Universidade Federal do Rio de Janeiro

Processos Gaussianos para Estimação de Dados Ausentes de Tráfego

Douglas Picciani de Souza

Rio de Janeiro

Agosto de 2017

Universidade Federal do Rio de Janeiro

Processos Gaussianos para Estimação de Dados Ausentes de Tráfego

Douglas Picciani de Souza

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Matemática Aplicada, Instituto de Matemática da Universidade Federal do Rio de Janeiro(UFRJ), como parte dos requisitos necessários à obtenção do título de Mestre em Matemática Aplicada.

Orientador: Prof. Fábio Antonio Tavares Ramos.

Rio de Janeiro
Agosto de 2017

Universidade Federal do Rio de Janeiro

Processos Gaussianos para Estimação de Dados Ausentes de Tráfego

Douglas Picciani de Souza

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Matemática Aplicada, Instituto de Matemática da Universidade Federal do Rio de Janeiro(UFRJ), como parte dos requisitos necessários à obtenção do título de Mestre em Matemática Aplicada.

Aprovada por:

_____

Presidente, Prof. Fábio Antonio Tavares Ramos

_____

Prof. Heudson Tosta Mirandola

_____

Prof. Glaydston Mattos Ribeiro

_____

Prof. Rodrigo dos Santos Targino

_____

Prof. Paulo Sergio Ramirez Diniz

Rio de Janeiro
Agosto de 2017

Picciani de Souza, Douglas
V487c Processos Gaussianos para Estimação de Dados Ausentes de Tráfego / Douglas Picciani de Souza. –
Rio de Janeiro: UFRJ/ IM, 2017.
77 f. 29,7 cm.
Orientador: Fábio Antonio Tavares Ramos.
Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, Instituto de Matemática, Programa de Pós-Graduação em Matemática, 2017.

Agradecimentos

Para começar a escrever esta página, preciso daquele suco de uva aqui do lado.
(...)
Agora que estou aqui com o meu suco, posso começar. Eu gostaria de agradecer primeiramente a duas pessoas: Fábio Ramos e Cláudio Verdun. Estas duas pessoas não sabem o que fizeram com a minha vida nos últimos um ano e meio, mas graças a elas, ao olhar para trás, para o passado, sinto que amadureci bastante minhas ideias para daqui em diante. Em seguida agradeço a todos aqueles com quem compartilhei momentos no NIDF, são eles: Ivani Ivanova, mulher mais sagaz que eu conheço, desenha até pássaros bonitos, Pedro Schmidt, obrigado pelos outliers, Tiago Dominges, sua eloquência explicando qualquer coisa é uma inspiração, Jonathas Oliveira a.k.a. "Bangu", este que me prometeu um futebol e a quase 2 anos não cumpriu e Leonardo Assumpção, um cara que quando vejo chegando com o laptop nas mãos sinto-me intimidado com tanta programação.

Gostaria também de agradecer à todos do LESFER por tantos momentos ótimos juntos em lugares que eu nunca havia ido em toda minha vida: Romulo Orrico Filho, Saul Quadros, Heider Dantas, Eliézer Vieira da Silva, Gerusa Ravache, Cristiane Bernardo, Marcus Vinicius, Carlos Abramides, Leonardo Perim, André Nunes e a todos aqueles que me faltam em minha memória cansada agora.

Agradeço também ao DNIT em geral por me proporcionar crescer com um problema tão desafiador.

Agradeço também a Abilio Pereira de Lucena Filho, pelas ajudas no meio do caminho e pelas poucas conversas muito profundas que me fizeram levantar questões sobre a vida que me pego pensando de vez em quando andando por aí.

Agradeço à todos os envolvidos no meu crescimento como aluno e pessoa no Instituto de Matemática, da secretaria aos professores.

Ressalto aqui o agradecimento à todos aqueles que aceitaram compor a banca para a minha defesa de mestrado: Heudson Tosta Mirandola, Rodrigo dos Santos Targino, Glaydston Mattos Ribeiro, Paulo Sergio Ramirez Diniz. Muito obrigado por decidirem estar presentes neste momento de engrandecimento.

Aos meus amigos em Araruama, Cabo Frio, São Pedro... todos aqueles que se encontram em uma bola aberta centrada no Rio de Janeiro com um raio de 12 mil km, para também incluir o Paulo Ricardo Arantes Gilz em Toulouse, na França e o Guilherme Sales em Niterói.

Agradeço também aos meus pais, Antônio Arlindo Augusto de Souza e Glória Picciani de Souza, por terem me dado a base necessária para chegar até aqui como um ser humano. Agradeço Natasha Picciani de Souza e Michelle Picciani de Souza por terem um lugar especial no meu coração independente de qualquer coisa.

E por fim eu deixo um agradecimento caloroso a Paula Motta Pacheco, a mulher que amo e que me alimenta em troca da louça lavada.

Douglas Picciani de Souza
Agosto de 2017

"Gimli: – Oh, Come on. We can take them !
Aragorn: – It's a long way.
Gimli: – Toss me.
Aragorn: – What?
Gimli: – I cannot jump the distance, so you'll have to toss me!
(...)
Gimli: – Don't tell the Elf.
Aragorn: – Not a word."

Helms Deep, gate scene.
The Lord of the Rings:
The Two Towers.

Resumo

Processos Gaussianos para Estimação de Dados Ausentes de Tráfego

Douglas Picciani de Souza

Resumo da dissertação de Mestrado submetida ao Programa de Pós-graduação em Matemática Aplicada, Instituto de Matemática da Universidade Federal do Rio de Janeiro(UFRJ), como parte dos requisitos necessários à obtenção do título de Mestre em Matemática Aplicada.

Resumo: Em 2014 o DNIT, Departamento Nacional de Infraestrutura de Transportes, retomou o Plano Nacional de Contagem de Tráfego, PNCT, tendo instalado na malha rodoviária brasileira 320 postos de coleta de dados de tráfego com equipamentos de contagem de tráfego atuando de forma ininterrupta desde o início de suas atividades. Devido a motivos variados como mau funcionamento do sistema, clima, dificuldade de manutenção pela localização remota, problemas na linha de transmissão, etc., alguns dos registros efetuados acabam apresentando anomalias, como por exemplo, dados faltantes ou aberrantes. Tendo em vista a importância dos dados de contagem de tráfego para estudos de planejamento em geral, como análises econômicas e projetos rodoviários em escala nacional, o trabalho aqui presente possui a ambição de resolver este problema da área de Transporte Rodoviário utilizando técnicas de Aprendizado de Máquina. Muitos pesquisadores apresentam modelos variados para a estimação de dados faltantes em séries temporais, como por exemplo o modelo Auto-regressivo Integrado de Médias Móveis, ARIMA. Alguns estudos têm apresentado melhor performance com métodos baseados em Aprendizado de Máquina. Porém devido à falta de interpretabilidade, à complexidade destes modelos e custos computacionais elevados, estes deixam de ser empregados pelo usuário final. Considerando estes fatores, uma proposta para abordar este problema é a utilização de uma técnica conhecida como Processos Gaussianos, que apresenta um bom compromisso entre complexidade e interpretabilidade, no contexto do Aprendizado de Máquina. Portanto este trabalho possui o objetivo de fazer uma exposição desta técnica e em seguida desenvolver uma metodologia de imputação de séries temporais apoiando-se nela.

Palavras–chave: Processos gaussianos, aprendizado de máquina, séries temporais, transporte rodoviário, seleção de modelos, inferência bayesiana, regressão polinomial, sobreajuste.

Rio de Janeiro
Agosto de 2017

Abstract

Gaussian Processes for Estimation of Traffic Missing Data

Douglas Picciani de Souza

Abstract da dissertação de Mestrado submetida ao Programa de Pós-graduação em Matemática Aplicada, Instituto de Matemática da Universidade Federal do Rio de Janeiro (UFRJ), como parte dos requisitos necessários à obtenção do título de Mestre em Matemática Aplicada.

Abstract: In 2014, the DNIT, National Department of Transportation Infrastructure, restarted the Brazilian National Traffic Counting Plan, PNCT, having installed in the Brazilian highway network 320 traffic data collection stations with sensors operating continuously from the beginning of their activities. Due to several reasons such as system malfunction, weather, remote regions, communication line problems, etc., some of the data collected presents abnormal behavior, such as missing or aberrant data. Given the importance of traffic counting data for planning studies in general, such as economic analyzes and highway projects on a national scale, the work presented here has an ambition to solve this problem underlying the area of Transportation using techniques from Machine Learning. Many researchers have proposed different models for the estimation of missing data in time series, such as the Autoregressive Integrated Moving Average, ARIMA. Some studies have shown a better performance of methods based on Machine Learning. However due to the lack of interpretability, the complexity of their models and associated prohibitive costs, practitioners rarely employ these models as a solution. Overcoming some of these factors, a proposal to address this problem is the use of a technique known as Gaussian Processes, which presents a good compromise between complexity and interpretability, in the context of Machine Learning. This work aims the exposition of this technique and then the development of a methodology of time series imputation based on it.

Keywords: Gaussian process, machine learning, time series, transportation, model selection, bayesian inference, polynomial regression, overfitting.

Rio de Janeiro
Agosto de 2017

Conteúdo

Capítulo 1

Introdução

Em 2014 o *Departamento Nacional de Infraestrutura de Transportes*, DNIT, retomou o *Plano Nacional de Contagem de Tráfego*, PNCT, onde então foram instalados equipamentos permanentes de contagem, e classificação de diferentes categorias de veículos, em 320 postos de coleta distribuídos na malha rodoviária brasileira. A escolha dos locais para instalação desses equipamentos foi suportada através de critérios científicos bem estabelecidos realizados anteriormente pelo Instituto de Pesquisas Rodoviárias, IPR, em parceria com a Universidade Federal de Santa Cantarina, UFSC [32].

Desde então o PNCT realiza a coleta de dados diariamente de forma ininterrupta. Sendo inevitável o surgimento de problemas no sistema vigente, como, por exemplo, o mau funcionamento do equipamento de contagem ou até mesmo a perda de dados ao longo da linha de transmissão, uma quantidade de dados ausentes e anômalos acabou se mostrando significativa.

O interesse por parte do DNIT na coleta deste tipo específico de dado ocorre pela fundamental importância que ele possui para estudos de planejamento em geral, como análises econômicas e projetos rodoviários, que são essenciais para a sua tomada de decisão estratégica ao que concerne a infraestrutura rodoviária brasileira. Dentre as diversas atividades previstas nos projetos associados ao PNCT está o **tratamento estatístico** destes dados coletados pelos equipamentos permanentes de contagem.

Além do mais, a necessidade de uma base unificada de dados consolidados ocorre pelos seguintes motivos:

1. Permite o uso de métodos de inferência para dados completos como VMD[1], VMDA[2], médias, medianas, etc;

2. Permite o uso de informações disponíveis ao equipamento coletor de dados, ou seja, de sua origem, mas não disponíveis ao usuário final; e

3. Permite a resolução da presença de dados anômalos em seu cerne, evitando assim a sua propagação em estudos posteriores e futura confusão na análise de dados, e tendo também como consequência a redução de custos adiante.

Vale ressaltar que o PNCT segue o princípio da Integridade da Base de Dados condizente com as Diretrizes para Programas de Dados de Tráfego da American Association of State Highway and Transportation Officials, AASHTO [4]. Ou seja, a base de dados corrigida é disponibilizada para análises, porém a base de dados original é mantida intacta.

Portanto pode-se "grosseiramente" reduzir nosso problema ao de **séries temporais com dados ausentes ou anômalos**. A exigência por conseguinte é o desenvolvimento de uma metodologia capaz de estimar e reconstruir as anomalias provenientes de equipamentos permanentes de contagem do PNCT.

[1] VMD - Volume Médio Diário
[2] VMDA - Volume Médio Diário Anual

Tendo em vista esta exigência e a quantidade massiva de dados, torna-se clara a necessidade por uma metodologia que automatize a estimação e a reconstrução destes outliers. Para esta finalidade temos a presença forte da área do Aprendizado de Máquina.

Muitos trabalhos tem sido realizados para abordar o problema de dados de tráfego ausentes com técnicas de predição sofisticadas utilizando o Aprendizado de Máquina. Por exemplo, em [24], a estimação de contagem de tráfego foi investigada através de modelos de rede neural utilizando algoritmos genéticos, de regressão, de fatores, etc. Neste mesmo trabalho o autor também comenta que muitas destas técnicas de inteligência artificial são raramente utilizadas por usuários finais devido à sua complexidade, à falta de interpretabilidade e aos custos elevados associados. Em [25], um algoritmo fuzzy c-means combinado com um algoritmo genético foi proposto como um método de imputação assumindo uma hipótese de similaridade semanal.

Muitos pesquisadores propuseram métodos de séries temporais para estimar dados ausentes, como por exemplo o Modelo Auto-regressivo Integrado de Médias Móveis, ARIMA, ou ARIMA sazonal, [26, 27, 28, 29]. Estas técnicas geralmente propõem modelos de séries temporais, treinados por alguma base de dados histórica no intervalo de tempo corrente, para estimar outliers em intervalos subsequentes. Em [24], podemos observar entretanto que modelos ARIMA apresentam acurácias distintas para tipos diferentes de padrões de tráfego. Essencialmente, modelos ARIMA apresentam resultados melhores quando aplicados em séries temporais com padrões de tráfego estáveis, porque neste caso o modelo captura padrões de tráfego complexos sem o fenômeno de *overfitting*. Em [30], técnicas baseadas em métodos de Análise de Componentes Principais, PCA, são propostas para a imputação de volume de tráfego.

Nas últimas décadas foram criadas várias técnicas de Aprendizado Supervisionado de Máquina, uma subcategoria que consiste em aprender uma regra geral que mapeia entradas e saídas tendo como disponíveis alguns exemplos desta relação. Dentre elas, a regressão por *Processos Gaussianos* tem sido amplamente utilizada em problemas envolvendo séries temporais. Com esta técnica é possível capturar automaticamente correlações temporais de curto e médio prazo, além de ser capaz de lidar com a não-estacionariedade natural das séries de contagem de tráfego, incluindo possíveis sazonalidades diárias e semanais [16].

O Capítulo 2 é dedicado à exposição da regressão via Processos Gaussianos. Uma abordagem geralmente adotada para iniciar esta discussão utiliza uma noção abstrata demais envolvendo probabilidade no espaço de funções [5]. Neste trabalho, a abordagem considerada parte de problemas mais simples de regressão para então culminar nos Processos Gaussianos, tornando melhor a compreensão. Em seguida serão discutidos pontos mais gerais relacionados à seleção de modelos e à otimização dos hiperparâmetros.

Em seguida, no capítulo 3, é discutida a metodologia adotada para a imputação. O problema é detalhado de uma maneira mais profunda no qual torna-se mais clara a conexão com a metodologia proposta. Seguiremos com uma discussão em torno da metodologia envolvendo dois eixos principais: a seleção de modelos e a imputação dos outliers. Logo após os resultados para a seleção de modelos e estimação de outliers em séries temporais do PNCT são apresentados.

Finalmente concluímos brevemente o trabalho aqui realizado no capítulo 4. Algumas propostas futuras também são discutidas neste capítulo.

1.1 Por que prever é difícil ?

"Prediction is very difficult, especially about the future!"
Niels Bohr

Uma regressão consiste na adaptação de um determinado modelo $f(\mathbf{x})$, subjacente às hipóteses do modelador, aos dados $\{\mathbf{x}^{(i)}, y_i\}_{i=1}^n$ em questão, ou seja, consiste na *inferência* de uma função baseando-se nos dados existentes. No caso de séries temporais, a finalidade comum deste método é capturar alguma estrutura implícita nos dados para realizar uma estimação, podendo esta ser sobre o comportamento futuro dos dados ou mesmo sobre trechos passados com dados faltantes. Porém nem sempre o modelo escolhido é capaz de extrair certas estruturas. Portanto, para efetuarmos uma regressão é necessário termos definido as seguintes entidades:

I - **Base de dados**

II - **Modelo**

III - **Método de adaptação do modelo aos dados**

Começaremos analisando um caso simples de regressão polinomial para deixarmos claro alguns conceitos importantes.

Regressão polinomial

Para realizarmos uma regressão polinomial, seguimos adiante definindo as entidades necessárias como no início da seção:

I - **Base de dados** - Para este exemplo serão gerados sinteticamente 10 dados (x_i, y_i). Neste caso $\mathbf{x}$ será obtido através de 10 pontos equidistantes no intervalo $[0, 1]$. Cada y_i será gerado através da perturbação de uma função trigonométrica por um erro gaussiano proveniente de uma distribuição normal univariada de forma independente e identicamente distribuída. Assim:

$$\mathcal{D} = \{(x_i, y_i) \in \mathbb{R}^2 : i = 1, ..., n\}, \text{ onde } \begin{cases} y_i = sin(2\pi x_i) + \epsilon_i \\ \epsilon_i \sim \mathcal{N}(0, 0.3) \\ n = 10 \end{cases} \tag{1.1}$$

A Fig. 1.1a mostra em azul os pontos gerados e em verde a curva original. Esta será a base de dados de treino para o nosso modelo.

II - **Modelo** - O modelo considerado para esta regressão será polinomial:

$$f(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{1.2}$$

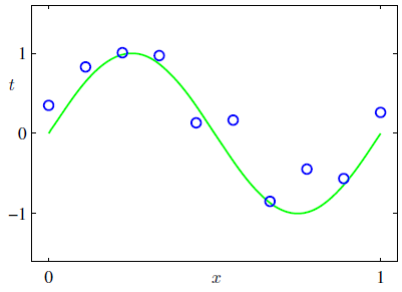
Neste caso M é o grau do polinômio e $\mathbf{w} = (w_0, ..., w_M) \in \mathbb{R}^{M+1}$ é o vetor de parâmetros. Observe que este modelo não é linear em relação a x, porém é linear em relação a $\mathbf{w}$ e por isto consideramos este como um modelo linear.

III - **Método de adaptação** - Cada vetor $\mathbf{w} \in \mathbb{R}^{M+1}$ resulta em uma função polinomial diferente. Uma forma de se obter valores para os parâmetros que adaptem o modelo à base de dados $\mathcal{D}$ é minimizando alguma função de erro que meça a diferença entre os valores y_i dos dados observados em $\mathcal{D}$ e os valores $f(x_i, \mathbf{w})$ estimados pelo modelo, como na Fig. 1.1b. Uma função de erro bastante utilizada neste caso é descrita abaixo:

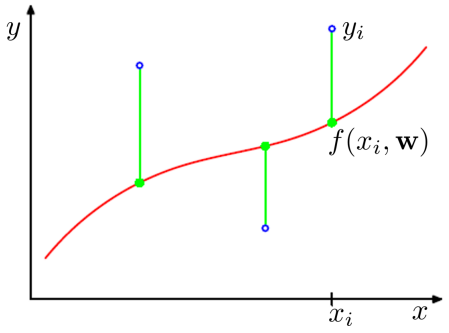
$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{n} \{f(x_i, \mathbf{w}) - y_i\}^2 \tag{1.3}$$

A linearidade do nosso modelo nos garante uma solução única e fechada $\mathbf{w}^*$ que minimiza nossa função de erro quadrática $E(\mathbf{w})$. Assim, o polinômio resultante desta regressão advém de $f(x, \mathbf{w}^*)$.

