso, é necessário ajustar os modelos utilizados para encontrar os parâmetros ótimos para cada um deles, obtendo-se os melhores resultados possíveis. Por fim, terão que ser utilizadas as métricas apropriadas para fazer as comparações propostas.

1.2 Estrutura do Documento

No Capítulo 2, é apresentado o referencial teórico do projeto, sendo abordados conceitos importantes para o desenvolvimento da solução, além da abordagem dos principais trabalhos com soluções relacionadas ao tema. No Capítulo 3, é explicada a metodologia utilizada e os resultados obtidos são mostrados e analisados. Por fim, no Capítulo 4 as conclusões são apresentadas.

2

Referencial Teórico

Neste Capítulo, são apresentados os conceitos importantes e que servem de base para este trabalho. Técnicas semelhantes à proposta são descritas, além de ser feita uma revisão dos conceitos de aprendizagem de máquina e teoria da perturbação necessários.

2.1 Localização baseada em Radiofrequência

A área de pesquisa de localização em redes sem fio é uma área bastante estudada, devido à sua importância no contexto atual de disseminação de dispositivos móveis e Internet das Coisas, como foi mencionado no Capítulo 1.

As técnicas de localização podem ser classificadas como *indoor* e *outdoor*. No caso da localização *indoor*, um dos maiores desafios são os obstáculos que atenuam ou desviam a propagação dos sinais. Já no caso da localização *outdoor*, apesar do GPS fornecer uma boa acurácia, esta técnica não atende a requisitos de eficiência energética, por exemplo, demandados por toda uma classe de problemas de Internet das Coisas. Nesse contexto, nota-se a importância dos métodos de localização baseados em radiofrequência.

Existem algumas técnicas diferentes de posicionamento baseadas em sinais de rádio. Uma delas é a trilateração (LI et al., 2017; ONALAJA; ADJRAD; GHAVAMI, 2014), que a partir da localização e sinal recebido de pontos fixos, calcula as distâncias desses pontos ao ponto que se deseja localizar, sendo realizado posteriormente um método de otimização para determinar a localização desejada, que geralmente é um processo custoso.

Uma outra forma de localizar dispositivos móveis é a técnica de *fingerprinting* (VO; DE, 2016). Ela consiste de duas fases: offline e online. A primeira fase é responsável pela construção de um mapa de *fingerprints* de sinais, utilizando-se de medições reais, modelos teóricos e modelos de aprendizagem de máquina (TIMOTEO et al., 2016; TIMOTEO; CUNHA; CAVALCANTI, 2014). A segunda fase é aquela em que o *fingerprint* correspondente ao móvel é comparado com aqueles do mapa, buscando-se o que melhor se aproxima, através de métricas como distância euclidiana. Assim, a posição do *fingerprint* mais similar ao do móvel vai corresponder à sua posição predita. Uma das desvantagens dessa técnica é que a fase online de busca é altamente custosa, principalmente quando considerada em grande escala (HE; TAN; CHAN, 2016).

A forma de localização na qual esse trabalho se baseia é a de regressão direta. Essa técnica consiste do uso de métodos de predição para obter diretamente as coordenadas geográficas dos dispositivos, a partir dos sinais recebidos das estações rádio base (ERBs), por exemplo. Nesse contexto, há dois trabalhos que devem ser mencionados. Um que realiza essa regressão utilizando regressão linear múltipla (MLR) (EZEMA; ANI, 2016) e um que usa técnicas mais atuais de aprendizagem (OLIVEIRA et al., 2018). O segundo, mais recente, utiliza *k-Nearest Neighbors* e *Support Vector Regression* para achar modelos para previsão da latitude e longitude de forma separada, combinando depois essas duas medidas para estimar a posição do móvel. Assim, esse trabalho serviu como uma das principais bases para o desenvolvimento deste.

2.2 Teoria da Perturbação

A teoria da perturbação é um conceito de aproximação de soluções bastante utilizado em diversas áreas como matemática e astronomia (GALLAVOTTI, 2007). A ideia é que um problema complexo possa ser representado como a soma de termos mais simples, aproximando a solução desejada.

Considera-se que, inicialmente, temos a solução exata de um problema 1 relacionado a um problema 2 mais complexo, que é o problema para o qual realmente queremos encontrar uma solução. Assim, de acordo com a teoria, somando sucessivos termos, as perturbações, podemos fazer a solução original convergir assintoticamente para a solução desejada. A ideia é que esses termos reflitam a diferença entre os dois problemas.

Assim, considerando que temos uma solução P_0 inicial para um problema, esperamos encontrar a solução P desejada através da adição dos termos perturbativos, como mostrado na equação 2.1 (BENDER; ORSZAG, 2013):

$$P = \varepsilon^0 P_0 + \varepsilon^1 P_1 + ... + \varepsilon^n P_n \tag{2.1}$$

em que o termo ε é um coeficiente de valor menor que 1, que determina a contribuição de cada termo para a solução final. Dessa forma, pode-se notar que o primeiro termo (P_0) é o mais importante e as perturbações contribuem sucessivamente menos para o resultado final.

Como modelos preditivos são geralmente sub-ótimos, particularmente quando considerada a complexidade de problemas do mundo real, a ideia é que essa teoria possa ser aplicada a um problema de predição, melhorando-o iterativamente. Com essa premissa, essa abordagem já foi aplicada na área de predição de séries temporais (NETO et al., 2017; MATTOS NETO et al., 2010) de forma que P_0 é considerada a predição inicial da série e os termos perturbativos são correções baseadas nas medidas de erro residuais do modelo atual. Por fim, a soma de todos esses resultados se aproxima assintoticamente da solução ótima para a série.

Baseando-se nessa premissa, decidiu-se testar essa abordagem no contexto de localização de dispositivos móveis em redes celulares, concentrando-se em uma iteração, devido ao fato de

que sucessivas iterações tendem a contribuir cada vez menos para o resultado final.

2.3 Aprendizagem de Máquina

Esta seção é dedicada a explicar conceitos básicos da técnica de aprendizagem de máquina, focando nos algoritmos usados nesse trabalho. Existem tarefas que não podem ser programadas para que um computador execute de forma determinística, assim aprendizagem de máquina são algoritmos capazes de aprender com dados crus, extraindo informações e modelos a partir deles (HARRINGTON, 2012).

A aprendizagem de máquina pode ser de dois tipos: supervisionada e não-supervisionada. O primeiro assume que o aprendizado envolve dados com valores alvo definidos, assim o objetivo do algoritmo é achar padrões nas variáveis de entrada para prever as variáveis de saída. Já no segundo, não existe um valor alvo buscado, o algoritmo busca apenas padrões e aspectos em comum nos dados de entrada. Os algoritmos utilizados nesse projeto são métodos de aprendizado supervisionado.

Na área do aprendizado supervisionado, há dois casos possíveis. O primeiro ocorre quando a variável de saída buscada é discreta, categorias em que os dados estão separados, tratando-se de uma classificação. O segundo caso é quando a variável buscada está dentro de um conjunto de espectro contínuo de valores, havendo infinitas possibilidades, esse caso é chamado de regressão. Nosso caso de estudo busca valores contínuos, tratando-se de uma regressão.

Muitas vezes, os parâmetros desses algoritmos não podem ser determinados diretamente a partir da análise dos dados. Assim, faz-se uma busca para determinar a melhor escolha desses parâmetros. Uma das formas de busca, que é a usada nesse projeto, consiste em definir conjuntos de possíveis valores para esses parâmetros, aplicar esses parâmetros ao modelo e definir qual a combinação ótima de parâmetros baseado em alguma métrica de acurácia. Esse método é chamado de uma busca em *grid* (KUHN; JOHNSON, 2013).