Perceba que em nenhum momento definimos o valor de M, o grau do polinômio. Tendo em vista que a quantidade de parâmetros depende de M, concluímos que a gama de polinômios explorada pelo modelo é definida por M. Dada esta hierarquia de dependências, denominaremos M como sendo um **hiperparâmetro** do nosso modelo. Na Fig. 1.2 temos quatro resultados desta regressão polinomial para os seguintes valores de M: 0, 1, 3 e 9. Para $M = 0$ ou $M = 1$ podemos verificar que o modelo não se torna



(a) Em azul se encontra a função seno perturbada por um erro gaussiano. Em verde a função seno original.



(b) A função de erro mede o quão distante o modelo estará dos dados através do quadrado das distâncias em verde.

Figura 1.1
Fonte: Bishop, 2006.

flexível [3] o bastante para capturar a estrutura oscilatória do seno. O resultado muda completamente para $M = 3$ sendo até agora o mais satisfatório. No caso em que $M = 9$, a função polinomial resultante se adapta exatamente aos pontos de treino, de fato para este caso $E(\mathbf{w}^*) = 0$, porém não parece capturar muito bem a estrutura subjacente da função seno em $\mathcal{D}$.

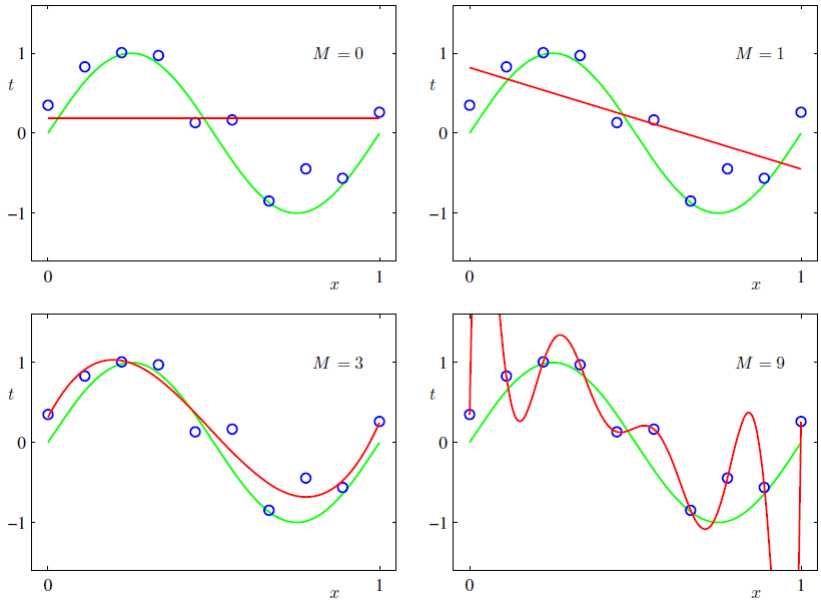


Figura 1.2: Alguns resultados de regressão polinomial. *Fonte: Bishop, 2006.*

Uma forma de avaliarmos o resultado de nosso modelo é verificando o cumprimento de sua finalidade: capturar a estrutura implícita em $\mathcal{D}$. Para isto iremos criar uma base de dados de validação $\mathcal{V}$ análoga à $\mathcal{D}$, porém com N = 100 dados novos no intervalo $[0, 1]$. Para que possamos realizar comparações do modelo utilizando bases de dados com tamanhos diferentes, utilizaremos o *erro quadrático médio* (root-

[3] A *flexibilidade*, ou complexidade, de um modelo pode ser encarada como a capacidade de se adaptar à comportamentos diversos com o fim de extrair alguma estrutura em particular presente na base de dados. Quanto mais flexível, maior a quantidade de comportamentos que poderão ser reproduzidos pelo modelo.

mean-square error) que é mais conveniente neste caso:

$$E_{RMS} = \sqrt{2E(\mathbf{w}^*)/n} \tag{1.4}$$

Na Fig. 1.3 podemos verificar o comportamento da regressão em função de M e das bases de dados de treino $\mathcal{D}$ e validação $\mathcal{V}$. Claramente percebemos que para $0 \leq M < 3$ o modelo erra bastante em todas as bases de dados, indicando aquilo que já havíamos constatado: para estes valores de M o modelo não é flexível o bastante para capturar a estrutura dos dados de treino. Para $3 \leq M < 9$ o modelo parece representar bem o comportamento esperado dos dados em $\mathcal{D}$. Agora para $M = 9$ ocorre um fenômeno interessante. O modelo se adapta tão bem aos dados de treino que ele passa *exatamente* por eles, $E_{RMS} = 0$, porém erra enormemente ao validarmos com $\mathcal{V}$. O comportamento oscilatório entre os pontos de D faz com que $f(x, \mathbf{w}^*)$ esteja muito aquém do comportamento suave da função senoidal. É interessante ver que mesmo incorporando todos os polinômios de grau menor naturalmente [4], ele erre bastante na validação. Este fenômeno é conhecido como "**over-fitting**" [2], ou seja ajustar o modelo além do necessário. Para $M = 9$ o modelo se torna tão flexível que passa a se ajustar ao erro aleatório gerado sinteticamente.

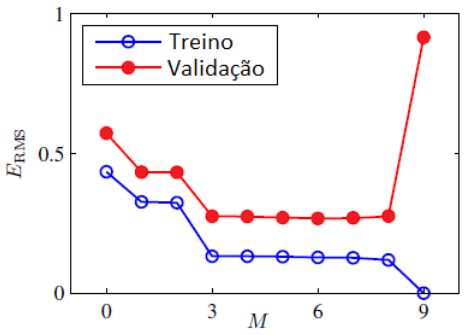


Figura 1.3: Um resultado de generalização. *Fonte: Bishop, 2006.*

Vamos observar o comportamento dos parâmetros em função de M na Fig. 1.4a. Note que ao tornar o modelo cada vez mais flexível, já que polinômios de grau menor se tornam casos particulares, o método de adaptação explora cada vez mais o espaço dos parâmetros. Para o modelo com $M = 9$ podemos observar uma *enorme variabilidade* na escolha dos parâmetros para criar o esforço necessário na adaptação exata dos dados em $\mathcal{D}$. Uma forma de conter esta variabilidade nos parâmetros, e por conseguinte o fenômeno de over-fitting, é adicionando um termo penalizador à função de erro:

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}\{f(x_i, \mathbf{w}) - y_i\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{1.5}$$

Esta técnica é conhecida como *regularização*. Neste caso $\|\mathbf{w}\|^2 = w_0^2 + w_1^2 + ... + w_M^2$ e λ é o hiperparâmetro que salienta a importância do termo regularizador frente à função de erro quadrática. A regularização impede que tenhamos parâmetros de magnitude muito elevada após o passo de adaptação do modelo, onde este é realizado fixando-se λ e calculando o minimizador $\mathbf{w}^*$ para $\tilde{E}$. Na Fig. 1.4b podemos ver o resultado para o modelo com $M = 9$. Quanto maior o valor de λ, mais restritiva se torna a variabilidade dos parâmetros. Na Fig. 1.5 mostramos a regressão resultante para dois valores de λ. No caso $\ln\lambda = -18$ percebe-se uma melhora significativa na captura do comportamento senoidal em $\mathcal{D}$ e para $\ln\lambda = 0$ ocorre um fenômeno contrário ao over-fitting, o "**under-fitting**", que fornece tanto uma adaptação ruim aos dados de treino $\mathcal{D}$ quanto uma predição ruim para os dados de validação $\mathcal{V}$. A escolha de λ pode ser feita realizando-se o passo de adaptação e a validação do modelo para vários valores de λ. Na Fig. 1.6 podemos observar que para um certo intervalo de valores de λ obtemos um bom compromisso entre os erros de treino e validação do modelo, mesmo utilizando o modelo com $M = 9$.

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

(a) Tabela dos parâmetros ótimos $\mathbf{w}^*$ para diferentes modelos polinomiais. Podemos verificar o aumento da variabilidade da magnitude dos parâmetros ao consideramos modelos polinomiais mais flexíveis, ou seja, de maior grau. *Fonte: Bishop, 2006.*

	$\ln\lambda=-\infty$	$\ln\lambda=-18$	$\ln\lambda=0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

(b) Tabela dos parâmetros ótimos $\mathbf{w}^*$ do modelo polinomial de grau $M=9$ para diferentes valores de λ. Observe a contensão na variabilidade da magnitude dos parâmetros ao aumentarmos λ. Note que no caso $\ln\lambda=-\infty$ não há regularização. *Fonte: Bishop, 2006.*

Figura 1.4

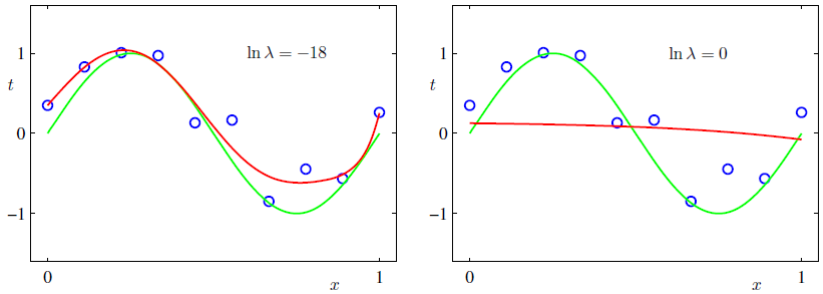


Figura 1.5: Regressão polinomial para diferentes valores de λ. *Fonte: Bishop, 2006.*

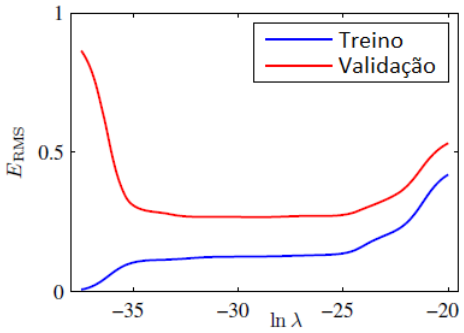


Figura 1.6: Gráfico de $\tilde{E}_{RMS}$ vs $\ln\lambda$ para o modelo polinomial com $M=9$. *Fonte: Bishop, 2006.*

Existem outras técnicas para controlar o fenômeno de over-fitting, como por exemplo o aumento da base de dados de treino como apresentado na Fig. 1.7, porém o foco desta seção é dar ênfase à relação entre a flexibilidade de um modelo e o fenômeno de over-fitting. Modelos mais flexíveis tendem a incluir os resultados de modelos menos flexíveis. Ao considerarmos modelos mais complexos, aumentando M, permitimos ao modelo abordar uma nova gama de curvas polinomiais para efetuar a regressão. Ao variarmos o hiperparâmetro λ, restringimos a exploração no espaço dos parâmetros numa tentativa de

[4] Um polinômio de grau $P < M$ pode ser obtido através do polinômio de grau M ao se fazer $w_{P+1} = ... = w_M = 0$.

controlar efeitos ruins provenientes da liberdade que o modelo possuía devido à sua flexibilidade. Porém novas questões começam a surgir: O quão flexível deve ser o modelo a ser utilizado para uma regressão dos dados em $\mathcal{D}$? Como achar parâmetros baseados em um senso de optimalidade que resultem em um equilíbrio entre over-fitting e under-fitting ? Neste momento torna-se claro que uma boa regressão provém deste equílibrio.

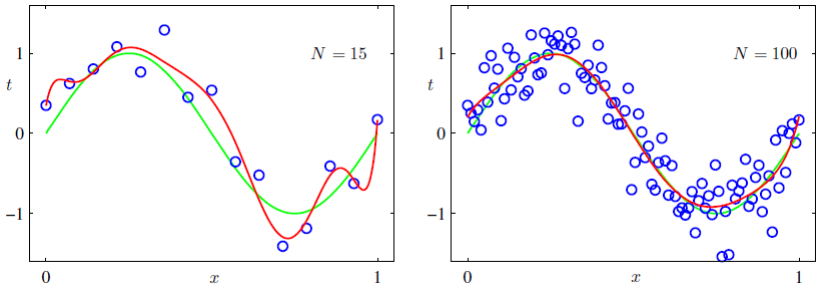


Figura 1.7: Efeito do aumento da base de dados de treino $\mathcal{D}$ sobre o fenômeno de over-fitting para o modelo com $M = 9$. *Fonte: Bishop, 2006.*

No capítulo seguinte veremos como uma regressão pode ser mais robusta utilizando princípios Bayesianos que incorporam naturalmente respostas para algumas das questões mais comuns, como estas a que chegamos anteriormente.

Capítulo 2

Regressão Bayesiana por Processos Gaussianos

Este capítulo é fortemente baseado no livro "Gaussian Process for Machine Learning"[1] cujos autores são Carl Edward Rasmussen e Christopher K. I. Williams.

A regressão bayesiana surge ao adotarmos as técnicas de inferência bayesiana para suprirmos o interesse em realizar inferência sobre a relação entre x_i e y_i em $\mathcal{D}$. No contexto Bayesiano, as três entidades necessárias para efetuarmos uma regressão, como visto na Seção 1.1, são complementadas da seguinte maneira:

I - **Base de dados**

II - **Modelo**:

i - **Modelo de erro**

ii - **Modelo a priori dos parâmetros**

III - **Método de adaptação do modelo aos dados**

A seguir iremos abordar um caso simples de regressão pelo método bayesiano.

2.1 Regressão linear Bayesiana

Para realizarmos uma regressão linear utilizando técnicas de inferência Bayesiana iremos aplicar as entidades estabelecidas no início deste capítulo.

I - **Base de dados** - Daqui em diante iremos supor uma base de dados de treino definida da seguinte maneira:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^{D+1} : i = 1, ..., n\} \tag{2.1}$$

Chamaremos de *matriz de design* X a matriz $D \times n$ que agrega todos os n vetores de entrada $\mathbf{x}_i$ em cada uma de suas colunas. Definiremos também o vetor de observações $\mathbf{y} = (y_1, ..., y_n)^\top$ de tal forma que possamos fazer o seguinte abuso de notação:

$$\mathcal{D} = (X, \mathbf{y}) \tag{2.2}$$

II.i - **Modelo de erro** - Considere o seguinte modelo de regressão linear simples com erro gaussiano:

$$f(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{w} \qquad y = f(\mathbf{x}) + \epsilon, \tag{2.3}$$

onde $\mathbf{x}$ é o vetor de entrada e $\mathbf{w}$ é um vetor coluna de pesos, neste caso parâmetros, do modelo linear considerado. Assumiremos ainda que o erro ϵ do nosso modelo é gerado de forma *independente e identicamente distribuída* e está associado a uma distribuição gaussiana com média nula e variância σ_n^2:

$$\epsilon \sim \mathcal{N}(0, \sigma_n^2) \tag{2.4}$$

Claramente temos o seguinte resultado:

$$y - f(\mathbf{x}) \sim \mathcal{N}(0, \sigma_n^2) \tag{2.5}$$

Podemos então diretamente estabelecer *a função de verossimilhança dos parâmetros*[1], também vista como uma densidade de probabilidade das observações $\mathbf{y}$ dado os parâmetros $\mathbf{w}$, $p(\mathbf{y}|X, \mathbf{w})$. Com a hipótese de independência do erro podemos então fatorar a função de verossimilhança em relação à $\mathcal{D}$:

$$\begin{aligned} p(\mathbf{y}|X, \mathbf{w}) &= \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_n} \exp(-\frac{(y_i - \mathbf{x}_i^\top \mathbf{w})^2}{2\sigma_n^2}) \\ &= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp(-\frac{1}{2\sigma_n^2}\|\mathbf{y} - X^\top \mathbf{w}\|^2) = \mathcal{N}(\mathbf{y}|X^\top \mathbf{w}, \sigma_n^2 \boldsymbol{I}) \end{aligned} \tag{2.6}$$

A função de verossimilhança possui uma importância significativa na inferência Bayesiana, pois é neste momento que introduzimos nossas hipóteses sobre $\mathbf{y}$ ao fixarmos algum vetor de parâmetros $\mathbf{w}$.

II.ii - **Modelo a priori dos parâmetros -** Agora temos que determinar a nossa distribuição a priori dos parâmetros $p(\mathbf{w})$, onde, de acordo com a inferência Bayesiana, será expressada nossas hipóteses sobre os parâmetros antes de obtermos as informações dos dados em $\mathcal{D}$. Escolheremos uma priori gaussiana de média nula com uma matriz de covariância Σ_p nos pesos:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \Sigma_p) \tag{2.7}$$

O papel da priori e suas propriedades serão discutidos mais adiante na Seção 2.2. Por enquanto considere esta priori assim especificada.

III - **Método de adaptação do modelo aos dados -** Um método bastante utilizado para a estimação dos parâmetros $\mathbf{w}$ com o fim de adaptar o modelo aos dados é o da *máxima verossimilhança*[2]. Ele se baseia em estimar $\mathbf{w}_{ML}$ de forma que maximize a função de verossimilhança dos parâmetros, dada pela Eq. (2.6), e ignore todo o conhecimento a priori acerca dos parâmetros. Isto corresponde a escolher o valor de $\mathbf{w}$ para o qual a probabilidade do conjunto de dados observados é maximizada. O mesmo pode ser realizado para a estimação do hiperparâmetro σ_n. Porém como veremos mais adiante na Seção 2.2, este método abre a possibilidade de over-fitting. A seguir descreveremos uma abordagem mais robusta para a adaptação do modelo aos dados, usando o paradigma Bayesiano.

A distribuição a posteriori dos parâmetros

A inferência Bayesiana neste modelo linear é baseada na distribuição a posteriori dos pesos que é obtida através do teorema de Bayes:

$$\text{posteriori} = \frac{\text{verossimilhança} \times \text{priori}}{\text{verossimilhança marginal}} \qquad p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|X)} \tag{2.8}$$

onde a constante de normalização, também conhecida como *verossimilhança marginal* ou *evidência*, é independente dos pesos:

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})d\mathbf{w} \tag{2.9}$$

[1] A função de verossimilhança dos parâmetros $\mathcal{L}(\mathbf{w}) = p(\mathbf{y}|X, \mathbf{w})$ não é uma densidade de probabilidade em relação à $\mathbf{w}$.
[2] *maximum likelihood.*

A posteriori na Eq. (2.8) combina a função de verossimilhança e a priori, e captura tudo o que sabemos sobre os parâmetros. Escrevendo somente os termos da função de verossimilhança e da priori, e "completando o quadrado", obtemos o seguinte resultado:

$$\begin{aligned} p(\mathbf{w}|X,\mathbf{y}) &\propto \exp(-\frac{1}{2\sigma_n^2}(\mathbf{y}-X^\top\mathbf{w})^{\mathrm{T}}(\mathbf{y}-X^\top\mathbf{w}))\exp(-\frac{1}{2}\mathbf{w}^\top\Sigma_p^{-1}\mathbf{w}) \\ &\propto \exp(-\frac{1}{2}(\mathbf{w}-\bar{\mathbf{w}})^\top(\frac{1}{\sigma_n^2}XX^\top+\Sigma_p^{-1})(\mathbf{w}-\bar{\mathbf{w}})) \end{aligned} \tag{2.10}$$

onde $\bar{\mathbf{w}} = \sigma_n^{-2}(\sigma_n^{-2}XX^\top + \Sigma_p^{-1})^{-1}X\mathbf{y}$. Torna-se claro que o formato da distribuição a posteriori é de uma Gaussiana com média $\bar{\mathbf{w}}$ e matriz de covariância A^{-1}:

$$p(\mathbf{w}|X,\mathbf{y}) \sim \mathcal{N}(\mathbf{w}|\bar{\mathbf{w}} = \frac{1}{\sigma_n^2}A^{-1}X\mathbf{y}, A^{-1}) \tag{2.11}$$

onde $A = \sigma_n^{-2}XX^\top + \Sigma_p^{-1}$.