É válido mencionar que há dois problemas que podem acontecer quando se treina um modelo: subajuste e sobreajuste. O primeiro trata-se de quando o modelo final não fica flexível o suficiente para refletir os relacionamentos reais entre os dados. No segundo, a situação é a oposta, pois é quando o modelo se adapta demais aos casos de treino, aprendendo padrões dos dados que não são reproduzíveis, os ruídos únicos de cada amostra. Nesses dois casos, os modelos tendem a ter performances de baixa acurácia. Um exemplo de situação que pode levar tanto a um problema quanto ao outro é a escolha dos parâmetros do algoritmo, pois como muitas vezes esses parâmetros estão associados à complexidade do modelo, más escolhas podem levar a treinamentos especificados demais ou não suficientemente.

As Subseções seguintes focam em explicar os algoritmos utilizados nas regressões desse projeto: *k*-NN e SVR.

2.3.1 *k-Nearest Neighbors*

k-Nearest Neighbors (*k*-Vizinhos mais próximos) ou *k*-NN é um algoritmo de aprendizagem de máquina que tem treinamento baseado em instância, ou seja, o treino consiste apenas em armazenar os exemplos de dado em memória, que então vão ser usados apenas quando chega uma nova instância para ser testada. Esse tipo de avaliação costuma ser chamada de preguiçosa, já que o processamento dos dados é adiado até que se tenha uma consulta para ser avaliada. Uma vantagem desse tipo de treinamento é que a avaliação dos dados é feita de forma local, diferente para cada consulta, em vez de ser criado um modelo que se encaixa ao espaço de dados de forma geral. Já uma desvantagem é que o custo computacional de cada consulta pode ser muito alto, dependendo do tamanho da base de dados de treinamento (MITCHELL et al., 1997).

A cada consulta de um caso de teste $\mathbf{x}_a$, são encontradas as k amostras da base de treino mais próximas dela, representando os vizinhos. O resultado então é definido a partir dos resultados desses vizinhos. No caso de classsificação, a resposta é a classe mais frequente entre as classes dos vizinhos, como um esquema de voto, já no caso da regressão, usa-se a média das respostas das k instâncias selecionadas. O algoritmo 1 descreve o funcionamento do modelo descrito para o caso de regressão.

Algorithm 1 Descrição do *k*-NN para regressão

Treinamento:
    Salve os dados de treino $(\mathbf{x}, f(\mathbf{x}))$ *em memória como dados_treino.*
Predição:
    Para cada instância de teste $\mathbf{x}_a$*:*
        Encontre os $\mathbf{x}_1 \ldots \mathbf{x}_k$ *vizinhos mais próximos de* $\mathbf{x}_a$ *entre as instâncias em dados_treino*
        Retorne: $\hat{f}(\mathbf{x}_a) \leftarrow \sum_{i=1}^{k} \frac{f(x_i)}{k}$

Dessa forma, a ideia principal do *k*-NN depende de como essa distância entre as instâncias é definida. Considerando a instância de teste $\mathbf{x}_a$, a sua distância d para o vizinho $\mathbf{x}_b$ pode ser definida através da distância de Minkowski mostrada na equação (2.2).

$$d(\mathbf{x}_a, \mathbf{x}_b) = \left(\sum_{i=1}^{n} |x_{ai} - y_{bi}|^q \right)^{1/q}, \quad onde \quad q > 0 \tag{2.2}$$

Essa equação generaliza dois dos tipos mais comuns de distância que são a Manhattan e a Euclidiana, para as quais q é igual a 1 e 2, respectivamente.

2.3.2 *Support Vector Machine*

As máquinas de vetores de suporte (SVM) constituem uma técnica de aprendizagem de máquina que foi introduzida em 1995 (VAPNIK, 1995). Esse método consiste de um aprendizado supervisionado que busca achar um hiperplano ótimo que separe os dados de entrada.

Como pode ser observado na parte esquerda da Figura 2.1, reduzindo o problema ao plano 2D para visualização, poderão existir inifinitos separadores dos dados que atendem ao requisito de separar as diferentes classes de dados com a melhor acurácia possível. Assim, precisa-se de um critério para determinar qual desses possíveis divisores seria o divisor ótimo. A ideia usada é, então, a de que o quanto mais afastado um ponto está do limiar de divisão, mais confiança se tem sobre a previsão feita (HARRINGTON, 2012). Dessa forma, tenta-se achar o plano ótimo como aquele com a maior distância dele para o ponto mais próximo. Essa distância é chamada de margem, enquanto os pontos sobre a linha dessa margem são os vetores de suporte. Esses conceitos são ilustrados na parte direita da Figura 2.1.

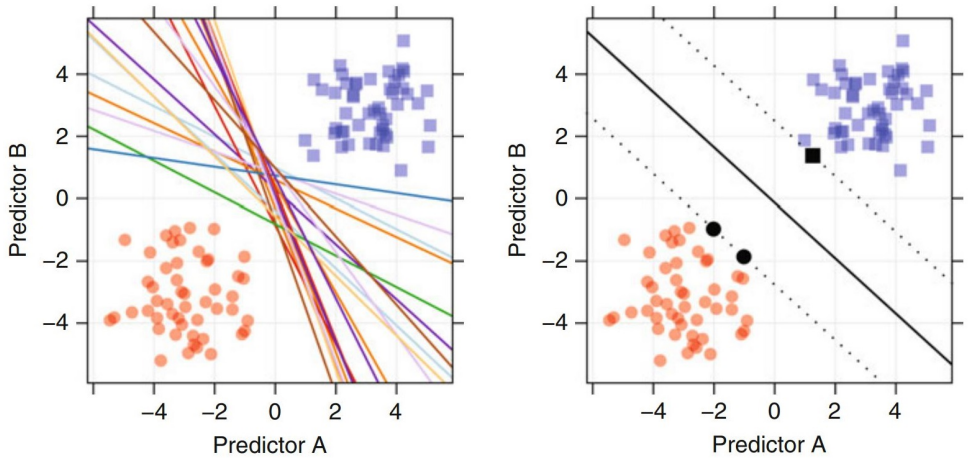

Figura 2.1: Esquerda: Conjunto de dados linearmente separável com várias possíveis divisões lineares que levariam à mesma performance de acurácia. Direita: Divisor ótimo associado com a menor margem. Pontos pretos indicam os vetores de suporte. Fonte: Applied predictive modeling, Kuhn e Johnson, 2013, p.344

Um conjunto de dados pode ser linearmente ou não-linearmente separável. O primeiro significa que o conjunto poderia ser separado em suas classes por uma linha, como é o caso da Figura 2.1. Já na segunda situação, as classes dos dados estão envolvidas de forma que uma linha não seria suficiente para separá-las. Primeiramente, vamos lidar com o caso linearmente separável, estendendo na Subseção 2.3.3 para o caso não-linearmente separável.

No caso linear, a classificação pode ser feita por meio de uma função real, tal que quando $f(\mathbf{x}) > 0$, a entrada $\mathbf{x} = (x_1, ..., x_n)$ é tida como pertencente a uma classe positiva, e quando contrário a uma classe negativa. Essa função é linear e definida pela equação (2.3), onde $\mathbf{w}$ e b, variáveis ajustadas na aprendizagem, são o vetor de pesos e o *bias*, respectivamente (LIMA, 2002), e podem ser observados melhor na Figura 2.2.