Outro método bastante utilizado para estimar o valor mais provável de $\mathbf{w}$ é baseado na maximização da distribuição a posteriori dos parâmetros, ou MAP. Entretanto, podemos ainda assim abordar o problema de uma forma mais Bayesiana levando em conta as incertezas de todos os possíveis valores para os parâmetros, ao invés de nos apoiarmos em uma estimativa pontual de $\mathbf{w}$.

A predição Bayesiana

Para efetuarmos predições para a observação y_* em algum $\mathbf{x}_*$ neste contexto Bayesiano, realiza-se uma média sobre todos os possíveis valores para os parâmetros sendo ponderada pela sua distribuição a posteriori. Neste caso o resultado final contém as incertezas de todos os possíveis parâmetros do modelo. Isto contrasta enormemente com as abordagens não-Bayesianas, como exemplo temos a Seção anterior, onde um único ponto no espaço dos parâmetros é escolhido respeitando algum tipo de critério estabelecido. Portanto a distribuição preditiva para $f_* \triangleq f(\mathbf{x}_*)$ em $\mathbf{x}_*$ é calculada realizando-se uma média entre todos os resultados de todos os possíveis modelos lineares em relação à posteriori gaussiana dos parâmetros:

$$\begin{aligned} p(f_*|\mathbf{x}_*,X,\mathbf{y}) &= \int p(f_*|\mathbf{x}_*,\mathbf{w})p(\mathbf{w}|X,\mathbf{y})d\mathbf{w} \\ &= \mathcal{N}(\mathbf{x}_*^{\mathrm{T}}\bar{\mathbf{w}}, \mathbf{x}_*^{\mathrm{T}}A^{-1}\mathbf{x}_*) \\ &= \mathcal{N}(\frac{1}{\sigma_n^2}\mathbf{x}_*^{\mathrm{T}}A^{-1}X\mathbf{y}, \mathbf{x}_*^{\mathrm{T}}A^{-1}\mathbf{x}_*) \end{aligned} \tag{2.12}$$

A distribuição preditiva é novamente uma Gaussiana com média dada pela multiplicação entre a média da posteriori dos parâmetros e o novo ponto de teste $\mathbf{x}_*$. A variância preditiva é uma forma quadrática de $\mathbf{x}_*$ com a matriz de covariância da posteriori dos parâmetros. Observe que a incerteza preditiva cresce com a magnitude de $\mathbf{x}_*$, como é de se esperar de um modelo linear.

A Fig. 2.1 é um exemplo bastante rico da construção de uma posteriori dos parâmetros para uma regressão linear bayesiana simples envolvendo a adaptação de retas. A seguir descreveremos o procedimento realizado para a geração deste exemplo. Primeiramente iremos definir a geração sintética de dados, o modelo de erro e a priori:

- **Base de dados sintética** - Os dados sintéticos $(x_n, \tilde{t}_n) \in \mathbb{R}^2$ foram gerados de forma a serem i.i.d.:

$$\tilde{t}_n = f(x_n,\mathbf{a}) + \epsilon, \text{ onde } x_n \sim \mathrm{U}(-1,1) \text{ e } \epsilon \sim \mathcal{N}(0,\sigma^2) \tag{2.13}$$

com $\sigma = 0.2$. A função latente é dada da seguinte forma:

$$f(x,\mathbf{a}) = a_0 + a_1 x, \text{ onde } (a_0,a_1) = (-0.3, 0.5) \tag{2.14}$$

- **Modelo de erro** - O modelo de erro assume que trabalharemos unicamente com retas. Consideraremos que o desvio padrão $\sigma = 0.2$ do erro dos dados sintéticos é conhecido. Assim:

$$t = f(x, \mathbf{w}) + \epsilon, \text{ onde } \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{2.15}$$

  onde $\mathbf{w} = (w_0, w_1)$ e $f(x_n, \mathbf{w}) = w_0 + w_1 x$.

- **Priori dos parâmetros** - Assumiremos a seguinte priori para os parâmetros $\mathbf{w}$:

$$p(\mathbf{w}|\lambda) = \mathcal{N}(\mathbf{0}|\lambda^{-1}\mathbf{I}) \tag{2.16}$$

  onde $\lambda = 2$.

Na Fig. 2.1 podemos ver o resultado do aprendizado Bayesiano para este problema de maneira que a quantidade de dados utilizados para treino é aumentada gradualmente para que possamos observar a natureza do método Bayesiano. A posteriori gerada para uma certa quantidade de dados de treino servirá como a priori para a adição de novos dados. A 1ª linha desta figura corresponde ao momento antecedente à observação dos dados de treino mostrando o gráfico da distribuição a priori em função dos parâmetros $\mathbf{w}$ e seis amostras da função $y(x, \mathbf{w})$ no qual os valores de $\mathbf{w}$ são aleatoriamente gerados por essa distribuição. Na 2ª linha temos o momento após a observação de um único dado $(x, \tilde{t})$ representado por um círculo em azul na 3ª coluna. Na 1ª coluna temos o gráfico da função de verossimilhança $p(t|x, \mathbf{w})$ para este dado em função de $\mathbf{w}$. Note que a função de verossimilhança apresenta uma restrição no espaço dos parâmetros que indica que as retas tem que passar perto do ponto de treino, onde a noção de proximidade é dada através do desvio padrão do erro, σ. Para fins de comparação, os verdadeiros parâmetros utilizados $(a_0, a_1) = (-0.3, 0.5)$ usados para a geração dos dados sintéticos são mostrados no espaço dos parâmetros por uma cruz branca. Quando multiplicamos esta função de verossimilhança pela priori da 1ª linha e normalizamos, obtemos a distribuição a posteriori na 2ª linha da coluna central. Amostras da função $y(x, \mathbf{w})$ são obtidas através da geração aleatória de $\mathbf{w}$ pela distribuição a posteriori e apresentadas na coluna da direita. A 3ª linha desta figura mostra o efeito da observação do segundo ponto de treino mostrado na coluna da direita novamente por um círculo azul. A função de verossimilhança correspondente unicamente ao segundo ponto de treino é mostrada na 1ª coluna. Ao multiplicarmos esta função de verossimilhança pela distribuição a posteriori encontrada na 2ª linha, nós obtemos a nova distribuição a posteriori na 3ª linha da coluna do meio. Note que esta é exatamente a distribuição a posteriori que obteríamos se combinássemos a priori original com a função de verossimilhança para os dois pontos de treino utilizados até agora. Como uma reta é bem definida por dois pontos no espaço, os dois pontos de treino utilizados são informação suficiente para obtermos uma distribuição a priori relativamente mais compacta. Na 3ª coluna podemos ver que as retas provenientes de amostras desta posteriori passam bem perto dos dois pontos de treino. A 4ª linha mostra o efeito obtido ao observarmos 20 pontos de treino. A coluna da esquerda mostra o gráfico da função de verossimilhança unicamente para o vigésimo ponto de treino e a coluna central mostra o resultado da distribuição a posteriori bem mais compacta do que na 3ª linha devido ao aprendizado de 20 pontos de treino. Se utilizássemos infinitos pontos de treino, veríamos uma função delta de dirac centrada exatamente na posição dos verdadeiros parâmetros indicada pela cruz branca.

Outro exemplo é dado pela Fig. 2.2 que também nos mostra uma regressão linear onde por fim temos o resultado da distribuição preditiva para novos valores de $x_* \in \mathbb{R}$.

2.1.1 Projeções das entradas no espaço de características

Modelos puramente lineares não possuem a flexibilidade necessária para abordar o problema da regressão de dados que possuam estruturas inerentes mais complexas. Neste caso, uma ideia muito simples para sobrepor esta limitação é projetar os dados de entrada em algum espaço de dimensão mais alta utilizando uma base de funções e então trabalhar com o modelo linear neste espaço ao invés de diretamente nos dados de entrada. Como exemplo temos a Seção 1.1, onde cada entrada escalar x foi projetada no espaço de potências de x, $\boldsymbol{\phi}(x) = (1, x, x^2, x^3, ...)^\top$, para que pudesse ser feita uma regressão polinomial. Desde

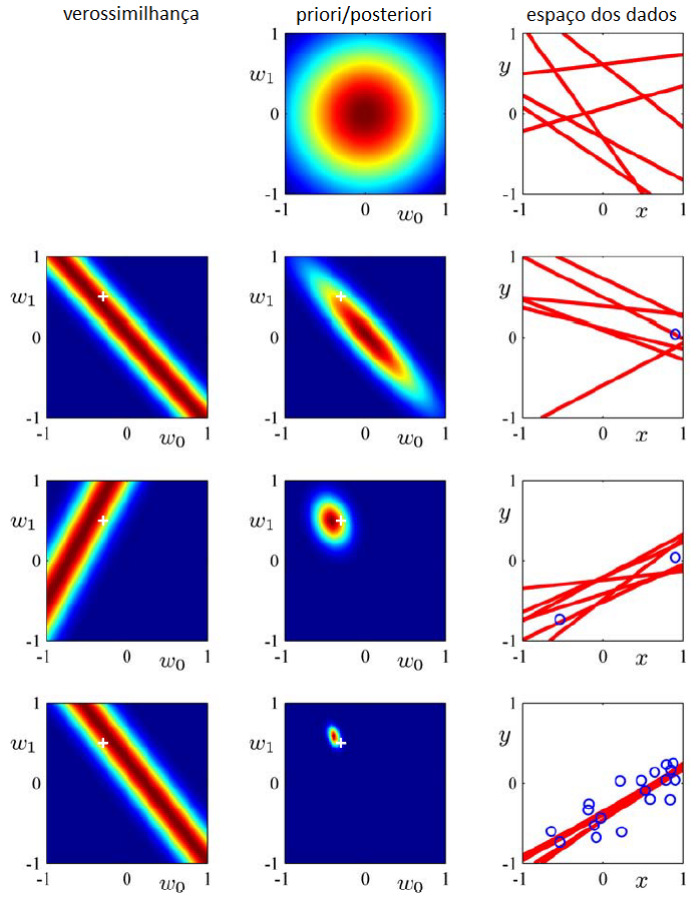


Figura 2.1: Ilustração do aprendizado Bayesiano para um simples modelo linear $y(x, \mathbf{w}) = w_0 + w_1 x$ onde são abordadas as ferramentas para a construção da distribuição a posteriori progressivamente com o aumento da quantidade de dados de treino. Uma discussão mais profunda se encontra no texto. *Fonte: Bishop, 2006.*

que nossa base de funções seja independente dos parâmetros w, o modelo resultante ainda será linear em relação aos parâmetros e portanto analiticamente tratável. Nesta Seção assumiremos que a base de funções é dada.

O Modelo através da projeção dos dados de entrada

Nossa base de funções será dada através da função $\boldsymbol{\phi}(\mathbf{x})$ que mapeia um vetor D-dimensional em um espaço de características N-dimensional. Além disso considere a matriz $\boldsymbol{\Phi}(X)$ onde suas colunas são os vetores $\boldsymbol{\phi}(\mathbf{x})$ para todo $\mathbf{x} \in \mathcal{D}$. O modelo então segue da seguinte forma:

$$f(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w}, \tag{2.17}$$

onde o vetor de parâmetros agora possui tamanho N.

A nova distribuição de probabilidade preditiva

A análise deste modelo é realizada de forma análoga ao do modelo linear da Seção anterior, de maneira que a distribuição preditiva, assim como os outros resultados, se mantém, exceto que toda ocorrência da

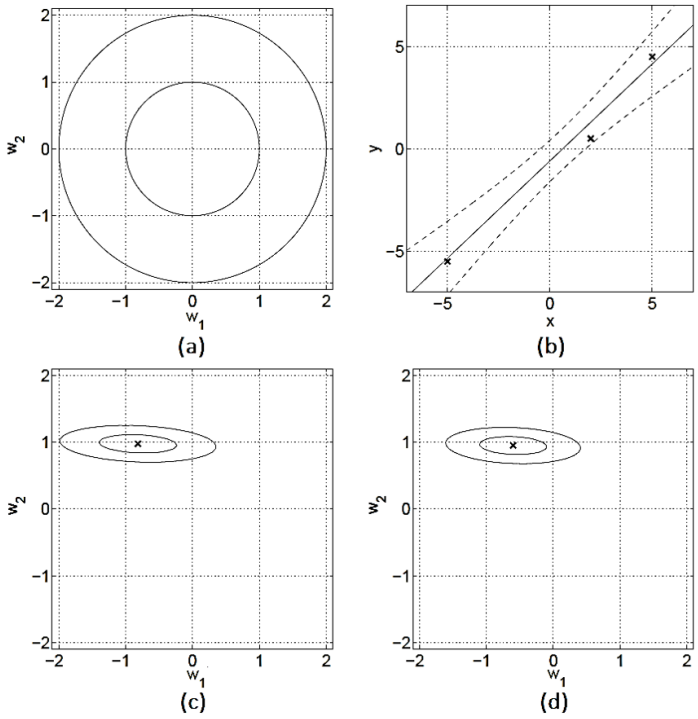


Figura 2.2: Exemplo de modelo linear Bayesiano $f(x) = w_1 + w_2 x$. A Fig.(a) mostra a distribuição a priori $p(\mathbf{w}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. A Fig.(b) mostra três pontos de treino marcados com cruzes. A Fig.(c) mostra os contornos da função de verossimilhança $p(\mathbf{y}|X, \mathbf{w})$ com desvio padrão $\sigma_n = 1$. Note que w_2 é muito mais *bem determinado* do que w_1. A Fig.(d) mostra a distribuição a posteriori dos parâmetros, $p(\mathbf{w}|X, \mathbf{y})$. Comparando os máximos da verossimilhança e da posteriori, podemos perceber um leve deslocamento do gráfico na direção do eixo $\mathbf{w}_1$. Todos os contornos equiprováveis são dados por 1 e 2 desvios padrões. Sobreposto aos dados de treino na Fig.(b) se encontram a média ± dois desvios padrões da distribuição preditiva. *Fonte: Rasmussen, 2006.*

matriz de design X e do vetor de entrada $\mathbf{x}$ será substituída, respectivamente, por $\mathbf{\Phi}(X)$ e $\boldsymbol{\phi}(\mathbf{x})$:

$$f_*|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}(\frac{1}{\sigma_n^2}\boldsymbol{\phi}(\mathbf{x}_*)^\top A^{-1}\mathbf{\Phi}\mathbf{y},\ \ \boldsymbol{\phi}(\mathbf{x}_*)^\top A^{-1}\boldsymbol{\phi}(\mathbf{x}_*)) \tag{2.18}$$

onde $\mathbf{\Phi} = \mathbf{\Phi}(X)$ e $A = \sigma_n^{-2}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}} + \Sigma_p^{-1}$.

Uma conexão com a regressão polinomial

Se considerarmos uma base de funções polinomiais $\boldsymbol{\phi}(x) = (1, x, ..., x^M)^\top$, retornamos ao problema da Seção 1.1. Ao analisarmos a regressão polinomial do ponto de vista Bayesiano com os métodos de adaptação do modelo descritos na Seção 2.1, podemos verificar conexões bastante interessantes com os resultados da Seção 1.1.

Na Seção anterior três métodos foram propostos para a adaptação de modelos: máxima verossimilhança, máximo a posteriori e a técnica de predição Bayesiana. A seguir veremos os resultados para cada um destes métodos:

- Máxima Verossimilhança

  Este método apoia-se na estimação de $\mathbf{w}_{ML}$ através da maximização da função de verossimilhança

que neste caso assume a seguinte forma:

$$p(\mathbf{y}|X,\mathbf{w}) = \prod_{i=1}^{n} \mathcal{N}(y_i|\boldsymbol{\phi}(\mathbf{x}_i)^\top \mathbf{w}, \sigma_n^2 I) \tag{2.19}$$

Por conveniência o processo de maximização se dará através do logaritmo da função de verossimilhança. Como a função logarítmica é monótona crescente, a posição em que o máximo ocorre não se altera. Assim, a função a ser maximizada possui a seguinte forma:

$$\log p(\mathbf{y}|X,\mathbf{w}) = -\frac{1}{2\sigma_n^2}\sum_{i=1}^{n}(y_i - \boldsymbol{\phi}(x)^\top \mathbf{w})^2 - n\,\log\sigma_n - \frac{n}{2}\log 2\pi \tag{2.20}$$

Note que os dois últimos termos não dependem de $\mathbf{w}$, assim podemos omiti-los desta otimização. Observe também que reescalonar o logaritmo da verossimilhança por um coeficiente constante positivo não altera a localização do máximo $\mathbf{w}_{ML}$. Portanto, podemos fixar o valor do desvio padrão do erro em $\sigma_n = 1$. Finalmente ao invés de maximizar o logaritmo da verossimilhança, podemos equivalentemente minimizar o logaritmo negativo da verossimilhança após terem sido feitas estas modificações:

$$\mathbf{w}_{ML} = \min_{\mathbf{w}} \{-\log \tilde{p}(\mathbf{y}|X,\mathbf{w})\} = \min_{\mathbf{w}} \left\{ \frac{1}{2}\sum_{i=1}^{n}(y_i - \boldsymbol{\phi}(x)^\top \mathbf{w})^2 \right\} \tag{2.21}$$

Portanto temos claramente que a maximização da função de verossimilhança em relação aos parâmetros $\mathbf{w}$ é equivalente a minimizar a *função de erro soma de quadrados*, definida na Eq. (1.3). Assim, podemos concluir que a função de erro soma de quadrados é uma consequência da maximização da função de verossimilhança dos parâmetros sob a hipótese de erro Gaussiano.