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \tag{2.3}$$

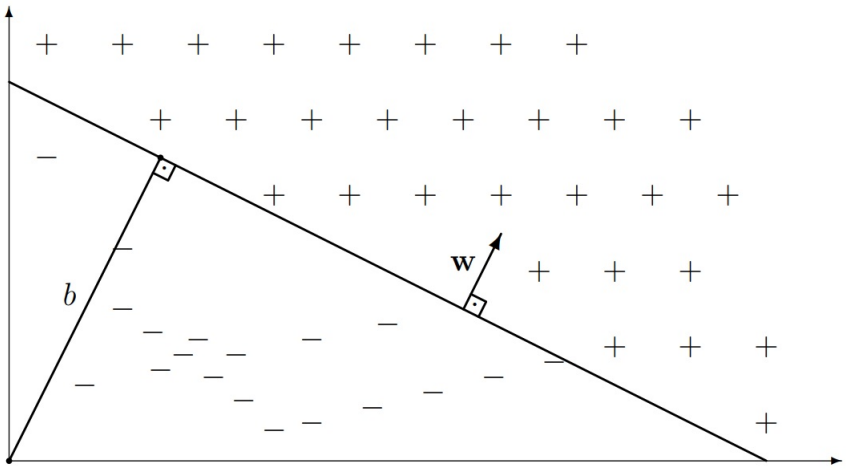

Figura 2.2: Vetor **w** e valor b em hiperplano que divide os conjuntos de dados de duas classes linearmente separáveis. Fonte: Máquinas de Vetores Suporte na Classiflcação de Impressões Digitais, Lima, 2002, p.48

Considerando que a margem tenha tamanho igual a 1 em ambas as direções, ou seja, a equação (2.4) seja satisfeita, então podemos concluir que o tamanho total da margem seria $\frac{2}{\|\mathbf{w}\|}$. Como o objetivo é maximizar a margem, o nosso problema de otimização passa a ser o de minimizar $\|\mathbf{w}\|$, que pode ser resolvido através do método clássico de multiplicadores de Lagrange (LIMA, 2002).

$$\min_{i=1,\ldots,n} |\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b| = 1 \tag{2.4}$$

Nas próximas Subseções, serão discutidas as funções de kernel, que permitem que a solução seja aplicada a problemas não-linearmente separáveis, e concluiremos apresentando a aplicação de SVM em regressão, que é a natureza do problema sendo estudado nesse trabalho.

2.3.3 Funções de Kernel

Problemas reais muitas vezes não vão poder ser linearmente separados, porém isso não quer dizer que não haja padrões a serem reconhecidos nesses dados, então precisamos apenas transformá-los de forma que o algoritmo descrito anteriormente possa reconhecer esses padrões (HARRINGTON, 2012).

Assim, uma solução é o mapeamento dos dados em um outro espaço no qual a sua distribuição possa ser aprendida de forma mais fácil pelo modelo desenvolvido. Esse mapeamento ϕ pode ser observado na Figura 2.3. Nela, podemos notar que o conjunto de dados da esquerda não poderia ser linearmente separado, porém depois do mapeamento no espaço da direita, os dados agora podem ser facilmente separados por uma linha.

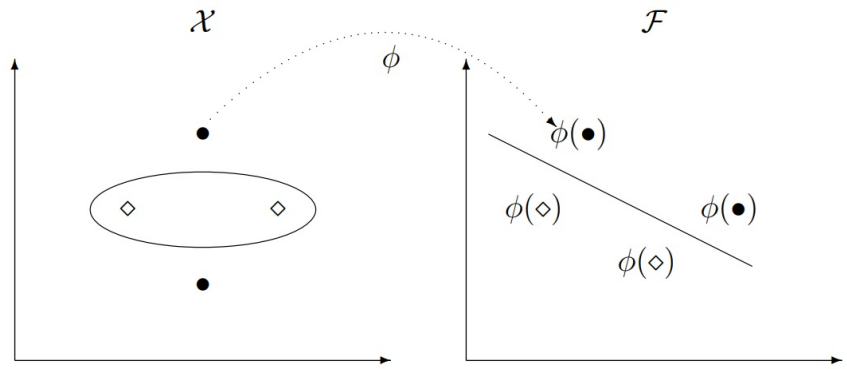

Figura 2.3: Mapeamento de um conjunto de dados X não-linearmente separável em um conjunto de dados F linearmente separável. Fonte: Máquinas de Vetores Suporte na Classiflcação de Impressões Digitais, Lima, 2002, p.48

Esses mapeamentos entre espaços ocorrem usualmente para espaços de maior dimensão, trazendo o problema de se trabalhar com dimensões muito grandes quando o mapeamento é feito de forma explícita para cada entrada. Assim, surge a necessidade das funções de kernel, equação (2.5), que realizam esse mapeamento de forma implícita e solucionam esse problema de lidar com dimensões altas demais (LIMA, 2002):

$$K(\mathbf{x},\mathbf{z}) = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle \tag{2.5}$$

em que $\langle \cdot \rangle$ representa o produto interno entre vetores e $\phi(\cdot)$ representa a função de mapeamento entre os espaços. Nesse projeto, escolhemos usar uma das funções de kernel mais famosas, *Radial Basis Function* ou RBF, que introduz o parâmetro σ a mais para ser ajustado no modelo.

2.3.4 Support Vector Regression

O uso de SVM em regressão, proposto em (VAPNIK, 1995), pode ser definido por diferentes algoritmos. O tipo de SVR escolhido nesse trabalho foi o ε-SVR, que busca encontrar uma solução f que dê resultados com erros menores que um limiar ε para os casos de treino. Para isso, tenta-se achar o vetor $\mathbf{w}$ que minimize o erro e tenha a menor norma possível, evitando sobreajuste (SMOLA; SCHÖLKOPF, 2004).

Esse problema de otimização pode ser descrito pelas equações (2.6) e (2.7), onde os termos extras ξ_i, ξ_i^* são folgas adicionadas ao problema. Quando definimos o problema de otimização acima para encontrar a função f, assumimos que existia tal função que atendia aos requisitos do problema, o que pode não ser verdade, logo esses elementos de folga surgem para lidar com restrições inviáveis.

$$\frac{1}{2}\|x\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \tag{2.6}$$

$$\begin{cases} y_i - \langle w, x_i\rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i\rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{2.7}$$

O custo C na equação (2.7) equilibra o quanto os erros influenciam na otimização, servindo de penalização aos erros. Assim, esse comportamento corresponde à função ε-*insensitive* e pode ser melhor observado na Figura 2.4. Pode-se notar, então, que pontos dentro do limiar ε não contribuem para o treino, e os demais contribuem de acordo com uma escala linear (KUHN; JOHNSON, 2013).

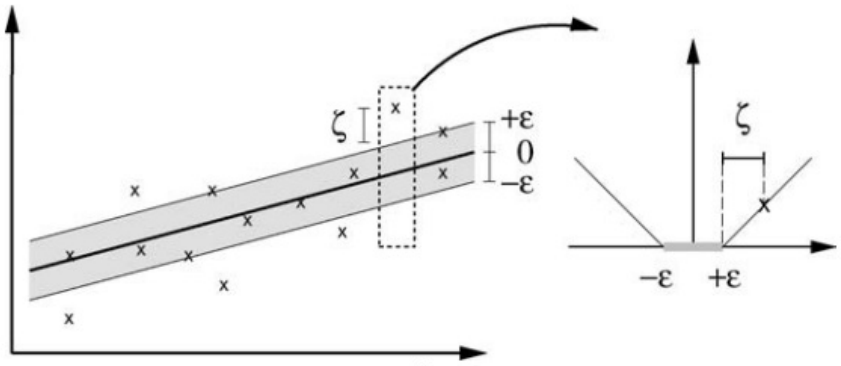

Figura 2.4: Comportamento da função ε-*insensitive*. Fonte: A tutorial on support vector regression, Smola e Scholkopf, 2004, p.2

3

Metodologia e Resultados

Neste capítulo, é apresentada a metodologia adotada no desenvolvimento dos experimentos, assim como os resultados encontrados. Primeiramente, a base de dados utilizada e o procedimento para obtê-la são descritos. Em seguida, é explicado em mais detalhes o treinamento dos modelos de aprendizagem e o algoritmo de combinação da correção de erro para determinação da localização. Por fim, são apresentados os resultados obtidos.