Uma vez que tenhamos obtido $\mathbf{w}_{ML}$, podemos realizar o mesmo procedimento para calcularmos o hiperparâmetro de precisão σ_{ML} sob a condição de que $\mathbf{w} = \mathbf{w}_{ML}$. Assim, maximizando a Eq. (2.20), obtemos:

$$\sigma_{ML}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \boldsymbol{\phi}(x)^\top \mathbf{w}_{ML}\right)^2 \tag{2.22}$$

- Máximo a Posteriori ou MAP

  Ao invés de maximizarmos a função de verossimilhança dos parâmetros, podemos utilizar a abordagem Bayesiana introduzindo algum conhecimento sobre os parâmetros a priori e maximizar sua distribuição a posteriori. De acordo com a Eq. (2.7) e definindo $\sum_p = \alpha^{-1}\boldsymbol{I}$, temos o seguinte resultado para a priori:

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\boldsymbol{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left(-\frac{\alpha}{2}\mathbf{w}^\top \mathbf{w}\right) \tag{2.23}$$

onde α é a precisão da distribuição a priori e $M+1$ é a quantidade de parâmetros em $\mathbf{w}$ para um polinômio de grau M. Do teorema de Bayes, segue que :

$$p(\mathbf{w}|\mathbf{y},X) \sim p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w}) \tag{2.24}$$

Assim, calculando o logaritmo da priori dos parâmetros e combinando com a Eq. (2.20), temos que:

$$\log\left[p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})\right] = -\frac{1}{2\sigma_n^2}\sum_{i=1}^{n}(y_i - \boldsymbol{\phi}(x)^\top \mathbf{w})^2 - n\,\log\sigma_n - \frac{n}{2}\log 2\pi + \frac{M+1}{2}\log\left(\frac{\alpha}{2\pi}\right) - \frac{\alpha}{2}\mathbf{w}^\top \mathbf{w} \tag{2.25}$$

Como vimos anteriormente, o reescalonamento por um coeficiente constante positivo e termos que não possuem a variável a ser maximizada não alteram a posição do máximo procurado, que neste

caso irá se encontrar em $\mathbf{w}_{MAP}$. Portanto, baseando-se na expressão negativa da Eq. (2.25), o problema de maximização a posteriori é equivalente ao seguinte problema de minimização:

$$\mathbf{w}_{MAP} = \min_{\mathbf{w}} \left\{ \frac{1}{2\sigma_n^2} \sum_{i=1}^{n} (y_i - \boldsymbol{\phi}(x)^\top \mathbf{w})^2 + \frac{\alpha}{2} \mathbf{w}^\top \mathbf{w} \right\} \tag{2.26}$$

Portanto podemos observar que maximizar a distribuição a posteriori é equivalente a minimizar a *função de erro soma de quadrados regularizada*, como na Eq. (1.5), com o parâmetro de regularização $\lambda = \sigma_n^2 \alpha$.

- Predição Bayesiana

  Vimos até agora que os dois métodos anteriores baseiam-se na estimação pontual dos parâmetros $\mathbf{w}$. A predição Bayesiana como vista na Seção 2.1 realiza uma média sobre todas as soluções possíveis para a função latente sendo ponderada pela posteriori dos parâmetros. O resultado do cálculo da distribuição preditiva é dada pela Eq. (2.18) e será reescrito aqui em termos de sua média e sua variância resultantes:

$$\begin{aligned} m(\mathbf{x}_*) &= \frac{1}{\sigma_n^2} \boldsymbol{\phi}(\mathbf{x}_*)^\top A^{-1} \boldsymbol{\Phi} \mathbf{y} \\ s^2(\mathbf{x}_*) &= \boldsymbol{\phi}(\mathbf{x}_*)^\top A^{-1} \boldsymbol{\phi}(\mathbf{x}_*) \end{aligned} \tag{2.27}$$

  Claramente podemos observar aqui que a predição Bayesiana além de ser um método mais robusto do ponto de vista de que se leva em conta todas as incertezas sobre todos os possíveis parâmetros, ela também gera *naturalmente* os intervalos de confiança para sua predição. A Fig. 2.3 apresenta a solução para a regressão polinomial Bayesiana tendo neste caso assumido a priori dos parâmetros como na Eq. (2.23). Os valores assumidos para os hiperparâmetros são $\alpha = 5 \times 10^{-3}$ e $\sigma_n = 0.3$. O curva em vermelho representa a média $m(\mathbf{x}_*)$ e o sombreado em vermelho representa o intervalo de confiança correspondendo a ± 1 desvio padrão $s(\mathbf{x}_*)$ em torno da média.

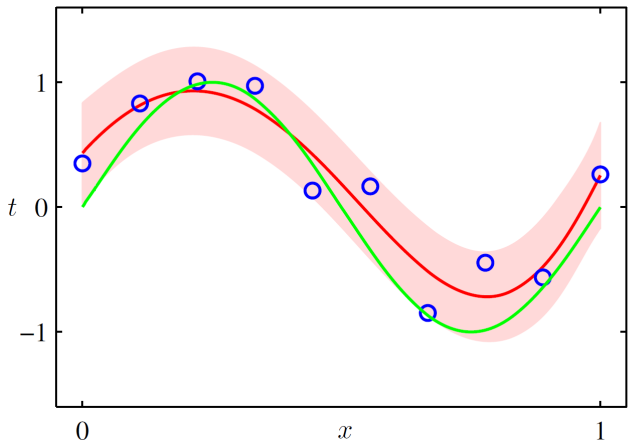


Figura 2.3: Resultado da distribuição preditiva para a regressão polinomial Bayesiana utilizando uma base de funções polinomiais de grau $M = 9$, com hiperparâmetros $\alpha = 5 \times 10^{-3}$ e $\sigma_n = 0.3$, no qual a curva em vermelho representa a média e a região sombreada vermelha corresponde a ± 1 desvio padrão em torno da média. *Bishop, 2006.*

A distribuição preditiva reescrita

Para que possamos realizar predições com a equação (2.18), é necessário inverter a matriz A de tamanho $N \times N$ e isto pode vir a ser um grande problema caso a dimensão N da base de funções $\phi(\mathbf{x})$ seja grande

demais. Entretanto, podemos reescrever esta equação da seguinte maneira:

$$f_*|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}(\boldsymbol{\phi}_*^\top \Sigma_p \boldsymbol{\Phi}(K + \sigma_n^2 I)^{-1}\mathbf{y}, \quad \boldsymbol{\phi}_*^\top \Sigma_p \boldsymbol{\phi}_* - \boldsymbol{\phi}_*^\top \Sigma_p \boldsymbol{\Phi}(K + \sigma_n^2 I)^{-1}\boldsymbol{\Phi}^\top \Sigma_p \boldsymbol{\phi}_*) \tag{2.28}$$

onde utilizamos $\boldsymbol{\phi}(\mathbf{x}_*) = \boldsymbol{\phi}_*$ e definimos $K \triangleq \boldsymbol{\Phi}^{\mathbf{T}} \Sigma_p \boldsymbol{\Phi}$. Para a reescrita da média, note que com as definições de A e K temos a seguinte identidade:

$$\sigma_n^{-2}\boldsymbol{\Phi}(K + \sigma_n^2 I) = \sigma_n^{-2}\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top \Sigma_p \boldsymbol{\Phi} + \sigma_n^2 I) = A\Sigma_p \boldsymbol{\Phi} \tag{2.29}$$

Multiplicando à esquerda por A^{-1} e à direita por $(K + \sigma_n^2)^{-1}$ resulta na seguinte equivalência:

$$\sigma_n^{-2} A^{-1}\boldsymbol{\Phi} = \Sigma_p \boldsymbol{\Phi}(K + \sigma_n^2)^{-1} \tag{2.30}$$

mostrando assim que a média nas equações 2.18 e 2.28 são equivalentes. Agora para reescrevermos a variância iremos utilizar o seguinte lema [10, pag. 18]:

Lema 2.1. *(Woodbury, Sherman & Morrison) O lema da inversão da matriz:*

$$(Z + UWV^\top)^{-1} = Z^{-1} - Z^{-1}U(W^{-1} + V^\top Z^{-1}U)^{-1}V^\top Z^{-1} \tag{2.31}$$

asumindo que todas as inversas relevantes existem.

Então fazendo $Z^{-1} = \Sigma_p$, $W^{-1} = \sigma_n^2 I$ e $V = U = \boldsymbol{\Phi}$ chegamos à equivalência das variâncias nas equações (2.18) e (2.28):

$$(\Sigma_p^{-1} + \sigma_n^{-2}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}})^{-1} = \Sigma_p - \Sigma_p \boldsymbol{\Phi}(\sigma_n^2 I + \boldsymbol{\Phi}^{\mathrm{T}}\Sigma_p \boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\Sigma_p \tag{2.32}$$

Perceba que na Eq. (2.28) é necessário inverter uma matriz $n \times n$, o que se torna bem mais conveniente quando o número de observações é menor que a dimensão do espaço de características, i.e., $n < N$.

Representação utilizando Kernel (Kernel Trick)

Verificando esta mesma equação, percebe-se que o espaço de características sempre aparece nos seguintes formatos:

$$\boldsymbol{\Phi}^{\mathrm{T}}\Sigma_p \boldsymbol{\Phi}, \quad \boldsymbol{\phi}_*^{\mathrm{T}}\Sigma_p \boldsymbol{\Phi} \text{ ou } \boldsymbol{\phi}_*^{\mathrm{T}}\Sigma_p \boldsymbol{\phi}_* \tag{2.33}$$

Como as entradas destas matrizes assumem a seguinte forma, $\boldsymbol{\phi}(\mathbf{x})^\top \Sigma_p \boldsymbol{\phi}(\mathbf{x}')$, onde $\mathbf{x}$ e $\mathbf{x}'$ são pontos de treino ou teste, podemos definir uma função $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \Sigma_p \boldsymbol{\phi}(\mathbf{x}')$ que iremos chamar de *função de covariância* ou *kernel*, por razões que ficarão mais claras na próxima Seção.

Note que o termo $\boldsymbol{\phi}(\mathbf{x})^\top \Sigma_p \boldsymbol{\phi}(\mathbf{x}')$ é um produto interno com relação à Σ_p. Sendo Σ_p uma matriz positiva definida, podemos definir rigorosamente $\Sigma_p^{1/2}$, onde $(\Sigma_p^{1/2})^2 = \Sigma_p$, através por exemplo da decomposição SVD. Caso $\Sigma_p = UDU^\top$, então $\Sigma_p^{1/2} = UD^{1/2}U^\top$. Isto torna possível definir $\boldsymbol{\psi}(\mathbf{x}) = \Sigma_p^{1/2}\boldsymbol{\phi}(\mathbf{x})$ e consequentemente chegamos à seguinte representação: $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\psi}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{x}')$

Por conta disto, geralmente o foco na construção de um algoritmo capaz de realizar predições deixa de ser na computação dos vetores de características para vir a ser na computação do kernel em si [1]. Neste momento torna-se claro que a computação do kernel ditará a estrutura da função a ser adaptada aos pontos de treino. Isto será melhor estabelecido mais adiante.

Na próxima Seção veremos como é possível chegar aos mesmos resultados realizando-se uma inferência diretamente no espaço de funções.

2.2 Regressão Bayesiana via Processos Gaussianos

Como vimos anteriormente modelos paramétricos geralmente assumem uma estrutura $f(\mathbf{x}, \mathbf{w})$ a priori governada puramente por uma quantidade finita de parâmetros $\mathbf{w}$. A incerteza sobre o modelo f é então expressa por uma distribuição de probabilidade a posteriori em $\mathbf{w}$. Temos aqui duas hipóteses feitas,

uma sobre a estrutura de f e uma sobre a incerteza nos parâmetros $\mathbf{w}$. Entretanto a estrutura inerente aos dados em $\mathcal{D}$ não é conhecida, portanto fixar uma estrutura a priori desta maneira parece ser muito restritivo, pois a precisão com a qual f pode ser determinada é limitada ao melhor modelo com a estrutura predefinida.

Nesta Seção realizaremos uma abordagem não-paramétrica que torna mais flexível a forma de se lidar com a estrutura de f a priori, afinal, o interesse maior é sobre a função f e não sobre parâmetros. Para isto, trabalharemos diretamente no espaço de funções. Para realizarmos inferência sobre f é necessário formalizarmos uma distribuição a priori que envolva hipóteses no espaço de funções sobre a variável latente f. Uma forma de abordarmos esta necessidade é através dos *Processos Gaussianos*.

A distribuição de probabilidade a priori

Definição 2.2. Um Processo Gaussiano é uma coleção de variáveis aleatórias tais que fixado qualquer número finito dentre elas, sua distribuição de probabilidade conjunta é uma normal multivariada.

Em um contexto para funções, sendo nossa base de dados $\mathcal{D} = (X, \mathbf{y})$ como definida no início da Seção 2.1, associamos a cada $\mathbf{x}$, neste caso as colunas de X, uma variável aleatória $f(\mathbf{x})$. Portanto, utilizando a def. 2.2, fixaremos uma distribuição de probabilidade conjunta, uma normal multivariada, para a seguinte coleção finita de variáveis aleatórias $\mathbf{f} = [f(\mathbf{x}_1), ..., f(\mathbf{x}_n)]^\top$ associadas às entradas $[\mathbf{x}_1, ..., \mathbf{x}_n]^\top$:

$$p(\mathbf{f}|X, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{K}) \tag{2.34}$$

onde $\mathbf{m}$ é a média e $\mathbf{K}$ é a matriz de covariância. Um processo Gaussiano é completamente especificado por sua função de média $m(\mathbf{x})$ e sua função de covariância $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$ tal que $K_{ij} = k(\mathbf{x}_i, \mathbf{x}'_j; \boldsymbol{\theta})$. Dado um processo real $f(\mathbf{x})$, definimos estas duas funções da seguinte maneira:

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \end{aligned} \tag{2.35}$$

Um exemplo simples de processo Gaussiano pode ser obtido da Seção 2.1.1 onde utilizamos nosso modelo de regressão linear bayesiano $f(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w}$ com uma priori em $\mathbf{w} \sim \mathcal{N}(\mathbf{0}\,|\Sigma_p)$. Assim, para a média e a covariância:

$$\begin{aligned} \mathbb{E}[f(\mathbf{x})] &= \boldsymbol{\phi}(\mathbf{x})^\top \mathbb{E}[\mathbf{w}] = 0 \\ \mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] &= \boldsymbol{\phi}(\mathbf{x})^\top \mathbb{E}[\mathbf{w}\mathbf{w}^\top]\boldsymbol{\phi}(\mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \Sigma_p \boldsymbol{\phi}(\mathbf{x}') \end{aligned} \tag{2.36}$$

Portanto $f(\mathbf{x})$ e $f(\mathbf{x}')$ são conjuntamente provenientes de uma distribuição normal com média nula e covariância dada por $\boldsymbol{\phi}(\mathbf{x})^\top \Sigma_p \boldsymbol{\phi}(\mathbf{x}')$.

O modelo de erro: A função de verossimilhança

Para realizarmos inferência no espaço de funções, precisamos agora colocar em pauta a informação proveniente dos dados em $\mathcal{D}$, uma vez que a priori já está bem definida. Neste caso é necessário definir a função de verossimilhança de f dado que temos $\mathcal{D}$. O modelo adotado nesta Seção será da seguinte forma:

$$y(\mathbf{x}_i) = f(\mathbf{x}_i) + \epsilon_i, \qquad i = 1, ..., n, \tag{2.37}$$

onde $f : X \mapsto \mathbb{R}$ é nossa função latente e ϵ_i, nesse momento, é uma quantidade aleatória gerada independentemente para cada $i = 1, ..., n$. Note que y depende de f unicamente através de $f(\mathbf{x})$, portanto temos a função de verossimilhança de f dado $\mathcal{D}$ da seguinte forma:

$$p(\mathbf{y}\,|f, \boldsymbol{\psi}) = \prod_{i=1}^{n} p(y_i\,|f(\mathbf{x}_i), \boldsymbol{\psi}) = p(\mathbf{y}\,|\mathbf{f}, \boldsymbol{\psi}) \tag{2.38}$$

onde $\boldsymbol{\psi}$ representa o vetor de hiperparâmetros adicionais à função de verossimilhança além de f, como por exemplo $\boldsymbol{\psi} = (\sigma_n^2)$ na Eq. (2.6).

A distribuição a posteriori e a distribuição preditiva

Para completarmos a inferência de f devemos calcular sua distribuição a posteriori de acordo com o teorema de Bayes:

$$p(\mathbf{f}|\mathcal{D},\boldsymbol{\psi},\boldsymbol{\theta}) \triangleq \frac{p(\mathbf{y}|\mathbf{f},\boldsymbol{\psi})p(\mathbf{f}|X,\boldsymbol{\theta})}{p(\mathcal{D}|\boldsymbol{\psi},\boldsymbol{\theta})} = \frac{\mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{K})}{p(\mathcal{D}|\boldsymbol{\psi},\boldsymbol{\theta})}\prod_{i=1}^{n} p(y_i|f_i,\boldsymbol{\psi}) \tag{2.39}$$

onde $f_i = f(x_i)$.

Uma vez que tenhamos a distribuição a posteriori de $\mathbf{f}$ podemos calcular a distribuição preditiva para novos valores de entrada $\mathbf{x}_*$. Definindo X_* como uma matriz onde suas colunas são novas entradas $\mathbf{x}_*$, e $\mathbf{f}_*$ seu vetor correspondente de valores da função latente, temos que a distribuição preditiva é dada da seguinte maneira:

$$p(\mathbf{f}_*|\mathcal{D},X_*,\boldsymbol{\psi},\boldsymbol{\theta}) = \int p(\mathbf{f}_*|\mathbf{f},X,X_*,\boldsymbol{\theta})p(\mathbf{f}|\mathcal{D},\boldsymbol{\psi},\boldsymbol{\theta})d\mathbf{f} \tag{2.40}$$

A predição então resulta da média de todos os resultados do primeiro termo à direita, a distribuição de probabilidade dos novos pontos condicionada aos dados observados, sendo ponderados pela distribuição à posteriori.

Note que como $f(\mathbf{x})$ é um processo Gaussiano, então temos que a distribuição a priori conjunta de $\mathbf{f}$ e $\mathbf{f}_*$ é dada da seguinte forma:

$$p(\mathbf{f},\mathbf{f}_*|X,X_*,\boldsymbol{\theta}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_*\end{bmatrix}\middle|\begin{bmatrix}\mathbf{m}\\ \mathbf{m}_*\end{bmatrix},\begin{bmatrix}\mathbf{K} & \mathbf{K}_*\\ \mathbf{K}_*^\top & \mathbf{K}_{**}\end{bmatrix}\right) \tag{2.41}$$

onde $\mathbf{K}_* = k(X,X_*)$ e $\mathbf{K}_{**} = k(X_*,X_*)$

Por ser uma distribuição normal multivariada, a propriedade dada pela Eq.(A.3b) nos assegura o seguinte resultado:

$$p(\mathbf{f}_*|\mathbf{f},X,X_*,\boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}_*|\mathbf{m}_* + \mathbf{K}_*^\top\mathbf{K}^{-1}(\mathbf{f}-\mathbf{m}),\ \mathbf{K}_{**} - \mathbf{K}_*^\top\mathbf{K}^{-1}\mathbf{K}_*) \tag{2.42}$$

onde claramente nos mostra que a dependência de $\mathbf{f}_*$ a $\mathbf{f}$ é outra normal multivariada.

Neste momento temos uma bagagem ferramental de inferência Bayesiana por processos gaussianos que nos permite abordar casos variados de regressão. O objetivo final é termos uma distribuição preditiva para novas entradas $\mathbf{x}_*$ condicionada à base de dados de treino $\mathcal{D}$ e às hipóteses sobre o modelo.

Predição com hipótese do modelo sem ruído

Consideraremos uma distribuição a priori Gaussiana simples no espaço de funções onde a média é nula e a matriz de covariância é $\mathbf{K}$:

$$p(\mathbf{f}|X,\boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K}), \tag{2.43}$$

A hipótese mais simples a ser abordada é quando assumimos nosso modelo como sendo determinístico:

$$y = f(x) \tag{2.44}$$

Desta forma não existe incerteza nas observações $\mathbf{y}$ e consequentemente nossa distribuição a posteriori torna-se

$$p(\mathbf{f}|\mathcal{D},\boldsymbol{\psi},\boldsymbol{\theta}) = \delta(\mathbf{f}-\mathbf{y}) \tag{2.45}$$

Portanto a distribuição preditiva, Eq. (2.40), resulta após fazermos $\mathbf{f}=\mathbf{y}$ na Eq. (2.42):

$$p(\mathbf{f}_*|\mathcal{D},X_*,\boldsymbol{\psi},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}_*|\mathbf{K}_*^\top\mathbf{K}^{-1}\mathbf{y},\ \mathbf{K}_{**} - \mathbf{K}_*^\top\mathbf{K}^{-1}\mathbf{K}_*) \tag{2.46}$$

Um exemplo de regressão desconsiderando o ruído, $\sigma_n = 0$, pode ser visto na Fig. 2.4.

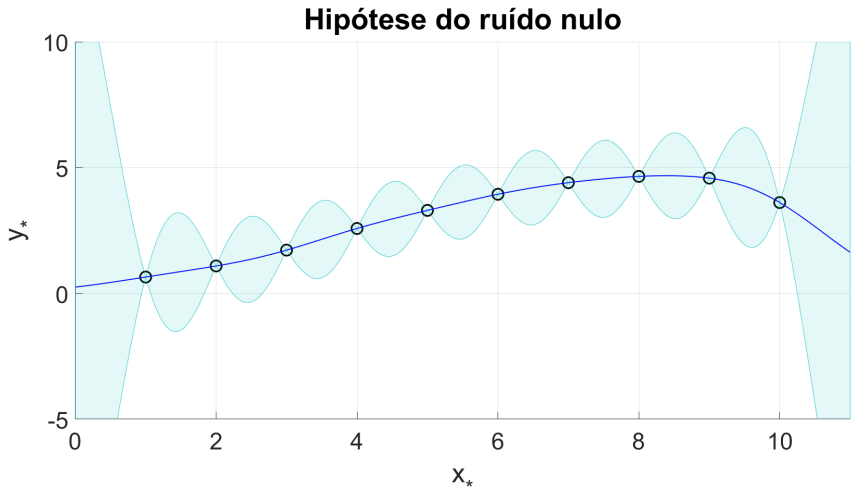


Figura 2.4: Exemplo de regressão com processos Gaussianos utilizando o modelo sem ruído. Na posição das observações, círculos de cor preta, o modelo se torna determinístico, já que não possuímos incertezas nestes pontos. Note que ao se afastar das observações determinísticas, a incerteza começa a aumentar, gerando assim a série de *gomos* que podemos observar.

Predição com hipótese do modelo com ruído gaussiano

Considere nosso modelo tal que as observações realizadas são perturbadas por um erro i.i.d. proveniente de uma distribuição normal:

$$y = f(\mathbf{x}) + \epsilon \quad \text{, onde } \epsilon \sim \mathcal{N}(0, \sigma_n^2) \tag{2.47}$$

Diretamente podemos determinar a função de verossimilhança de f :

$$p(\mathbf{y}\,|f, \boldsymbol{\psi}) = \prod_{i=1}^{n} p(y_i\,|f(\mathbf{x}_i), \boldsymbol{\psi}) = \prod_{i=1}^{n} \mathcal{N}(y_i\,|f(\mathbf{x}_i), \sigma_n^2) = \mathcal{N}(\mathbf{y}\,|\mathbf{f}, \sigma_n^2\mathbf{I}) \tag{2.48}$$

Vamos agora considerar uma distribuição a priori Gaussiana no espaço de funções. Por simplicidade iremos considerar a média nula e a matriz de covariância $\mathbf{K}$:

$$p(\mathbf{f}|X, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}) \tag{2.49}$$

onde $\boldsymbol{\theta}$ é o vetor de hiperparâmetros da função de covariância que estabelece $\mathbf{K}$.

Determinando a distribuição a posteriori no espaço de funções pela regra de Bayes temos:

$$p(\mathbf{f}|\mathcal{D}, \boldsymbol{\psi}, \boldsymbol{\theta}) = \frac{p(\mathbf{y}\,|\mathbf{f}, \boldsymbol{\psi})p(\mathbf{f}|X, \boldsymbol{\theta})}{p(\mathcal{D}\,|\boldsymbol{\psi}, \boldsymbol{\theta})} \propto \mathcal{N}(\mathbf{y}\,|\mathbf{f}, \sigma_n^2\mathbf{I})\mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}) \tag{2.50}$$

Utilizando as identidades A.8 e A.10 no apêndice A, obtemos as seguintes expressões:

$$p(\mathcal{D}\,|\boldsymbol{\psi}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K} + \sigma_n^2\mathbf{I}) \tag{2.51}$$

$$p(\mathbf{f}|\mathcal{D}, \boldsymbol{\psi}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}|\mathbf{K}(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}, (\mathbf{K}^{-1} + \sigma_n^{-2}\mathbf{I})^{-1}) \tag{2.52}$$

ou seja, nossa distribuição a posteriori também é uma distribuição normal, assim como nossa verossimilhança marginal. Agora podemos calcular a distribuição preditiva das novas observações $\mathbf{f}_*$ aplicando a Eq. (2.52) e a Eq. (2.42) na Eq. (2.40). Assim:

$$\begin{aligned} p(\mathbf{f}_*\,|\mathcal{D}, X_*, \boldsymbol{\psi}, \boldsymbol{\theta}) = \int & \mathcal{N}(\mathbf{f}_*\,|\mathbf{K}_*^\top\mathbf{K}^{-1}\mathbf{f},\ \mathbf{K}_{**} - \mathbf{K}_*^\top\mathbf{K}^{-1}\mathbf{K}_*) \\ & \mathcal{N}(\mathbf{f}|\mathbf{K}(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}, (\mathbf{K}^{-1} + \sigma_n^{-2}\mathbf{I})^{-1})d\mathbf{f} \end{aligned} \tag{2.53}$$

Note que podemos rearrumar a Eq. (2.53) da seguinte maneira:

$$p(\mathbf{f}_*|\mathcal{D}, X_*, \boldsymbol{\psi}, \boldsymbol{\theta}) = \int \mathcal{N}(\mathbf{f}_*|\mathbf{M}\mathbf{f} + \mathbf{c},\ \mathbf{L})\mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \mathbf{A})d\mathbf{f} \tag{2.54}$$

onde

$$\begin{array}{rclcrcl} \mathbf{M} &=& \mathbf{K}_*^\top \mathbf{K}^{-1} & & \boldsymbol{\mu} &=& \mathbf{K}(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y} \\ \mathbf{c} &=& \mathbf{m}_* - \mathbf{K}_*^\top \mathbf{K}^{-1}\mathbf{m} & & \mathbf{A} &=& (\mathbf{K}^{-1} + \sigma_n^{-2}\mathbf{I})^{-1} \\ \mathbf{L} &=& \mathbf{K}_{**} - \mathbf{K}_*^\top \mathbf{K}^{-1}\mathbf{K}_* & & & & \end{array} \tag{2.55}$$

Aplicando a identidade A.6a no apêndice A e notando que:

$$\mathbf{K}^{-1} - \mathbf{K}^{-1}(\mathbf{K}^{-1} + \sigma_n^{-2}\mathbf{I})^{-1}\mathbf{K}^{-1} = (\mathbf{K} + \sigma_n^2)^{-1} \tag{2.56}$$

chegamos ao resultado principal:

$$p(\mathbf{f}_*|\mathcal{D}, X_*, \boldsymbol{\psi}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}_*|\mathbf{K}_*^\top(\mathbf{K} + \sigma_n^2)^{-1}\mathbf{y}, \mathbf{K}_{**} - \mathbf{K}_*^\top(\mathbf{K} + \sigma_n^2)^{-1}\mathbf{K}_*) \tag{2.57}$$

De onde derivamos as seguintes equações de predição para modelos com erro Gaussiano utilizando processos gaussianos:

$$\begin{aligned} \bar{\mathbf{f}}_* \triangleq \mathbb{E}[\mathbf{f}_*|\mathcal{D}, X_*, \boldsymbol{\psi}, \boldsymbol{\theta}] &= \mathbf{K}_*^\top(\mathbf{K} + \sigma_n^2)^{-1}\mathbf{y} \\ cov(\mathbf{f}_*) &= \mathbf{K}_{**} - \mathbf{K}_*^\top(\mathbf{K} + \sigma_n^2)^{-1}\mathbf{K}_* \end{aligned} \tag{2.58}$$

Note que ao identificar $\mathbf{K}(X, X_*) = \boldsymbol{\Phi}(X)^\top \Sigma_p \boldsymbol{\Phi}(X)$, obtemos os mesmos resultados que na Seção 2.1.1. De fato, para qualquer conjunto de funções-base podemos computar a função de covariância correspondente fazendo $k(\mathbf{x}_p, \mathbf{x}_q) = \boldsymbol{\phi}(\mathbf{x}_p)^\top \Sigma_p \boldsymbol{\phi}(\mathbf{x}_q)$. O mesmo ocorre ao contrário, para toda função de covariância k, existe uma decomposição em termos de funções-base, podendo esta ser infinita, o que é um indício de uma maior flexibilização na modelagem dos dados.

Podemos verificar também que todas as operações resultaram em uma distribuição normal multivariada. Do ponto de vista computacional, a escolha do erro Gaussiano é fundamental para obtermos fórmulas analíticas e tornar o método da regressão por processos Gaussianos exato. Um exemplo de regressão com ruído Gaussiano pode ser visto na Fig. 2.5.

2.2.1 Funções de covariância

A função de covariância $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$ é fundamental para realizar a predição através de processos gaussianos, já que é ela quem determina a noção de similaridade ou proximidade entre um ponto de treino e um ponto de teste. Através dela é possível explorar as propriedades das funções a serem abordadas pelo processo Gaussiano manipulando seus hiperparâmetros.

A única condição para a determinação de uma função de covariância é que dado um conjunto de pontos de entrada $X = \{\mathbf{x}_i | i = 1, ..., n\}$, a matriz $\mathbf{K}$ com entradas $K_{ij} = k(\mathbf{x}_i, \mathbf{x}'_j; \boldsymbol{\theta})$ deve ser simétrica positiva semidefinida, ou seja, $\mathbf{v}^\top \mathbf{K}\mathbf{v} \geq 0$, $\forall \mathbf{v} \in \mathbb{R}^n$.

As funções de covariância são classificadas entre sendo *estacionárias* ou não-estacionárias.

Funções de covariância estacionárias

Definição 2.3. Uma função de covariância é dita *estacionária* caso seja invariante à translações, ou seja, caso satisfaça

$$k(\mathbf{x}_i, \mathbf{x}'_j; \boldsymbol{\theta}) = D(\mathbf{x}_i - \mathbf{x}'_j; \boldsymbol{\theta}) \tag{2.59}$$

para alguma função de covariância D. Caso contrário ela é dita *não-estacionária*.

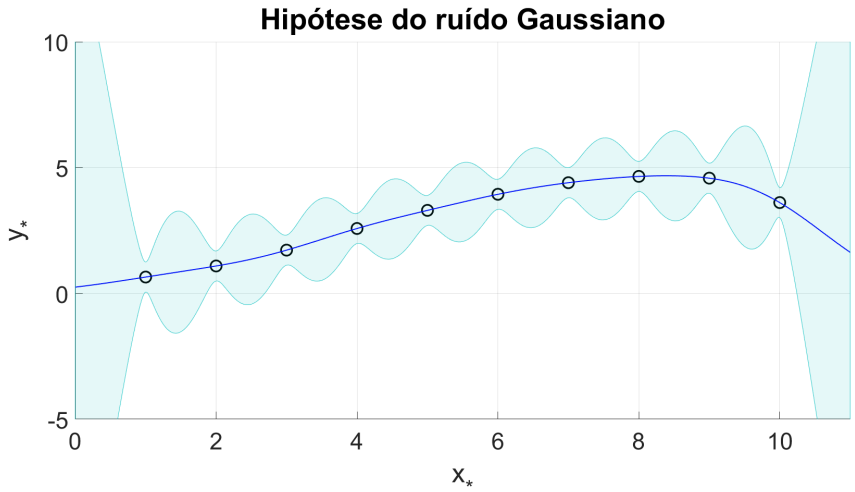


Figura 2.5: Exemplo de regressão com processos Gaussianos utilizando o modelo com ruído Gaussiano. Em contraste com a Fig. 2.4, note que nas posições das observações em círculos pretos, o intervalo de confiança não é nulo, pois está levando em conta a hipótese de que há um ruído Gaussiano presente nas observações.

A seguir veremos alguns exemplos comuns para o caso unidimensional abordado no tratamento de séries temporais :

- **Isotropic Squared Exponential**

$$k_{SE}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\sum_{i=1}^{D} \frac{||x_i - x_i'||^2}{2l^2}\right) \tag{2.60}$$

  Este é o kernel mais popular utilizado hoje em dia. Toda função proveniente de uma priori com este kernel é infinitamente diferenciável, o que sugere uma forte suavidade dos dados como hipótese a priori. Ela contêm apenas dois hiperparâmetros:

  - A escala de comprimento l determina um grau de correlação entre os pontos próximos fazendo com que este se traduza no comprimento das suas ondulações.
  - A variância de saída σ^2 determina a distância média de sua função em relação à média da sua priori. A maioria dos kernels possui este fator de escala vertical.

  Três amostras de uma distribuição a priori utilizando o kernel SE podem ser vistas na Fig. 2.6. Neste exemplo consideramos a variância de saída $\sigma = 1$ e cada amostra possui uma escala de comprimento l diferente. Foram consideradas as seguintes escalas de comprimento: $l = 0.1$, $l = 1$ e $l = 10$.

- **Isotropic Rational Quadratic**

$$k_{RQ}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \sum_{i=1}^{D} \frac{||x_i - x_i'||^2}{2\alpha l^2}\right)^{-\alpha} \tag{2.61}$$

  Este kernel pode ser visto como uma soma infinita de vários kernels SE com diferentes escalas de comprimento l. O parâmetro α determina uma proporção relativa entre a variação de escalas de comprimento longas e curtas. Quando $\alpha \to \infty$, o kernel RQ torna-se idêntico ao kernel SE.

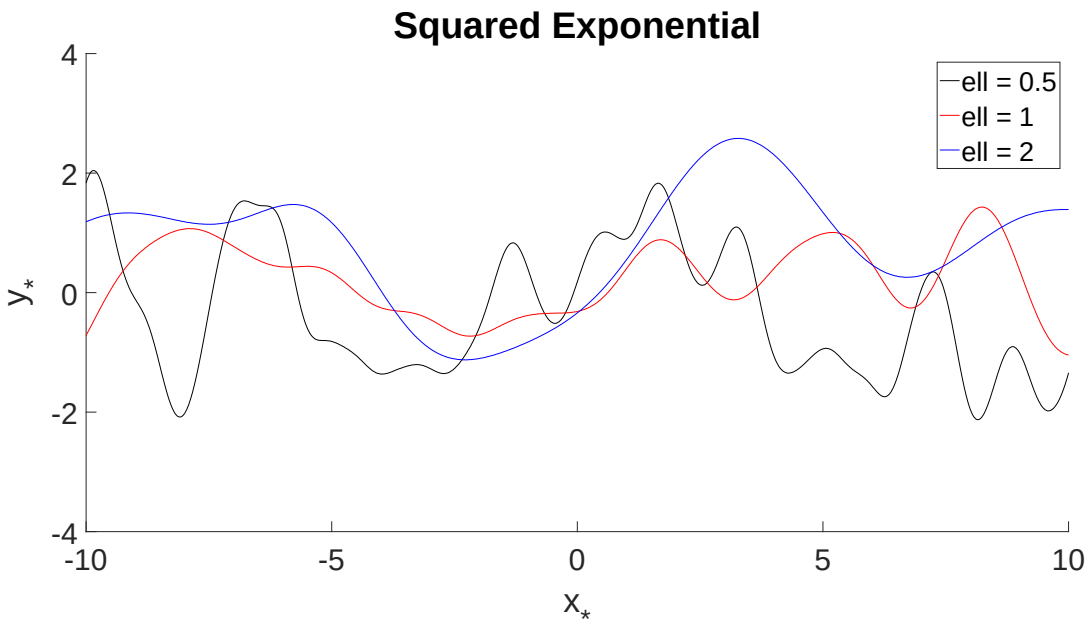


Figura 2.6: Amostras de um processo gaussiano com média nula e kernel SE. Os exemplos foram gerados com variância de saída $\sigma = 1$ e três escalas de comprimento diferentes: $l = 0.1$, $l = 1$ e $l = 10$.

Os parâmetros l e σ possuem propriedades semelhantes aos do kernel SE. Três amostras de uma distribuição a priori utilizando o kernel RQ podem ser vistas na Fig. 2.7. Para este exemplo foram considerados a variância de saída $\sigma = 1$, a escala de comprimento $l = 1$ em todas as amostras. Foi utilizado um valor diferente para α em cada amostra. Neste caso, $\alpha = 0.0001$, $\alpha = 0.01$ e $\alpha = 10$.

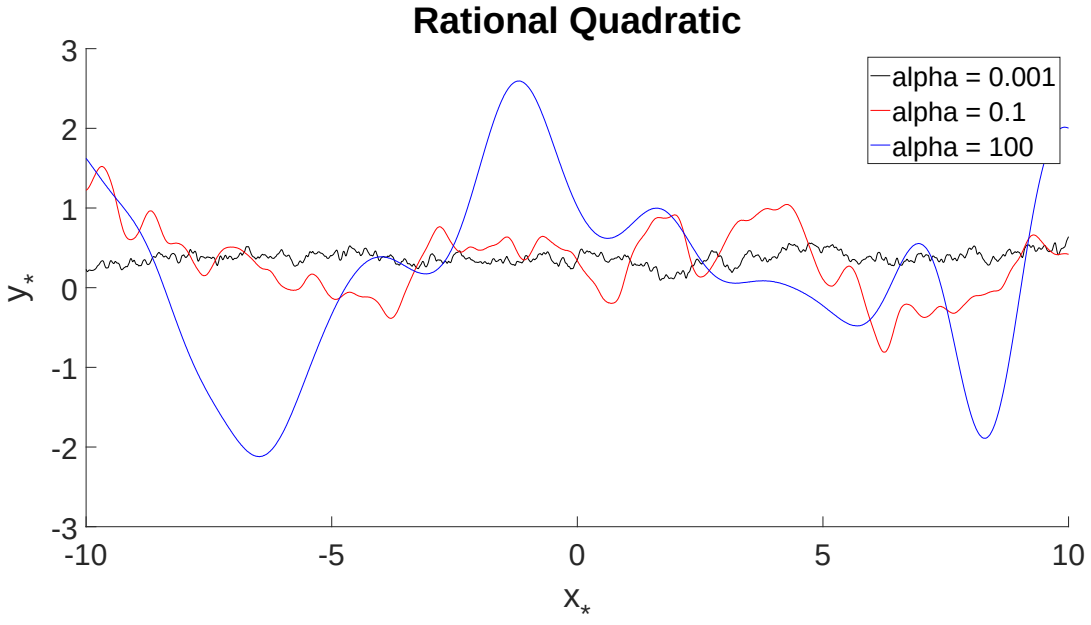


Figura 2.7: Amostras de um processo gaussiano com média nula e kernel RQ. Os exemplos foram gerados com variância de saída $\sigma = 1$, escala de comprimento $l = 1$ e três valores diferentes para α.