3.1 Base de dados

Para obtenção da base de dados usada, foi feito um *drive-test*, em que foi medida a força do sinal recebido em um ambiente urbano localizado em Recife-PE com uma área de aproximadamente $1,4km^2$. Para isso, foram consideradas medidas de propagação de ondas de rádio na banda de frequência de 1,8GHz GSM (*Global System for Mobile Communications*). As medidas foram capturadas com a ferramenta NEMO FSR1, um receptor de varredura digital que faz medições precisas de sinais de radiofrequência em redes sem fio. Foram feitas 2956 medições, das quais 2756 foram usadas para treino dos modelos de predição e 200 foram usadas para teste e obtenção dos resultados finais. Na Figura 3.1, pode-se observar a distribuição desses pontos medidos em um mapa do local onde o teste foi realizado, com os pontos usados para treino em verde e os pontos usados para teste em vermelho.

Duas variáveis foram observadas em relação às 6 Estações Rádio Base (ERBs), cujas posições também estão indicadas na Figura 3.1: o nível de potência do sinal recebido em *dBm* (*received signal strength indicator*, RSSI) e o *timing advance* (TA). O TA corresponde a um atraso do sinal transmitido, que era usado para evitar sobreposições de sinais (KLOZAR; JAN, 2012). Ele é representado como um número de 0 a 63, onde cada valor implica em um passo incremental de 550 metros de distância do ponto em relação à ERB. O estudo mais recente do uso de regressão direta para localização mostrou melhores resultados quando o TA era empregado como variável no modelo de regressão (OLIVEIRA et al., 2018), mostrando sua importância como parâmetro no posicionamento em redes celulares. Dessa forma, a base de dados é formada por essas duas medidas (6 de cada pela quantidade de ERBs) e posições espaciais representadas em latitudes e longitudes.

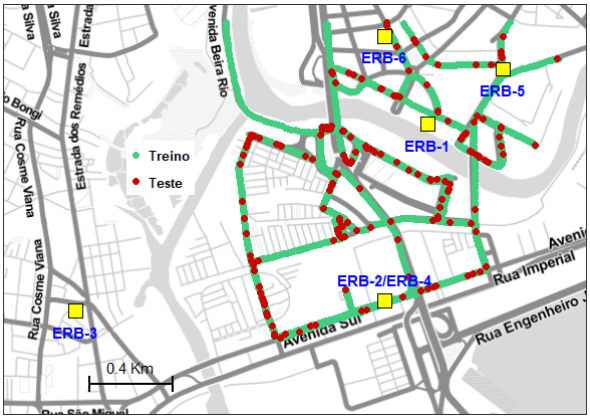

Figura 3.1: Ambiente urbano de Recife-PE com indicações das posições onde foram feitas as medidas utilizadas como base de dados para treino e teste do método proposto, assim como as localizações das Estações Rádio Base (ERBs).

3.2 Técnica proposta

A técnica proposta é um método de localização que busca encontrar de forma direta as latitudes e longitudes dos dispositivos móveis. Esse tipo de técnica já foi explorado anteriormente, como mencionado na Seção 2.1, porém até onde sabemos, nunca foi associado a uma abordagem perturbativa. A ideia de trazer uma abordagem perturbativa vem do fato de geralmente os modelos de aprendizagem para problemas do mundo real não serem ótimos, pois estes problemas carregam uma complexidade grande. Dessa forma, se aplicarmos essa abordagem, poderemos nos aproximar da solução exata do problema através da aplicação iterativa de correções baseadas no erro residual do modelo atual (NETO et al., 2017; MATTOS NETO et al., 2010).

Na teoria perturbativa, a contribuição dos termos ou perturbações vai sucessivamente contribuindo menos para a solução final do problema, já que a soma das perturbações têm a propriedade de convergir para a solução desejada, como explicado na Seção 2.2. Assim, a técnica proposta se limitou a uma iteração de correção do erro, já que o resultado dessa iteração já deveria permitir concluir a respeito da relevância e eficácia dessa abordagem aplicada ao problema de localização.

A solução desenvolvida consiste, então, de uma regressão base combinada com uma regressão do erro para encontrar os valores de latitude e longitude a partir das variáveis RSSI e TA. A proposta está detalhada no Algoritmo 2. A solução pode ser dividida em duas fases: offline e online. A primeira consiste dos primeiros 4 passos, enquanto a segunda abrange os demais.

A fase offline concentra-se, então, na obtenção dos dados e treino dos modelos. O primeiro passo consiste na obtenção dos dados de treino e teste mencionados na Seção anterior. O segundo foca em construir a base de dados com os erros residuais provenientes do modelo de regressão base. Maiores detalhes sobre esse processo são discutidos na Subseção 3.2.1. Os

Algorithm 2 Descrição do método de localização proposto

1: *Coletar medidas de RSSI e TA em uma quantidade suficiente de pontos;*
2: *Construir base de dados com os erros residuais baseados no modelo de regressão base;*
3: *Treinar modelo de regressão base, encontrando as funções* $f_1(\cdot)$ *e* $f_2(\cdot)$*;*
4: *Treinar modelo de regressão do erro, encontrando as funções* $g_1(\cdot)$ *e* $g_2(\cdot)$*;*
5: *Dados os RSSI e TA dos dispositivos da base de dados de teste, prever a latitude e longitude usando* $f_1(\cdot)$ *e* $f_2(\cdot)$*;*
6: *Dados os RSSI e TA dos dispositivos da base de dados de teste, prever os erros de latitude e longitude usando* $g_1(\cdot)$ *e* $g_2(\cdot)$*;*
7: *Estimar a posição do dispositivo através da combinação das latitudes e longitudes preditas e os seus respectivos erros;*

passos 3 e 4, por fim, treinam os modelos de regressão base e de erro, e uma explicação mais aprofundada desse estágio é feito na Subseção 3.2.2. A fase online, por sua vez, executa a estimação final da posição do dispositivo móvel. Essa estimação é feita através da soma das predições feitas pelos regressores de base e erro para as duas coordenadas, latitude e longitude, obtendo-se, assim, as coordenadas estimadas do móvel. O processo descrito anteriormente pode ser observado esquematicamente na Figura 3.2, onde as fases offline e online são apresentadas em forma de diagrama.

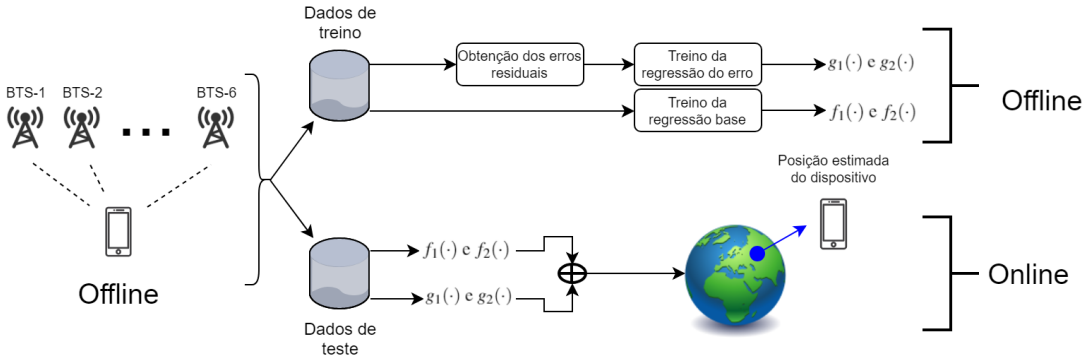

Figura 3.2: Diagrama da técnica proposta. Fase offline: Coleta das amostras, obtenção dos erros residuais e treinamento dos modelos de predição. Fase online: Predição da posição do dispositivo.

3.2.1 Fase Offline - Obtenção dos Erros Residuais

A construção de uma base de dados com os erros residuais requer que o modelo de regressão base seja treinado com algum conjunto de dados. Depois, esse modelo deve ser usado para prever resultados de localização em alguma base de dados. Finalmente, os erros desses resultados devem ser calculados, formando o conjunto de dados que é usado para treinar o modelo de predição do erro. Existe mais de uma possível forma que esse processo pode ser executado:

1. Usar toda a base de dados de treino tanto na hora do treino do regressor base, quanto na hora da predição dos resultados para o cálculo do erro. Nesse caso, os mesmos dados usados no treinamento são usados na predição, o que pode deixar os resultados tendenciosos, influenciando nos resultados finais obtidos;

2. Separar a base de dados de treino em dois subconjuntos, de forma que um deles seja usado para o treino e o outro para a predição. Nessa abordagem, os dados usados no treino e na predição são diferentes, eliminando a preocupação com resultados tendenciosos. Porém, nesse caso surge uma nova preocupação quanto ao tamanho final dos conjuntos de dados usados nos treinos, pois a quantidade de dados pode influenciar no treino e, consequentemente, nos resultados, então o ideal seria que os conjuntos de treino fossem do tamanho da base de dados de treino total, 2756 amostras, ou pelo menos o mais perto disso possível;

3. Montar o conjunto de dados dos erros baseado na técnica de *stacking*, o que elimina ambas as preocupações existentes nas abordagens anteriores, motivo pelo qual essa foi a técnica escolhida.

Stacking é uma técnica de combinação de modelos preditivos diferentes para formar um novo modelo (DŽEROSKI; ŽENKO, 2004; BREIMAN, 1996). No nível base, são usados um ou mais modelos para fazer previsões sobre os dados, que então vão ser usadas como variáveis no próximo nível, chamado de meta nível. A ideia é que nos casos em que o modelo 1 tem uma performance ótima, o modelo 2 possa ter uma performance pior, porém no resto dos casos, pode ser que o modelo 2 tenha uma melhor performance do que o modelo 1. Dessa forma, a abordagem de *stacking* tende a ressaltar um modelo base nos casos em que ele tem melhor performance e a desconsiderá-lo nos casos em que a sua performance não é ideal.

Uma das técnicas de *stacking* para construir o conjunto de dados com as previsões dos modelos base é dividir a base de dados de treino original em k conjuntos disjuntos de dados, *folds*, de tamanhos aproximadamente iguais e usar validação cruzada (REFAEILZADEH; TANG; LIU, 2009). Essa validação consiste em k iterações, onde cada vez $k-1$ *folds* são usados para treino e o *fold* restante é usado na validação, construindo-se o conjunto de dados de previsão de forma iterativa.

Tomando como base a técnica descrita, o conjunto de treino é dividido em 4 *folds*, porém sem o uso de reamostragem como costuma ser o caso de validação cruzada. A cada iteração, 3 *folds* são usados para treino e o restante é usado para prever os resultados que são comparados com os dados reais. Assim, o conjunto de erros residuais é construído de forma iterativa, fazendo-se a cada iteração que os resultados preditos X_P sejam subtraídos dos reais X_R para obter-se o erro X_E, como mostrado na Figura 3.3.

Para o treino do modelo de regressão base usado na construção do conjunto de erros residuais, escolhemos utilizar o k-NN. Esse algoritmo de aprendizagem foi o que demonstrou

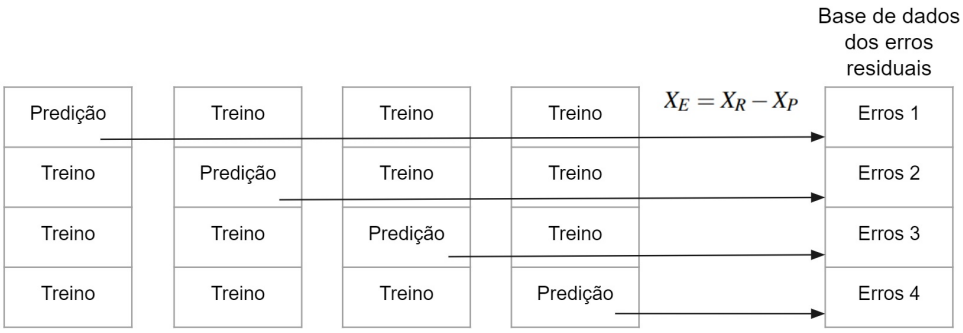

Figura 3.3: Ilustração da técnica utilizada para contrução do conjunto de dados dos erros residuais, onde X_P são os resultados preditos, X_R os valores reais e X_E os erros residuais.

melhores resultados no estudo mais recente de predição de localização através do uso de regressão direta (OLIVEIRA et al., 2018). Dessa forma, para cada iteração, o modelo escolhido é ajustado para encontrar os melhores parâmetros. Uma vez que esses parâmetros tenham sido escolhidos, os modelos podem ser treinados e os resultados de predição e erro podem ser obtidos.

Antes de realizar os treinos e predições nos conjuntos de dados, foram aplicadas técnicas de pré-processamento, os dados das variáveis foram centralizados e escalados (KUHN; JOHNSON, 2013). Para centralizar, a média é subtraída de cada valor das variáveis, de forma que cada uma fique com média zero. Já para escalar, cada valor é dividido pelo desvio padrão, resultando em um desvio padrão final unitário. Esses procedimentos são importantes para a performance final dos modelos. Tomando como exemplo o caso do *k*-NN, se uma das variáveis do modelo estiver numa escala muito maior do que as outras, essa variável poderá influenciar de forma injusta o cálculo das distâncias entre as amostras, logo um pré-processamento dos dados como o mencionado pode garantir que cada variável influencia os cálculos de forma igual, evitando um potencial viés.

O ajuste dos parâmetros é feito através do uso de validação cruzada *k-fold*. Como já mencionado, esse princípio consiste da divisão do conjunto de dados em *k folds*, sendo feitas *k* iterações onde um *fold* é validado por vez. Dessa forma, foi escolhido $k = 10$ e a métrica escolhida para selecionar o melhor conjunto de parâmetros foi a raiz do erro quadrático médio (RMSE), que pode ser observado na equação (3.1):

$$\sqrt{\frac{1}{n}\ \Sigma_{i=1}^{n}(x_i)^2}. \tag{3.1}$$

Os parâmetros ajustáveis do *k*-NN são a quantidade de vizinhos *k*, a função usada no cálculo da distância e a função de kernel (HECHENBICHLER; SCHLIEP, 2004). Para a função de kernel, utilizamos kernel retangular, o que significa que os vizinhos não têm pesos associados a eles baseados em suas distâncias ao ponto sendo testado, de forma que todos eles contribuem

igualmente para o resultado. O parâmetro k possui influência direta no custo computacional do algoritmo, pois determina quantos vizinhos vão ser considerados em cada predição. Com o aumento do k, o problema pode ser subajustado devido à influência de vizinhos muito distantes do caso sendo testado. Já a diminuição do k pode levar ao sobreajuste, já que estará acontecendo um ajuste altamente localizado (KUHN; JOHNSON, 2013). No projeto, foram considerados os valores de k como os números naturais ímpares de 1 a 33. Já no caso da função de distância, foram testadas na distância de Minkowski o q com os valores 1, distância de Manhattan, e 2, distância Euclidiana.

Usando esse processo de calibragem dos parâmetros dos modelos, os melhores valores encontrados para os 4 modelos usados podem ser vistos na Tabela 3.1. Assim, cada modelo pode ser treinado com esses parâmetros e usado para predição no *fold* de validação correspondente. Com essa predição, estão associados erros residuais, que incrementalmente compõem o conjunto de dados de erro buscado.