- **Matérn Class**

$$k_{\text{Matérn}}(\tau) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\tau}{l} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}\tau}{l} \right) \quad \text{, onde } \tau = \sqrt{(\mathbf{x}-\mathbf{x}')^\top (\mathbf{x}-\mathbf{x}')} \tag{2.62}$$

Os parâmetros σ, ν e l são positivos e K_ν é uma função de Bessel modificada [7, sec. 9.6, pag 374]. As funções provenientes de um processo gaussiano com este kernel são k-diferenciáveis se e somente se $\nu > k$. A função de covariância Matérn assume uma forma bastante simples quando $\nu = p + 1/2$, onde p é um inteiro não-negativo. Para os casos onde $p \in 1, 2, 3$ temos:

$$k_{\nu=1/2}(\tau) = \sigma^2 \exp\left(-\frac{\tau}{l}\right) \tag{2.63}$$

$$k_{\nu=3/2}(\tau) = \sigma^2 \left(1 + \frac{\sqrt{3}\tau}{l}\right) \exp\left(-\frac{\sqrt{3}\tau}{l}\right) \tag{2.64}$$

$$k_{\nu=3/2}(\tau) = \sigma^2 \left(1 + \frac{\sqrt{5}\tau}{l} + \frac{5\tau^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}\tau}{l}\right) \tag{2.65}$$

Raramente é utilizado $\nu \geq 7/2$, pois é difícil obter conhecimento a priori sobre a existência de ordens superiores na diferenciabilidade apenas tendo em mãos amostras finitas com ruído da função latente. Esta função de covariância também assume a foma do kernel SE quando $\nu \to \infty$.

Um exemplo utilizando cada uma das três formas da função de covariância Matérn discutidas aqui pode ser visto na Fig. 2.8. Neste caso foram considerados a variância de saída $\sigma = 1$ e a escala de comprimento $l = 1$ para a geração de todas as amostras.

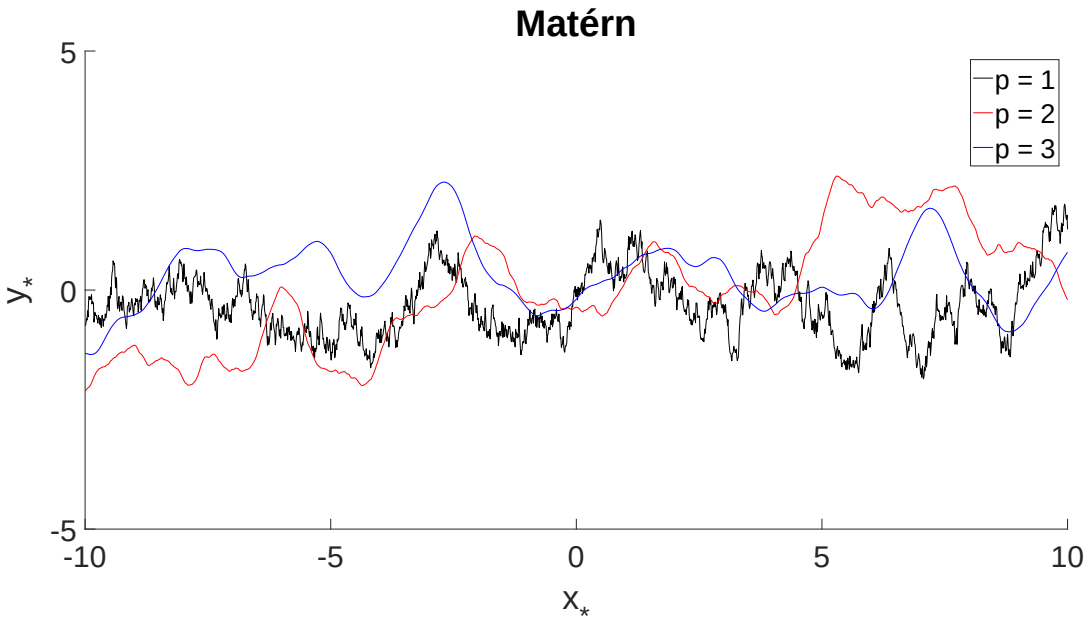


Figura 2.8: Amostras de um processo gaussiano com média nula e kernel Matérn. Os exemplos foram gerados com variância de saída $\sigma = 1$, escala de comprimento $l = 1$ e três valores diferentes para ν.

Funções de covariância não-estacionárias

- **Produto escalar**

$$k(\mathbf{x}, \mathbf{x}') = \sigma_0^2 + \mathbf{x}^\top \mathbf{x}' \tag{2.66}$$

As amostras do processo gaussiano que utiliza este kernel são puramente funções lineares. Utilizar este kernel em uma regressão com processos gaussianos é equivalente a fazer uma regressão linear. Uma forma mais geral ocorre utilizando uma matriz de covariância Σ_p:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_0^2 + \mathbf{x}^\top \Sigma_p \mathbf{x}' \tag{2.67}$$

Para problemas de regressão, este kernel é raramente utilizado, pois a variância da distribuição a priori cresce rapidamente com $|\mathbf{x}|$, para $|\mathbf{x}| > 1$.

Combinação de funções de covariância

Uma maneira de se obter novas funções de covariância é pela combinação de outras funções de covariância como apresentado na Fig. 2.9. Considere k_1 e k_2 funções de covariância. A função k_3 gerada através das seguintes propriedades também é uma função de covariância:

- **Soma**: $k_3(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$.
- **Produto**: $k_3(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$.
- **Periodização**: $k_3(\mathbf{x}, \mathbf{x}') = k_1(u(\mathbf{x}), u(\mathbf{x}'))$, onde $u(\mathbf{x}) = \begin{bmatrix}\sin(2\pi \frac{\mathbf{x}}{p}) \\ \cos(2\pi \frac{\mathbf{x}}{p})\end{bmatrix}$, para $p > 0$.

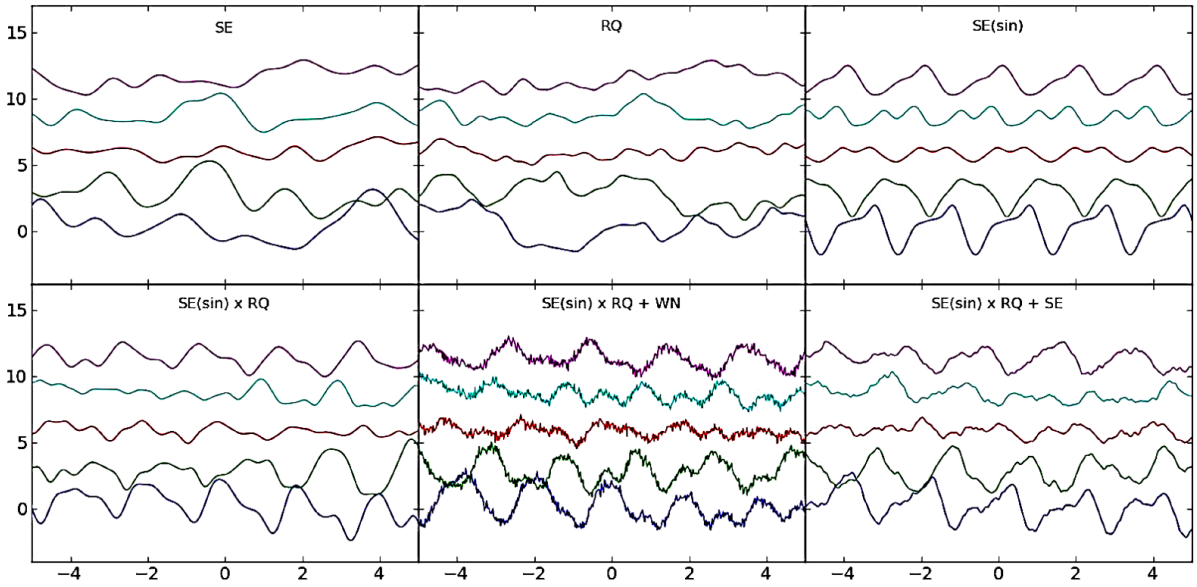


Figura 2.9: Amostras aleatórias de processos Gaussianos com diferentes kernels. *Fonte: Roberts, 2012.*

A escolha do kernel que melhor possa explicar a estrutura inerente dos dados observados é uma das questões cruciais na modelagem de dados com processos gaussianos. Não existe um kernel padrão que possa ser aplicado genericamente. Geralmente a construção do kernel é uma atividade baseada em conclusões tiradas por quem está modelando os dados em questão. Alguns trabalhos recentes [21, 20] propõem alternativas para a construção de uma matriz de covariância, como a utilização de redes neurais para inferir uma matriz de covariância baseada nos dados ou como a automatização da escolha da função de covariância através da análise de várias combinações geradas com operações simples como a soma ou o produto. A metodologia proposta no capítulo seguinte aborda este problema utilizando uma pilha de possíveis funções de covariância para o tratamento específico de alguns tipos de trechos comprometidos na série temporal analisada. Vários outros tipos de funções de covariância podem ser explorados não somente com as combinações aqui descritas. Para mais funções de covariância e mais propriedades, ver [11].

2.2.2 Seleção de modelos e ajuste dos hiperparâmetros

Geralmente, dado uma função de covariância escolhida, não sabemos especificar exatamente qual a configuração de seus hiperparâmetros $\boldsymbol{\theta}$ para melhor abordar as características da estrutura dos dados utilizados. O mesmo problema ocorre com a configuração dos hiperparâmetros $\boldsymbol{\psi}$ da função de verossimilhança. O que pode-se fazer é escolher um valor que traduza alguma subjetividade dos dados, como, por exemplo, o período em dados periódicos ou a escala de comprimento ao se verificar algum grau de suavidade na curva. Porém gostaríamos de "aprender" estes hiperparâmetros utilizando os dados. Este processo de aprendizado também pode ser abordado pela ótica da inferência Bayesiana. A partir de agora iremos nos referir aos hiperparâmetros $\boldsymbol{\theta}$ como sendo **todos** os hiperparâmetros do modelo Bayesiano, o que inclui os hiperparâmetros da função de covariância e da função de verossimilhança, ou seja, $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\text{cov}}, \boldsymbol{\psi})$.

A hierarquia dos métodos bayesianos

Nas seções anteriores vimos uma certa hierarquia nos modelos propostos, onde os hiperparâmetros mantêm um certo controle sobre os parâmetros. É comum utilizarmos uma especificação hierárquica de modelos. Note que ao mudarmos os valores dos parâmetros e hiperparâmetros, estamos mudando de modelo. Assim, para realizarmos uma seleção de modelos, iremos dividir nosso problema em três níveis de inferência que irão estabelecer as incertezas nos parâmetros $\mathbf{w}$, nos hiperparâmetros $\boldsymbol{\theta}$ e nas possíveis estruturas de modelos $\mathcal{H}_i$ consideradas:

- **1° nível de inferência**

  Este é o nível considerado mais baixo e onde trataremos os parâmetros $\mathbf{w}$ que, por exemplo, podem ser os parâmetros de um modelo linear. A incerteza sobre os parâmetros será representada através da distribuição a posteriori sobre $\mathbf{w}$ obtida através do teorema de Bayes:

  $$p(\mathbf{w}|\mathbf{y}, X, \boldsymbol{\theta}, \mathcal{H}_i) = \frac{p(\mathbf{y}|X, \mathbf{w}, \mathcal{H}_i)p(\mathbf{w}|\boldsymbol{\theta}, \mathcal{H}_i)}{p(\mathbf{y}|X, \boldsymbol{\theta}, \mathcal{H}_i)} \tag{2.68}$$

  onde $p(\mathbf{y}|X, \mathbf{w}, \mathcal{H}_i)$ é a *função de verossimilhança dos parâmetros* e $p(\mathbf{w}|\boldsymbol{\theta}, \mathcal{H}_i)$ é a *distribuição a priori* dos parâmetros. A priori representa em si nosso conhecimento sobre os parâmetros do modelo antes de se observar os dados. Caso tenhamos pouco conhecimento sobre os parâmetros, a priori será construída de forma a refletir uma ampla incerteza sobre os parâmetros.

  A distribuição a posteriori resulta da combinação da priori com os dados através da função de verossimilhança. A constante de normalização no denominador da Eq. (2.68) $p(\mathbf{y}|X, \boldsymbol{\theta}, \mathcal{H}_i)$ é independente dos parâmetros e é denominada como *verossimilhança marginal* ou *evidência*, e podemos obtê-la através da seguinte forma:

  $$p(\mathbf{y}|X, \boldsymbol{\theta}, \mathcal{H}_i) = \int p(\mathbf{y}|X, \mathbf{w}, \mathcal{H}_i)p(\mathbf{w}|\boldsymbol{\theta}, \mathcal{H}_i)d\mathbf{w} \tag{2.69}$$

- **2° nível de inferência**

  Neste nível trataremos de maneira análoga ao nível anterior os hiperparâmetros $\boldsymbol{\theta}$. Novamente aplicamos o teorema de Bayes para obtermos a posteriori dos hiperparâmetros:

  $$p(\boldsymbol{\theta}|\mathbf{y}, X, \mathcal{H}_i) = \frac{p(\mathbf{y}|X, \boldsymbol{\theta}, \mathcal{H}_i)p(\boldsymbol{\theta}|\mathcal{H}_i)}{p(\mathbf{y}|X, \mathcal{H}_i)} \tag{2.70}$$

  onde $p(\boldsymbol{\theta}|\mathcal{H}_i)$ é a *priori dos hiperparâmetros*. Note a conexão entre os 1° e 2° níveis através da verossimilhança marginal dos parâmetros, Eq. (2.69), que neste momento assume o papel de *função de verossimilhança dos hiperparâmetros*. Neste caso a constante de normalização é dada da seguinte maneira:

  $$p(\mathbf{y}|X, \mathcal{H}_i) = \int p(\mathbf{y}|X, \boldsymbol{\theta}, \mathcal{H}_i)p(\boldsymbol{\theta}|\mathcal{H}_i)d\boldsymbol{\theta} \tag{2.71}$$

- **3° nível de inferência**

  Este é o nível mais alto, onde serão calculadas a posteriori para os modelos. Geralmente possuímos uma quantidade finita de modelos $\mathcal{H}_i$ sob consideração. Assim, analogamente aos níveis anteriores:

$$p(\mathcal{H}_i|\mathbf{y}, X) = \frac{p(\mathbf{y}|X, \mathcal{H}_i)p(\mathcal{H}_i)}{p(\mathbf{y}|X)} \tag{2.72}$$

  onde $p(\mathbf{y}|X) = \sum_i p(\mathbf{y}|X, \mathcal{H}_i)p(\mathcal{H}_i)$.

Com esta hierarquia fixada, podemos discutir mais claramente as dificuldades encontradas para a realização dos cálculos destes métodos, a seleção de modelos e como podemos ajustar os hiperparâmetros para um modelo baseado em processos gaussianos.

As dificuldades computacionais

Grande parte dos métodos de inferência Bayesiana aqui discutidos necessitam do cálculo de integrais. Dependendo de como os modelos são estabelecidos, estas integrais podem ter ou não soluções analíticas, e normalmente a resolução delas ocorre através de soluções aproximadas, como por exemplo o método de Monte Carlo. Ao realizarmos a implementação destes métodos, podemos encontrar dificuldades maiores especificamente no cálculo da evidência do modelo, dada pela Eq. (2.71). A importância deste termo ficará clara ao discutirmos a seleção de modelos. Caso esta integral seja não-analítica, podemos resolvê-la utilizando a aproximação de Laplace, [3] cap.27, que apoia-se numa aproximação quadrática através da expansão local do logaritmo do integrando em torno do ponto onde este é máximo. Uma abordagem mais barata que a aproximação de Laplace é a maximização da verossimilhança marginal, dada pela Eq. (2.69), em relação aos hiperparâmetros $\boldsymbol{\theta}$. Esta abordagem é conhecida como estimação da *máxima verossimilhança tipo-II*, por ser em relação aos hiperparâmetros. Como visto nas seções anteriores, esta abordagem pode acabar nos remetendo ao fenômeno de over-fitting, especialmente caso tenhamos um número considerável de hiperparâmetros. A aproximação de Laplace será boa caso a posteriori dos hiperparâmetros seja um pico proeminente concentrando boa parte da informação em uma certa região, o que ocorre com mais frequência para os hiperparâmetros do que para os parâmetros [13].

Temos então uma forma de obter a evidência do modelo, Eq. (2.71), e o hiperparâmetro[3] $\boldsymbol{\theta}_{ML}$ que maximiza a verossimilhança marginal dos parâmetros, Eq. (2.69). Caso tenhamos alguma informação a priori sobre os hiperparâmetros $\boldsymbol{\theta}$, também podemos realizar a técnica de *máximo a posteriori* para estimar pontualmente o hiperparâmetro $\boldsymbol{\theta}_{MAP}$ através da maximização da distribuição a posteriori dos hiperparâmetros, Eq.(2.70).

A seleção de modelos bayesiana

A posteriori do modelo $\mathcal{H}_i$, Eq. (2.72), quantifica a noção do quão provável é $\mathcal{H}_i$, dentro de seu espaço discreto e finito de modelos, dado o conhecimento dos dados. De um outro ponto de vista, podemos considerar esta posteriori como um julgamento do modelo pelos dados. Através dela é possível escolhermos o modelo mais provável para o tratamento dos dados. Porém raramente temos uma informação a priori sobre os modelos, então construímos uma priori tão *achatada* de tal forma que não favoreça nenhum modelo em particular. Assim, a posteriori acaba sendo proporcional à evidência do modelo a menos de uma constante multiplicativa, como podemos observar na Eq. (2.72). Portanto, dado dois modelos $\mathcal{H}_1$ e $\mathcal{H}_2$, podemos compará-los da seguinte maneira:

$$\frac{p(\mathcal{H}_1|\mathbf{y}, X)}{p(\mathcal{H}_2|\mathbf{y}, X)} = \frac{p(\mathbf{y}|X, \mathcal{H}_1)}{p(\mathbf{y}|X, \mathcal{H}_2)} \tag{2.73}$$

O que distingue a abordagem Bayesiana para a seleção de modelos de outras abordagens baseadas em otimização é a forma como a verossimilhança marginal assimila todas as incertezas relativas aos parâmetros e hiperparâmetros pelos níveis de inferência. Uma propriedade da verossimilhança marginal [14] é

[3] ML - Maximum Likelihood

automaticamente incorporar um equilíbrio entre o ajuste do modelo e sua complexidade[4], evitando assim o fenômeno de over-fitting, [3, cap. 27]. Isto é a razão pela qual a regressão Bayesiana é robusta quanto à seleção de modelos.

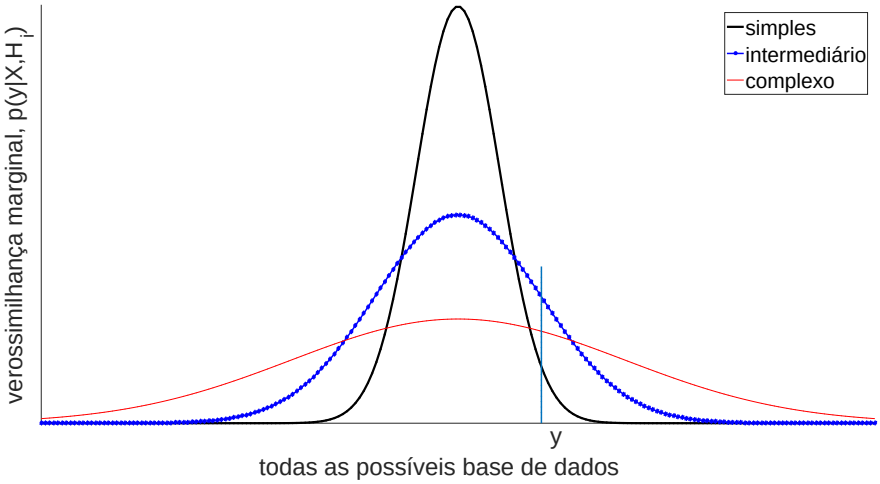


Figura 2.10: Nesta figura podemos ver a verossimilhança marginal $p(\mathbf{y}|X, \mathcal{H}_i)$ para três modelos com complexidade distintas. A quantidade de dados n e os dados de entrada X estão fixos, assim uma base de dados é representada apenas pelos vetores $\mathbf{y}$. Assim, o eixo horizontal representa todas as possibilidades de vetores y de dimensão n. Note a normalização das funções devido à verossimilhança marginal ser também uma distribuição de probabilidade quando em relação a $\mathbf{y}$. Para o caso particular de $\mathbf{y}$ mostrado na figura, podemos ver que a verossimilhança marginal intermediária é a que apresenta maior valor com relação às outras, indicando assim que o modelo que melhor captura a estrutura da base de dados é o modelo intermediário.

Na Fig. 2.10 podemos observar o comportamento da verossimilhança marginal para três modelos com complexidades distintas. Considere uma base de dados com n elementos de forma que $\mathcal{D}(\mathbf{y}) = (X, \mathbf{y})$, onde n e X são quantidades fixas. O eixo horizontal representa todos os possíveis vetores $\mathbf{y}$ da base de dados e o eixo vertical representa a verossimilhança marginal $p(\mathbf{y}|X, \mathcal{H}_i)$. Um modelo simples pode somente descrever o comportamento dos dados de uma certa região limitada, porém como a verossimilhança marginal é uma distribuição de probabilidade em $\mathbf{y}$, ela obrigatoriamente é normalizada. Então, as bases de dados que o modelo simples consegue descrever resultam em um alto valor da verossimilhança marginal. O contrário ocorre para o modelo complexo, que consegue abordar uma extensa gama de bases de dados maior que o modelo simples, porém sua verossimilhança marginal não atinge valores altos como para o modelo simples. Um exemplo de modelo simples é o linear e de modelo complexo um processo gaussiano com vários hiperparâmetros. Nesta figura podemos ver claramente o porquê de a verossimilhança marginal não favorecer modelos complexos que se adaptam além da conta aos dados. Este equilíbrio que ocorre automaticamente através dos métodos de inferência Bayesianos é conhecido como a *Navalha de Occam*, [14, 15], cujo princípio implica que não devemos tornar algo mais complexo além do necessário.

O ajuste dos hiperparâmetros

Vimos que os métodos de inferência Bayesiana possuem uma enorme dificuldade ao que tange a resolução de integrais. Porém, como pudemos ver na Seção 2.2, para o caso em que são empregados modelos de regressão via processos Gaussianos aliados à hipótese de ruído Gaussiano, as integrais sobre os parâmetros possuem solução analítica e ao mesmo tempo os modelos são bastante flexíveis. Como o modelo

[4]Nas seções anteriores foi utilizado o termo "flexibilidade"para uma melhor interpretação.

de processos Gaussianos é uma abordagem não-paramétrica, pode soar estranho em um primeiro momento conversarmos sobre os parâmetros do nosso modelo. Aplicando os métodos Bayesianos de forma hierárquica, vemos que o primeiro nível de inferência ocorre através das equações (2.50) e (2.51), onde consideramos $\mathbf{f}$ como sendo os *parâmetros* do nosso modelo de regressão.

Na subseção anterior observa-se que a verossimilhança marginal dos hiperparâmetros, Eq. (2.69), possui uma importância fundamental tanto para a seleção de modelos como para o ajuste dos hiperparâmetros. Sua expressão analítica no contexto da regressão discutida aqui assume a seguinte forma:

$$\log p(\mathbf{y}|X,\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^\top \boldsymbol{K}_y^{-1}\mathbf{y} - \frac{1}{2}\log\det\boldsymbol{K}_y - \frac{n}{2}\log 2\pi \tag{2.74}$$

onde $\boldsymbol{K}_y = \boldsymbol{K}_f + \sigma_n^2 I$ é a matriz de covariância para as observações perturbadas $\mathbf{y}$ e $\boldsymbol{K}_f$ neste caso é a matriz de covariância para a função latente $\mathbf{f}$.

Cada termo da verossimilhança marginal possui um determinado papel na regressão dos dados:

- $-\frac{1}{2}\mathbf{y}^\top \boldsymbol{K}_y^{-1}\mathbf{y}$ é o termo que denominaremos *ajuste dos dados* por ser o único termo que contêm as observações $\mathbf{y}$.
- $\frac{1}{2}\log\det\boldsymbol{K}_y$ é o termo de *penalização da complexidade* que depende somente da matriz de covariância e dos dados de entrada X.
- $\frac{n}{2}\log 2\pi$ é uma constante de normalização.

Na Fig. 2.11(a) podemos ver o comportamento dos termos relevantes da verossimilhança marginal, bem como o seu resultado final. O ajuste dos dados decresce monotonamente com o aumento da escala de comprimento, pois o modelo se torna cada vez mais menos flexível para explicar os dados, como podemos ver na Fig. 2.12(c). Isto é diretamente traduzido pelo termo negativo da penalidade de complexidade, que pelo mesmo motivo aumenta junto com o crescimento da escala de comprimento. O ponto de máximo da verossimilhança marginal ocorre em torno de 1 e é justamente onde conseguimos um equilíbrio entre o ajuste do modelo e sua complexidade, como exemplificado na Fig. 2.12(a). Para comprimentos de escala maiores que 1, a verossimilhança marginal decresce rapidamente, note que o gráfico está em escala logarítmica, devido ao modelo começar a explicar bem menos os dados em questão. Para comprimentos de escala bem menores que 1, os modelos começam a ganhar tanta flexibilidade que o fenômeno de over-fitting começa a aparecer, veja a Fig. 2.12(b) e note a alta variância nos intervalos de predição.

Na Fig. 2.11(b) é exposta a dependência entre o log da verossimilhança marginal e a escala de comprimento para três casos diferentes onde se variam a quantidade de dados na base de treino. Observa-se um comportamento comum que é a proeminência da verossimilhança marginal aumentando em uma região próxima de 1 quando aumentamos a quantidade de dados de treino. Note que ao termos poucos dados de treino, o gráfico da verossimilhança marginal torna-se quase que constante em boa parte do trecho considerado, nos indicando que o modelo consegue explicar os poucos dados com uma gama grande de valores para a escala de comprimento, neste caso os valores intermediários e os bem pequenos. Com a adição de mais dados, podemos ver que a verossimilhança marginal começa a penalizar de forma mais agressiva modelos mais flexíveis, com escala de comprimento bem pequeno, e modelos muito simplistas, com escala de comprimento bem grande.

Para que possamos ajustar os hiperparâmetros através da maximização da verossimilhança marginal, utilizamos as equações das derivadas parciais da verossimilhança marginal em relação aos hyperparâmetros:

$$\frac{\partial\mathcal{L}}{\partial\theta_j} = \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{K}_y^{-1}\frac{\partial K_y}{\partial\theta_j}\boldsymbol{K}_y^{-1}\boldsymbol{y} - \frac{1}{2}\,\text{Traço}\left(\boldsymbol{K}_y^{-1}\frac{\partial K_y}{\partial\theta_j}\right) \tag{2.75}$$

A complexidade da computação da verossimilhança marginal na Eq. (2.74) é dominada pela inversão da matriz $\boldsymbol{K}_y$. Os métodos comumente utilizados para inversão de matrizes simétricas positivas semi-definidas exigem uma complexidade computacional de $\mathcal{O}(n^3)$ para a inversão de uma matriz $n \times n$. Uma

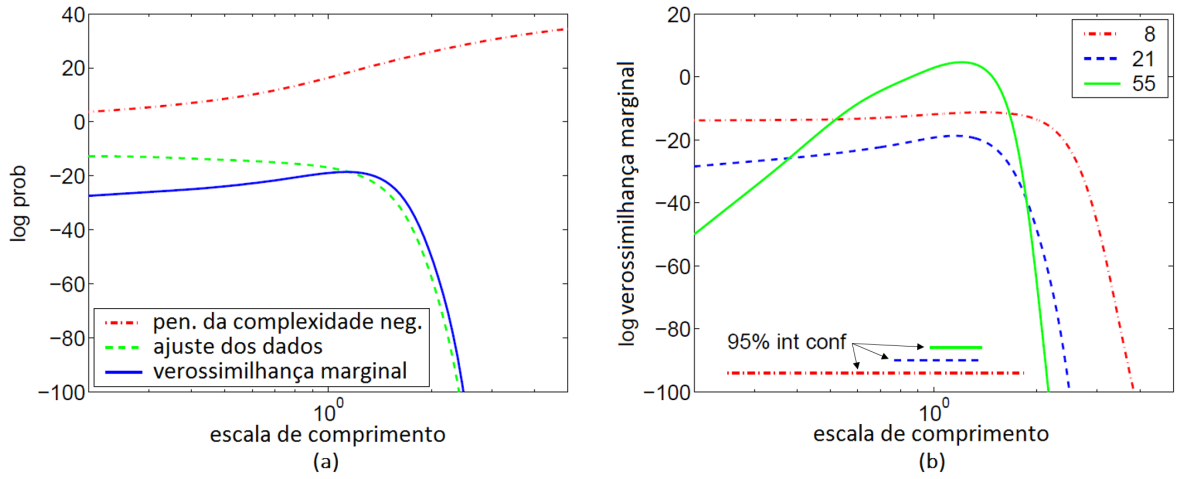


Figura 2.11: A Fig.(a) mostra a decomposição do logaritmo da verossimilhança marginal nos termos: ajuste de dados e1 penalização de complexidade, como função da escala de comprimento. Os dados de treino foram gerados por um processo Gaussiano com função de covariância SE e hiperparâmetros $\boldsymbol{\theta} = (l, \sigma, \sigma_n) = (1, 1, 0.1)$, como na Fig. 2.12, e estamos ajustando aqui apenas o hiperparâmetro de escala de comprimento l, pois os outros foram fixados de acordo com o processo de geração dos dados sintéticos. A Fig.(b) mostra o logaritmo da verossimilhança marginal como função da escala de comprimento para diferentes tamanhos da base de dados de treino. Nesta figura também se encontra o intervalo de confiança de 95% *a posteriori* para a escala de comprimento l em três bases de dados diferentes. Este intervalo é gerado através da Eq. (2.70), onde é considerada uma priori não informativa sobre o hiperparâmetro em questão. Assim podemos observar que a distribuição a posteriori para o hiperparâmetro l acompanha a formação do pico da verossimilhança marginal após o aumento da base de dados. *Fonte: Rasmussen, 2006.*

vez tendo $\boldsymbol{K}_y^{-1}$, a computação das derivadas na Eq. (2.75) exigem apenas uma complexidade computacional de $\mathcal{O}(n^2)$ por hiperparâmetro. Portanto o gargalo computacional para o cálculo das derivadas parciais da verossimilhança marginal é relativamente pequeno, nos permitindo tirar vantagem através da utilização de otimizadores baseados no gradiente.

A Fig. 2.13 mostra um exemplo da verossimilhança marginal como uma função dos hiperparâmetros escala de comprimento e desvio padrão do erro para a função de covariância *squared exponential*, como na Eq. (2.60). A variância de saída está configurada como $\sigma = 1$. A verossimilhança marginal possui claramente um máximo em torno em torno da região com os valores para os quais foram gerados os pontos de treino utilizando um processo Gaussiano. Duas regiões interessantes ocorrem nesse gráfico. Uma delas surge quando o desvio padrão do erro $\sigma_n = 1$, onde podemos observar que a verossimilhança marginal acaba se tornando quase independente da escala de comprimento. Isto é resultado da interpretação dos dados pelo modelo como sendo em sua maioria ruído. A outra região ocorre quando a escala de comprimento $l = 0.4$, onde a verossimilhança marginal torna-se quase independente do desvio padrão do erro, o que faz com que o modelo interpole exatamente os dados, sofrendo o fenômeno de overfitting. Porém apesar disso, a verossimilhança marginal desfavorece todas essas configurações devido ao posicionamento de seu máximo.

Na Fig. 2.14 podemos ver que a verossimilhança marginal pode possuir vários máximos locais, neste caso dois. Cada máximo local corresponde a uma interpretação particular dos dados que é apresentada nas outras duas figuras subsequentes. Um dos máximos corresponde à um modelo mais complexo com menos ruído, enquanto que o outro a um modelo mais simples com um ruído maior. Com apenas 7 dados de treino, não é possível para o modelo acertadamente rejeitar uma das duas possibilidades. O valor numérico da verossimilhança marginal para o modelo mais complexo é por volta de 60% maior que

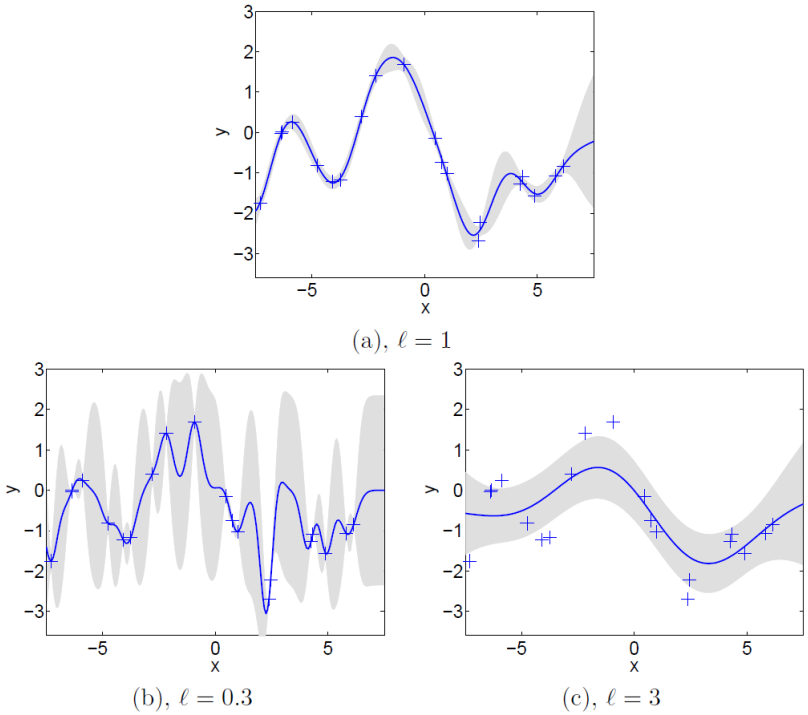


Figura 2.12: Na Fig.(a) os dados, em formato de cruz, são gerados de um processo Gaussiano com função de covariância SE e hiperparâmetros $\boldsymbol{\theta} = (l, \sigma, \sigma_n) = (1, 1, 0.1)$. Realizando uma regressão via processos Gaussianos utilizando a mesma função de covariância com estes mesmos hiperparâmetros, obtemos um intervalo de confiança de 95%, região sombreada, para a função latente f. As Figuras (b) e (c) também são resultantes desta mesma regressão, porém com os respectivos hiperparâmetros: (0.3, 1.08, 0.00005) e (3.0, 1.16, 0.89). *Fonte: Rasmussen, 2006.*

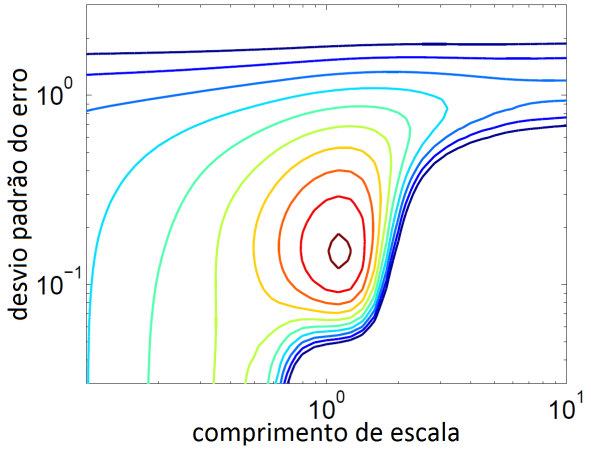


Figura 2.13: Curvas de nível do logaritmo da verossimilhança marginal como função da escala de comprimento e do desvio padrão do erro Gaussiano para a mesma base de dados de treino da Fig. 2.12. A variância de saída σ da função de covariância SE foi configurada como $\sigma^2 = 1$. Os valores otimizados estão perto dos hiperparâmetros utilizados na geração dos dados. *Fonte: Rasmussen, 2006.*

para o modelo mais simples. De acordo com o formalismo Bayesiano, deveria-se ponderar as predições realizadas, pelas diferentes soluções encontradas, utilizando a distribuição a posteriori dos parâmetros.