Fold		k	q
1	Lat.	3	2
	Lon.	33	2
2	Lat.	3	2
	Lon.	33	1
3	Lat.	3	1
	Lon.	3	2
4	Lat.	3	1
	Lon.	33	1

Tabela 3.1: Parâmetros ótimos encontrados para o treinamento do k-NN para cada iteração na construção da base de dados dos erros residuais

3.2.2 Fase Offline - Treinamento dos Modelos Finais

A segunda parte da fase offline consiste do treinamento dos modelos finais usados na solução. Dessa forma, é necessário o treinamento de um regressor para a predição base e de um regressor para a predição do erro. Primeiramente, os conjuntos de dados para cada um desses treinamentos já são conhecidos. A base de dados usados para o regressor base é todo o

conjunto de treino adquirido experimentalmente como descrito na Seção 3.1. Já a base utilizada no regressor do erro é a que é construída na primeira parte da fase offline, segundo o processo explicado na Subseção anterior. Nos dois casos as entradas são as mesmas, o RSSI e o TA em relação a cada ERB, mudando apenas a saída dos modelos. As saídas são as coordenadas geográficas para a regressão base e os fatores de correção dessas coordenadas para a regressão do erro.

Para o regressor base, o algoritmo de aprendizagem escolhido foi o *k*-NN pelos mesmos motivos mencionados anteriormente, pois esse foi o modelo que demonstrou melhor resultado em pesquisa similar e recente (OLIVEIRA et al., 2018). Já para o regressor do erro, dois algoritmos diferentes foram testados, o *k*-NN e o SVR. Esse dois modelos foram escolhidos devido à sua aplicação recorrente em problemas de localização em redes móveis (TIMOTEO et al., 2016; BRUNATO; BATTITI, 2005; WANG et al., 2016).

Primeiramente, as bases de dados são submetidas a um pré-processamento, sendo centralizadas e escaladas, assim como explicado na descrição da construção da base de dados de erro. Em seguida, os parâmetros de cada regressor são buscados através do método de validação cruzada *k-fold*, usando-se $k = 10$ e RMSE como métrica para determinar os parâmetros ótimos.

Os parâmetros ajustáveis do *k*-NN são os mencionados na Subseção 3.2.1, e aqui utilizamos os mesmos espaços de busca. O k é buscado entre os números ímpares entre 1 e 33, as distâncias entre Manhattan e Euclidiana e o kernel é considerado o retangular em todas as situações. Já no caso do SVR, os parâmetros configuráveis são C, ε e o kernel. O kernel foi considerado o RBF laplaciano com otimização do σ de acordo com (TIMOTEO et al., 2016). O custo C influencia na flexibilidade do modelo, logo um custo maior implica em maior flexibilidade e diminuição do erro, porém pode levar a sobreajuste, já a diminuição do custo leva ao cenário contrário, deixando o modelo mais propício ao subajuste (KUHN; JOHNSON, 2013). O parâmetro ε se comporta da forma oposta, uma vez que menores valores levam a redução do erro e possibilidade de sobreajuste, enquanto que maiores valores podem levar ao subajuste, já que haveria uma diminuição dos vetores de suporte. Neste trabalho, consideramos os valores de C como as potências de 2 entre -2 e 14, e os valores de ε foram escolhidos entre 0,05 e 0,1.

Dessa forma, o modelo de regressão base e os de regressão do erro são ajustados para que se encontre os valores ótimos para cada um dos parâmetros, e o resultado desses ajustes podem ser encontrados nas Tabelas 3.2, 3.3 e 3.4, que correspondem respectivamente aos parâmetros do regressor base, regressor do erro com *k*-NN e regressor do erro com SVR. Uma vez que esses parâmetros estão definidos, cada modelo pode ser treinado, estando pronto para fase online.

3.3 Resultados

Como abordado no Capítulo 1, o objetivo desse trabalho é analisar uma abordagem perturbativa aplicada ao cenário de localização de dispositivos móveis em redes sem fio. Para isso, o propósito é comparar a técnica de regressão direta já existente, só o *k*-NN como feito

	k	q
Lat.	3	1
Lon.	5	1

Tabela 3.2: Parâmetros ótimos encontrados para o treinamento do k-NN para a predição base

	k	q
Lat.	33	1
Lon.	33	1

Tabela 3.3: Parâmetros ótimos encontrados para o treinamento do k-NN para a predição do erro

	C	σ	ε
Lat.	0,25	0.08404642	0,05
Lon.	0,25	0.08153518	0,05

Tabela 3.4: Parâmetros ótimos encontrados para o treinamento do SVR para a predição do erro

em (OLIVEIRA et al., 2018), com essa técnica combinada a uma correção de um regressor baseado em seus erros residuais. Dessa forma, nesta Seção essas duas abordagens, regressão direta simples e com correção do erro, são comparadas quanto à sua acurácia e complexidade computacional, sendo mostrados aqui os principais resultados obtidos.

As três técnicas sendo comparadas são, então, uma de regressão direta simples e duas de regressão direta combinada com correção, sendo cada uma delas com um modelo diferente para a regressão do erro. Quando a técnica se trata da regressão direta simples, nos referimos a ela nos Gráficos e Tabelas apenas pelo modelo usado na regressão, que no caso é k-NN. Já nas técnicas com correção do erro, usamos a forma $modelo_{base} + modelo_{erro}$, dessa forma temos KNN-KNN e KNN-SVR.

Para realizar a comparação entre as técnicas, definimos uma métrica de erro η, que corresponde à distância em metros entre o ponto predito e o ponto real. Assim, algumas análises

estatísticas sobre esses erros foram feitas para cada uma das técnicas de localização consideradas, como pode ser visto na Tabela 3.5. Nesta tabela, pode-se observar 4 parâmetros: a média ($\bar{\eta}$), o desvio padrão (η_σ), o valor mínimo (η_{min}) e o valor máximo (η_{max}). Os melhores resultados para cada um dos parâmetros estão em negrito, onde os três primeiros desses melhores resultados são referentes à técnica KNN-SVR.

	KNN	KNN-KNN	KNN-SVR
$\bar{\eta}$	34.9053	35.0773	**34.5978**
η_σ	33.3888	32.8997	**33.3852**
η_{min}	0.5542	0.9116	**0.1646**
η_{max}	**195.8777**	217.7366	196.9813

Tabela 3.5: Análise estatística dos erros η (em metros) obtidos nos casos de teste para cada uma das técnicas de localização consideradas neste projeto

As variações do resultado de erro η podem também ser analisadas através da Figura 3.4. Esse gráfico representa os dados de forma que as arestas inferiores e superiores das caixas indicam, respectivamente, o primeiro e terceiro quartil, enquanto que a linha que divide cada caixa representa a mediana dos dados. Já as linhas que se estendem na vertical representam os erros que não estão entre o Q_1 e Q_3. Por fim, os pontos isolados representam valores discrepantes ou *outliers*. Observando-se o gráfico, nota-se uma semelhança entre as técnicas analisadas.

Outras duas visualizações dos resultados são mostradas nas Figuras 3.5 e 3.6. Na primeira delas, é mostrado um histograma para valores de erro η de cada uma das abordagens, de forma que no eixo x está esse erro em metros e no eixo y está a contagem de amostras que têm esse valor como erro. Já na segunda, os pontos preditos, vermelho, verde e azul, foram plotados num mapa do local real de coleta das amostras para comparação com os pontos reais coletados, que foram desenhados de cinza. Observando esses dois resultados, chega-se à mesma conclusão do gráfico mencionado anteriormente, as técnicas propostas parecem semelhantes à técnica de regressão direta simples, apesar das pequenas melhoras observadas na Tabela 3.5 para a técnica KNN-SVR. Porém, não podemos concluir se os resultados são realmente diferentes ou não só olhando para as visualizações e a tabela.

Nesse contexto, o teste de Friedman é aplicado aos resultados para que possamos tirar conclusões a respeito de se a diferença entre os resultados das técnicas estudadas são estatisticamente relevantes. O teste de Friedman é um teste não paramétrico em que a hipótese nula H_0 é a de que os conjuntos de dados são equivalentes, e a hipótese alternativa H_1 afirma que há alguma diferença entre pelo menos dois dos conjuntos (DEMŠAR, 2006). O teste fornece um

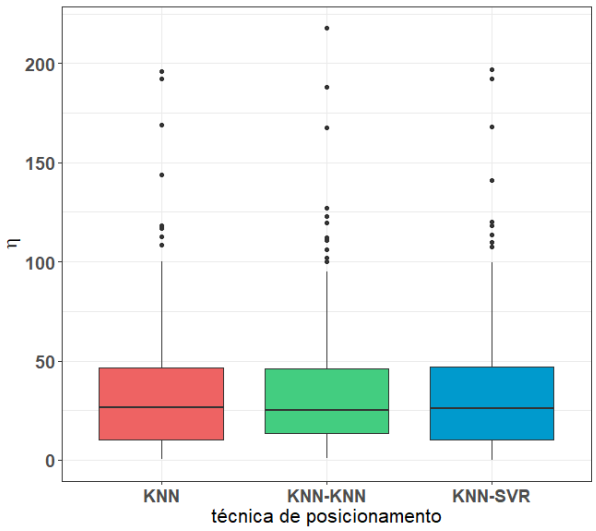

Figura 3.4: Gráfico *boxplot* para os erros η obtidos nos casos de teste para cada uma das técnicas de localização consideradas.

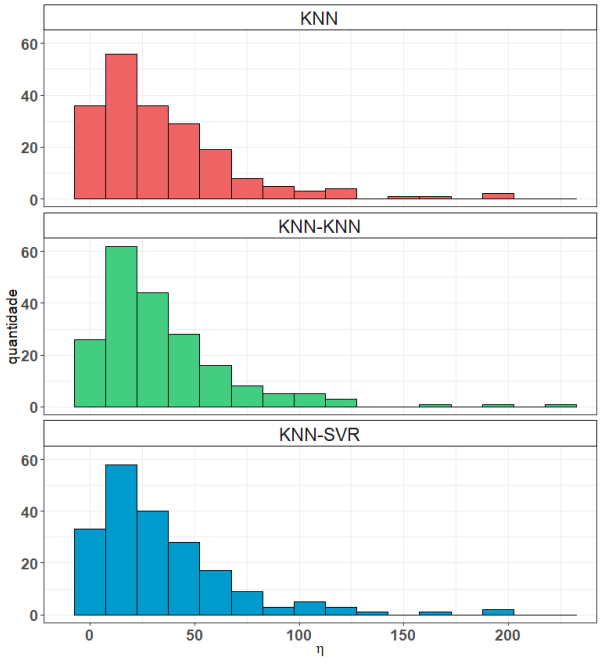

Figura 3.5: Gráfico dos histogramas dos erros η (em metros) obtidos nos dados de teste para cada uma das técnicas de localização consideradas.

valor p que corresponde à probabilidade de rejeitar a hipótese nula quando ela é verdade. Caso esse valor p seja menor do que um valor de significância α, geralmente tomado com o valor 0,05, podemos rejeitar a hipótese nula com segurança e aceitar a alternativa de que há diferença entre os conjuntos. Dado que há diferença entre pelo menos dois dos conjuntos analisados, resta saber entre quais. Para isso, há o teste *post-hoc* de Nemenyi, que realiza uma comparação par a

Figura 3.6: Mapas com os pontos reais e os preditos nos casos de teste para cada uma das técnicas de localização consideradas.

par, gerando valores p para cada um desses pares, que então são interpretados da mesma forma.

Para analisar se as técnicas são diferentes de forma relevante, os testes estatísticos mencionados são realizados para as latitudes e longitudes finais obtidas por cada abordagem, assim como para os erros η. Para a latitude, o valor de p é 0,8737, o que nos permite concluir que não podemos afirmar que alguma das técnicas é diferente das outras na obtenção das latitudes. Já no caso da longitude, o valor de p é 0,005887, demonstrando que há pelo menos duas técnicas diferentes, então prosseguimos para o teste de Nemenyi, obtendo os resultados mostrados na Tabela 3.6. Observando-se esses resultados, pode-se concluir que a hipótese nula pode ser rejeitada para os pares KNN/KNN-KNN e KNN/KNN-SVR, logo as longitudes obtidas através das técnicas que envolvem a predição do erro são diferentes das obtidas pela técnica de regressão direta simples.

Embora haja evidência de diferença na longitude, a realização do teste de Friedman para os erros η resulta em um valor p igual a 0,09633. A métrica η é a que mais importa, pois apesar das técnicas desenvolvidas estarem prevendo valores de latitude e longitude, o problema em questão consiste de um passo de abstração a mais depois dessas previsões, em que essas latitudes e longitudes são combinadas para formar posições. Dessa forma, uma acurácia de latitude ou longitude não nos importa se não for acompanhada de uma acurácia de posicionamento. Assim,

podemos concluir que não há uma diferença estatisticamente relevante entre as técnicas de combinação do erro propostas e a técnica de regressão direta simples.

	KNN	KNN-KNN
KNN-KNN	0,016	-
KNN-SVR	0,014	0,999

Tabela 3.6: Comparação em pares das longitudes obtidas nos casos de teste para cada uma das abordagens usando o teste *post-hoc* de Nemenyi

A segunda parte da comparação que tínhamos como objetivo nesse trabalho é a de custo computacional. Para fazer essa análise, supomos que os parâmetros para cada um dos modelos já foram escolhidos, tal que não é mais necessário o ajuste deles. As técnicas são divididas em dois processos: treino e predição, onde cada uma já consiste de todos os modelos envolvidos e necessários para a técnica em questão. Como as técnicas que contêm a predição do erro têm uma tarefa a mais na sua fase offline, a de construir a base de dados de erros residuais, esse processo também será incluso na comparação de custo computacional, sendo divido da mesma forma em treino e predição. Para cada um dos processos analisados, foram executados 100 experimentos, sendo assumido o mesmo ambiente computacional para todos eles. Na Tabela 3.7 são mostrados os tempos médios resultantes desses experimentos normalizados pela maior medição.

	KNN	KNN-KNN	KNN-SVR	Dados de erro
Treino	0,4101	0,8449	1,0000	0,8620
Predição	0,0242	0,0486	0,0481	0,1062

Tabela 3.7: Análise de custo computacional médio normalizado pelo maior valor

Para confirmar que os tempos obtidos para cada uma das estratégias são diferentes de um ponto de vista estatístico, o teste de Friedman é aplicado a cada um dos conjuntos de tempos, treino e predição. Para os dois, o valor de p obtido é menor que 2,2e-16, logo pode-se concluir que existem pelo menos duas estratégias com tempos que são diferentes em cada um dos casos. Assim, aplicamos em seguida o teste *post-hoc* de Nemenyi, comparando os processos par a par. Os resultados obtidos nessa segunda iteração de teste podem ser vistos nas Tabelas 3.8 e 3.9,

onde a primeira é referente aos tempos de treino e a segunda aos tempos de predição. Analisando os valores de p calculados, pode-se notar que todos eles estão abaixo do nível de significância α, logo os custos computacionais dos processos analisados são todos diferentes entre si.

	KNN	KNN-KNN	KNN-SVR
KNN-KNN	2e-08	-	-
KNN-SVR	< 2e-16	4e-14	-
Dados de erro	4e-14	2,5e-05	2e-08

Tabela 3.8: Comparação em pares dos custos computacionais de treino usando o teste *post-hoc* de Nemenyi

	KNN	KNN-KNN	KNN-SVR
KNN-KNN	3,4e-14	-	-
KNN-SVR	1e-08	6,9e-05	-
Dados de erro	< 2e-16	1e-08	3,4e-14

Tabela 3.9: Comparação em pares dos custos computacionais de predição usando o teste *post-hoc* de Nemenyi

Uma vez que provamos que os custos computacionais calculados são diferentes de forma estatisticamente relevante, podemos notar que as técnicas que envolvem a correção do erro têm uma complexidade computacional maior. Além de seus custos para o treino e predição já serem aproximadamente o dobro dos custos para a regressão simples, essas técnicas ainda têm a parte extra de construção da base de dados de erro na fase offline.

4

Conclusão

No contexto atual de Internet das Coisas, há cada vez mais dispositivos móveis conectados em redes sem fio e utilizando aplicações que precisam fazer uso de localização. Dessa forma, esse trabalho propôs uma nova técnica de posicionamento baseada em radiofrequência que tem como base a teoria da perturbação. Essa teoria busca explicar sistemas complexos da vida real em termos de sistemas mais simples, de forma que essa combinação chegue numa convergência que descreva de forma ótima o sistema que se buscava entender. Assim, tomando como base aplicações dessa teoria, o método proposto usa a predição do erro do regressor base para ajustar o problema, como um fator de correção, o que poderia ser aplicado iterativamente, mas nesse trabalho decidiu-se concentrar os esforços em uma iteração, já que de acordo com a teoria perturbativa a contribuição dos termos perturbativos para o problema diminui sucessivamente. Assim, o objetivo era comparar essa nova abordagem com a abordagen existente de regressão direta, fazendo um estudo sobre as diferenças em acurácia e custo computacional.

Os experimentos foram realizados para 3 técnicas: regressão direta simples usando KNN, regressão base combinada com regressão do erro usando KNN para os dois, e regressão base usando KNN combinada com regressão do erro usando SVR. As duas últimas são do método proposto, enquanto a primeira é baseada no estudo mais recente e parecido na área. Usando os resultados obtidos dessas 3 abordagens, foram feitas visualizações de dados e análises estatísticas que buscavam comparar principalmente os erros em distância dos pontos preditos aos pontos reais nos casos de teste. Essas comparações foram inconclusivas, pois apesar de pequenas melhoras conseguidas na média e desvio padrão dos erros, todas as visualizações dos dados pareciam indicar uma grande semelhança entre os resultados.

Dessa forma, foram realizados testes para descobrir sobre a relevância estatística das diferenças entre os resultados. Apesar de os resultados de longitude das técnicas que usavam a correção terem sido provados diferentes dos obtidos por regressão direta simples, os resultados da latitude e do erro em distância, a métrica mais significativa, não poderam ser provados diferentes. Assim, essa técnica como abordada nesse trabalho não parece trazer nenhum ganho relevante em acurácia para a solução do problema. Além disso, através de repetidos experimentos, foi feita a comparação entre o custo computacional dos métodos abordados, mostrando-se que as técnicas com abordagem perturbativa têm uma maior complexidade computacional.

Para trabalhos futuros modelos diferentes poderiam ser testados na regressão do erro para a latitude, já que essa medida não pareceu mostrar resultados diferentes, o que poderia influenciar numa mudança mais significativa na acurácia das posições finais. Além disso, nesse trabalho a regressão base e do erro são combinadas na proporção 1:1, de forma que poderiam ser testados algoritmos que buscassem a proporção ótima para essa combinação, como MLP ou algoritmos genéticos. Essa técnica introduziria uma maior complexidade, mas que poderia ser compensada por uma melhora em acurácia.

Referências

BARNES, R.; WINTERBOTTOM, J.; DAWSON, M. Internet geolocation and location-based services. **IEEE Communications Magazine**, v.49, p.102–108, 2011.

BENDER, C. M.; ORSZAG, S. A. **Advanced mathematical methods for scientists and engineers I**: asymptotic methods and perturbation theory. Springer Science & Business Media, 2013.

BREIMAN, L. Stacked regressions. **Machine learning**, v.24, n.1, p.49–64, 1996.

BRUNATO, M.; BATTITI, R. Statistical learning theory for location fingerprinting in wireless LANs. **Computer Networks**, v.47, n.6, p.825–845, 2005.

BULUSU, N.; HEIDERMANN, J.; ESTRIN, D. GPS-less low cost outdoor localization for very small devices. **IEEE Personal Communications**, v.7, p.28–34, 2000.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. **The Journal of Machine Learning Research**, v.7, p.1–30, 2006.

DŽEROSKI, S.; ŽENKO, B. Is combining classifiers with stacking better than selecting the best one? **Machine learning**, v.54, n.3, p.255–273, 2004.

EZEMA, L. S.; ANI, C. I. Multi linear regression model for mobile location estimation in GSM network. **Indian Journal of Science and Technology**, v.9, p.1–6, 2016.

GALLAVOTTI, G. Perturbation theory. , 2007.

HARRINGTON, P. **Machine Learning in Action**. Greenwich, CT, USA: Manning Publications Co., 2012.

HE, S.; TAN, J.; CHAN, S.-H. G. Towards area classification for large-scale fingerprint-based system. In: ACM INTERNATIONAL JOINT CONFERENCE ON PERVASIVE AND UBIQUITOUS COMPUTING, 2016. **Proceedings...** 2016. p.232–243.

HECHENBICHLER, K.; SCHLIEP, K. Weighted k-nearest-neighbor techniques and ordinal classification. , 2004.

KJAERGAARD, M. Location-based services on mobile phones: minimizing power consumption. **IEEE Pervasive Computing**, v.11, p.67–73, 2012.

KLOZAR, L.; JAN, P. Wireless network localization: optimization processing. In: INTERNATIONAL CONFERENCE ON DIGITAL TELECOMMUNICATIONS (ICDT 2012), 7. **Proceedings...** 2012. p.45–49.

KUHN, M.; JOHNSON, K. **Applied predictive modeling**. Springer, 2013. v.26.

LI, J. et al. A novel robust trilateration method applied to ultra-wide bandwidth location systems. **Sensors**, v.17, n.4, p.795, 2017.

LIMA, A. R. G. **Máquinas de Vetores Suporte na Classificação de Impressões Digitais**. 2002. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal do Ceará.

MATTOS NETO, P. S. de et al. An intelligent perturbative approach for the time series forecasting problem. In: NEURAL NETWORKS (IJCNN), THE 2010 INTERNATIONAL JOINT CONFERENCE ON. **Anais...** 2010. p.1–8.

MIORANDI, D.; SICARI, S.; CHLAMTAC, I. Internet of Things: vision, applications and research challenges. **Ad Hoc Networks**, v.10, p.1497–1516, 2012.

MITCHELL, T. M. et al. **Machine learning. WCB**. McGraw-Hill Boston, MA:, 1997.

NETO, P. S. M. et al. A perturbative approach for enhancing the performance of time series forecasting. **Neural Networks**, v.88, p.114–124, 2017.

OLIVEIRA, L. L. et al. An RSS-based regression model for user equipment location in cellular networks using machine learning. **Springer Wireless Networks**, v.24, p.1–10, 2018.

ONALAJA, O.; ADJRAD, M.; GHAVAMI, M. Ultra-wideband-based multilateration technique for indoor localisation. **IET Communications**, v.8, n.10, p.1800–1809, 2014.

REFAEILZADEH, P.; TANG, L.; LIU, H. Cross-validation. In: **Encyclopedia of database systems**. Springer, 2009. p.532–538.

SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. **Statistics and computing**, v.14, n.3, p.199–222, 2004.

TIMOTEO, R. D. A. et al. An approach using support vector regression for mobile location in cellular Networks. **Computer Networks**, v.95, p.51–61, 2016.

TIMOTEO, R. D.; CUNHA, D. C.; CAVALCANTI, G. D. A proposal for path loss prediction in urban environments using support vector regression. In: ADVANCED INT. CONF. TELECOMMUN. **Proceedings...** 2014. p.1–5.

VAPNIK, V. N. **The Nature of Statistical Learning Theory**. Berlin, Heidelberg: Springer-Verlag, 1995.

VO, Q. D.; DE, P. A survey of fingerprint-based outdoor localization. **IEEE Communications Surveys and Tutorials**, v.18, p.491–506, 2016.

WANG, L. et al. Energy efficiency on location based applications in mobile cloud computing: a survey. **Computing**, v.96, p.569–585, 2014.

WANG, Q. et al. IWKNN: an effective bluetooth positioning method based on isomap and wknn. **Mobile Information Systems**, v.2016, 2016.