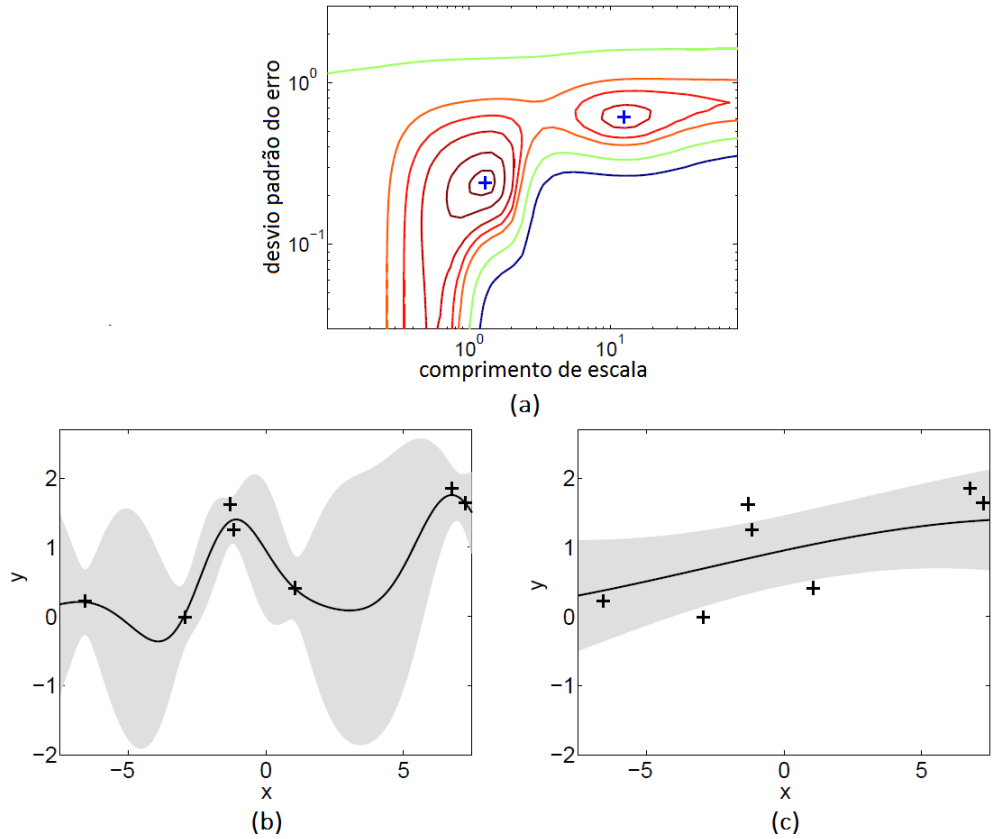


Figura 2.14: A Fig.(a) mostra a verossimilhança marginal como função dos hiperparâmetros l, a escala de comprimento, e σ_n^2, o desvio padrão do erro, onde a variância de saída $\sigma^2 = 1$, para uma base de dados com 7 pontos de treino mostrados nas figuras (b) e (c). Podemos observar que há dois máximos locais indicados com '+': o máximo global possui pouco ruído e uma escala de comprimento pequena e o máximo local possui muito ruído e uma escala de comprimento grande. As duas soluções com médias e intervalos de confiança de 95% de suas respectivas distribuições preditivas são apresentadas nas figuras (b) e (c) respectivamente. De fato, os dados de treino foram gerados por um processo Gaussiano com $\boldsymbol{\theta} = (l, \sigma^2, \sigma_n^2) = (1, 1, 0.1)$. *Fonte: Rasmussen, 2006.*

Entretanto, utilizando uma base de dados de treino com uma quantidade considerável de pontos de treino geralmente acarreta em um máximo local com uma magnitude muito maior que a de outros máximos locais, tornando a ponderação ditada pelo método Bayesiano um passo que pode ser descartado. Porém deve-se tomar cuidado ao ditar o hiperparâmetro inicial para que a otimização não termine em máximos locais com interpretações ruins sobre os dados.

Capítulo 3

Algoritmo para tratamento de dados do PNCT

Este capítulo é dedicado ao desenvolvimento de uma metodologia de imputação de séries temporais derivadas da área de Transporte. Para isto será utilizada a técnica de regressão Bayesiana através de processos Gaussianos vista no capítulo anterior. O problema será posto de forma mais clara para que possamos entender melhor o encadeamento das exigências com a solução proposta.

3.1 O problema

O PNCT possui 320 postos de coleta de dados de tráfego distribuídos em sua malha rodoviária federal. Cada posto possui um equipamento de contagem permanente de tráfego que além de registrar o evento da movimentação de um veículo automotor no local analisado, este também realiza a classificação deste veículo em diferentes categorias como, por exemplo: carro, moto, ônibus, caminhões de dois eixos, etc.

Um evento é determinado pelos seguintes componentes de informação:

- **Data e horário**: Instante exato do registro efetuado sendo gravados a data e o horário em formato AAAA-MM-DD HH:MM:SS. Por exemplo: 2015-06-09 12:16:01 .
- **Faixa**: Especificação de qual faixa o evento ocorreu. Por exemplo: 3.
- **Sentido da rodovia**: Especificação do sentido pré estabelecido na rodovia federal analisada, podendo este ser crescente (C) ou decrescente (D).
- **Classe**: Código pré estabelecido da classe em que o veículo no instante do registro é pertencente. No total são 11 categorias pré-estabelecidas.
- **Velocidade**: Velocidade do veículo no instante do registro em km/h.
- **Peso**: Peso total bruto do veículo no instante do registro.
- **Eixos**: Quantidade de eixos do veículo analisado.

Todos os eventos são registrados no banco de dados do Departamento Nacional de Infraestrutura de Transportes, DNIT. Assim, podemos escolher extrair de maneira simples e eficiente um arquivo de texto .txt contendo todos os eventos de um equipamento específico desde o início de suas atividades.

Como o interesse é apenas na frequência de veículos na rodovia, realiza-se um pré-tratamento com a finalidade de gerar um arquivo de texto .csv[1] contendo a *frequência de veículos a cada 15 min.* Denominaremos um trecho de 15 min como um ***bin*** e a frequência de veículos nesse trecho como ***volume de tráfego***. A disposição dos dados neste arquivo é de forma matricial onde cada entrada é o volume de

[1] csv: comma-separated values.

tráfego em um determinado bin. Então este arquivo de texto em formato .csv é organizado como uma matriz com $(1 + 11)\times 96$ colunas e cada linha representando 24h ($= 96$ bins) de atividade do equipamento:

$$CSV = [T \quad C_1 \quad ... \quad C_{11}] \tag{3.1}$$

onde $\dim(T) = \dim(C_i) = N \times 96$, $\forall i = 1, ..., 11$ e $T = \sum_{i=1}^{11} C_i$. Neste caso N é a quantidade de dias de funcionamento do equipamento de contagem, C_i é a matriz de volume de tráfego para uma determinada classe de veículos i e T é a matriz de volume de tráfego total, ou seja, a quantidade total de volume de tráfego sem distinção por classes de veículos.

Se redimensionarmos a matriz T para um vetor t de dimensão $(N \times 96) \times 1$ respeitando a ordem temporal de T, obtemos uma representação temporal do registro de contagens de tráfego como exemplificado na Fig. 3.1.

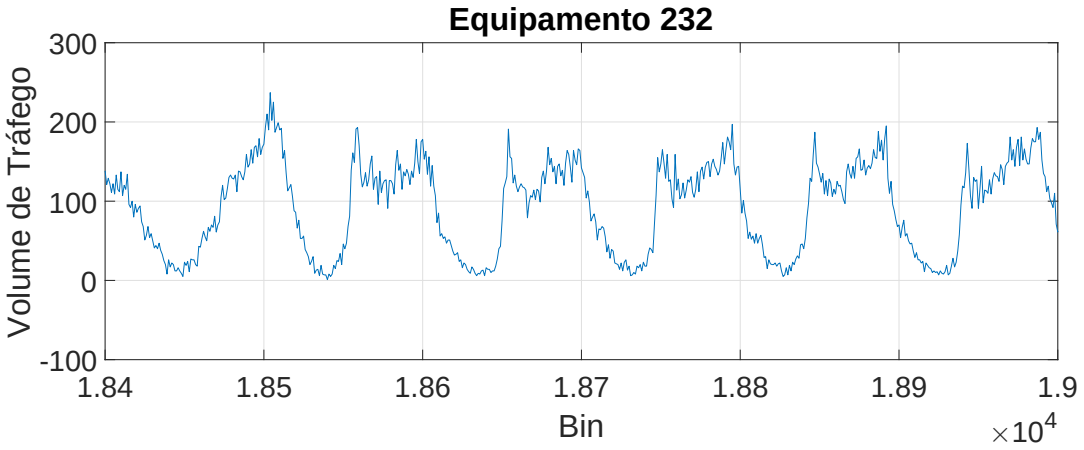


Figura 3.1: Exemplo de série temporal do equipamento 232. Cada Bin representa a posição de um intervalo de tempo de 15 min em um dia corrente. Assim, cada dia possui 96 Bins. O Volume de Tráfego é a quantidade de veículos totais registrados em um Bin, ou seja, em um intervalo de 15 min. Geralmente na madrugada o Volume de Tráfego diminui consideravelmente, gerando este comportamento periódico que podemos ver nesta Figura.

3.1.1 As possíveis anomalias nos dados

O sistema em tempo real utilizado pelo equipamento de contagem de tráfego designado pelo DNIT realiza a coleta de dados ininterruptamente desde sua ativação. Sendo inevitável a aparição de problemas ao longo de anos de coleta de dados, algumas anomalias sistemáticas acabam por gerar dados corrompidos. Alguns exemplos destas anomalias são o registro simultâneo de vários veículos em um mesmo instante de tempo e a ausência de operação aparente durante intervalos de tempo diversos. Iremos nos referir à um bin que possui um valor de volume de tráfego considerado anômalo como um ***outlier***. Na Fig. 3.2 podemos ver possíveis comportamentos anômalos acontecerem nos registros do equipamento 332.

Para que se possa tratar estas anomalias, o algoritmo precisa receber antecipadamente um arquivo .csv que servirá como um marcador para os bins que possuem valores considerados anômalos. Este arquivo deve conter uma matriz O de dimensões $N \times 96$ de tal forma que:

$$O_{i,j} = \begin{cases} 1 & \text{caso o bin } T_{i,j} \text{ seja um outlier} \\ 0 & \text{caso contrário.} \end{cases} \tag{3.2}$$

Deve-se ressaltar que a classificação dos bins em outliers é realizada por outro procedimento fora do escopo desta dissertação.

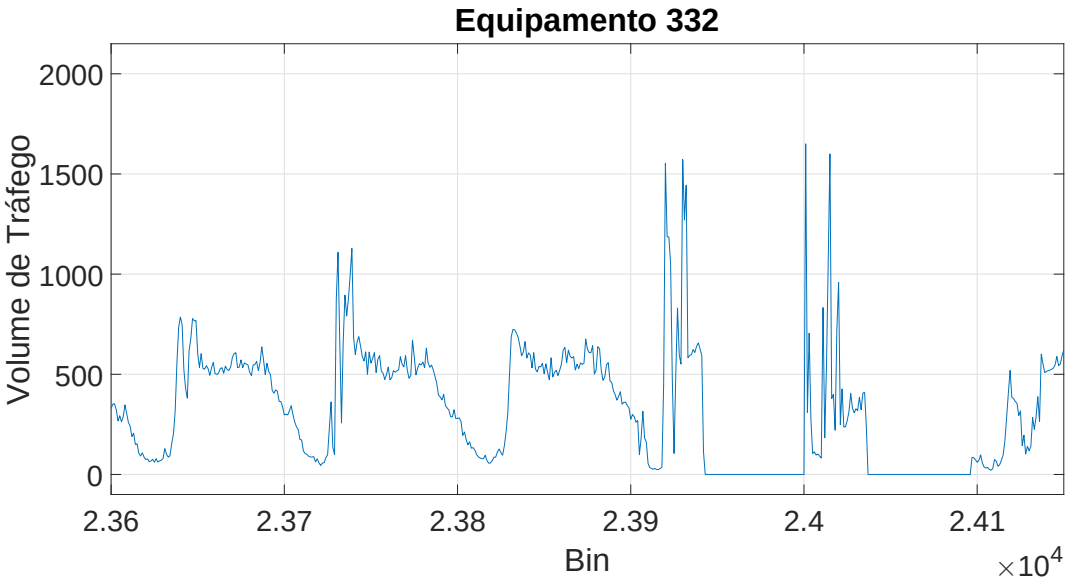


Figura 3.2: Exemplo de possíveis dados anômalos, outliers, no equipamento 332. Entre os bins 23900 e 24000, podemos verificar a aparente ausência de contagem de tráfego e a aparente contagem excessiva de veículos, neste caso, o registro de mais de 1500 veículos em um único intervalo de 15 min.

3.1.2 Propriedades da série temporal

Alguns comportamentos podem ser observados a priori nas séries temporais obtidas dos dados fornecidos pelos equipamentos de contagem. A mais óbvia delas é a sazonalidade diária que pode ser observada na Fig. 3.1. Pode-se também perceber as rápidas variações durante o dia. Outra propriedade observada é a sazonalidade semanal, que pode ser observada na Fig. 3.3.

São com estas propriedades que faremos a escolha das funções de covariância a serem utilizadas na imputação pelos processos Gaussianos, bem como dos valores iniciais para seus hiperparâmetros.

3.2 O Algoritmo para a estimação dos outliers

Nesta Seção serão mostradas as ideias que estão por trás da metodologia aplicada para a resolução do problema de estimação dos outliers na série temporal de um equipamento de contagem permanente de tráfego.

3.2.1 Visão externa do algoritmo

A visão mais externa do algoritmo pode ser obtida na Fig. 3.4, onde mostram-se suas entradas e saídas. Verifica-se então que a entrada é determinada pelos blocos: Equipamento de contagem, Pilha de modelos, Opções e Constantes. No caso da saída: Matriz de contagem volumétrica imputada, Vetor de dias válidos e Variáveis relevantes do algoritmo.

3.2.2 Entradas

Abordaremos agora cada uma das entradas descritas na Fig. 3.4.

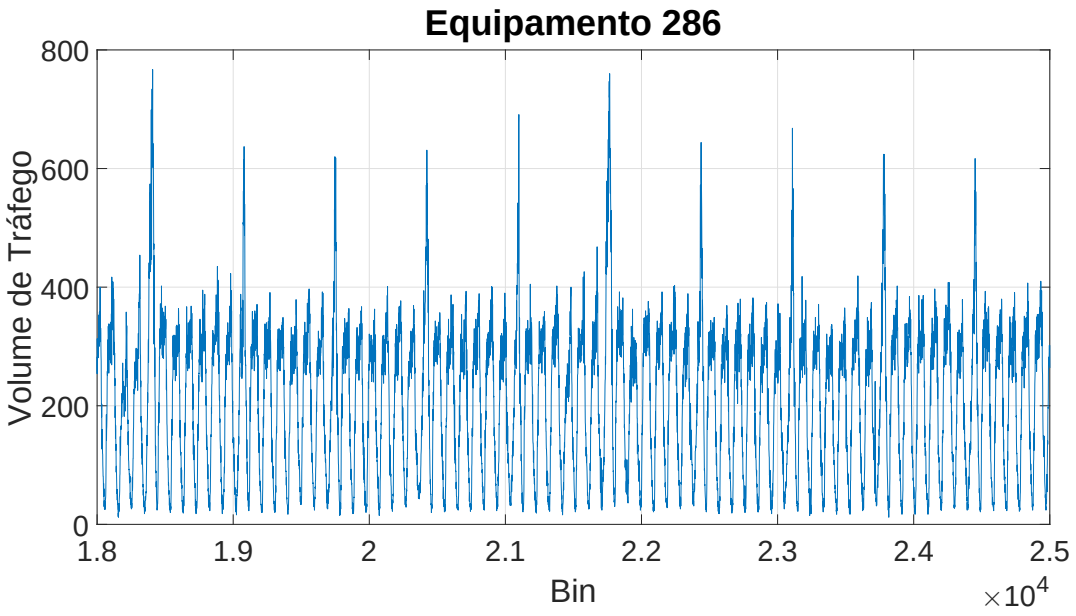


Figura 3.3: Exemplo de sazonalidade semanal.

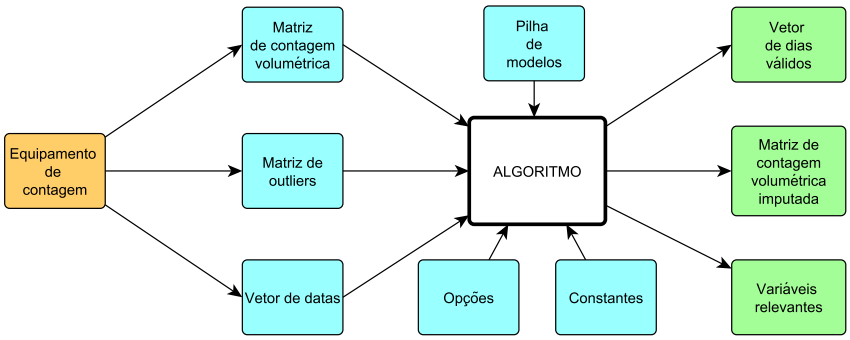


Figura 3.4: Esquema de entradas e saídas do algoritmo.

Equipamento de contagem

O equipamento de contagem nos fornece três arquivos .csv. Como visto anteriormente, dois destes arquivos .csv são relacionados aos registros de volume de tráfego e à marcação de outliers. Além destes, também é fornecido um arquivo .csv com um vetor coluna de datas indicando a data de cada dia de operação nos outros dois arquivos, neste caso a data de cada "linha" destes arquivos. Devemos levar em conta também os sentidos da rodovia na qual o posto de coleta de dados foi implementado. Nem todos os postos realizaram a contagem em dois sentidos da rodovia, porém a maioria realizou. Assim designamos pelas siglas SC e SD, respectivamente, o Sentido Crescente e o Sentido Decrescente da rodovia. Por fim, a estrutura de pastas e arquivos possui nomes autoexplicativos e é da seguinte maneira:

$$./\text{eq\_XXX} \begin{cases} /\text{freq\_eq\_XXX\_sent\_C\_pnct.csv} \\ /\text{freq\_eq\_XXX\_sent\_D\_pnct.csv} \\ /\text{lista\_datas\_eq\_XXX.csv} \\ /\text{outl\_eq\_XXX\_sent\_C.csv} \\ /\text{outl\_eq\_XXX\_sent\_D.csv} \end{cases}$$

onde XXX é a numeração do equipamento.

Constantes do algoritmo

A implementação do algoritmo utiliza uma certa configuração de constantes internas para seu funcionamento. Tendo a maioria destas constantes sido relacionadas a gaps, introduziremos brevemente a noção de *Gap* como sendo um trecho de bins consecutivos onde todos eles são classificados como outliers. A discussão em torno dos gaps será vista mais adiante.

Na Tabela 3.1 temos uma lista de todas as constantes internas utilizadas no algoritmo junto às suas respectivas configurações. Abaixo segue a descrição de cada uma destas constantes:

- **gap_types**: vetor de números naturais que descreve os tipos de gaps considerados no algoritmo através da sua quantidade de bins. Por exemplo, se um gap é classificado como sendo do tipo gap_types(4)= 48, então o tamanho dele verifica a seguinte desigualdade, $24 <$ tamanho_do_gap ≤ 48. Para o tipo de gap de 35040 bins, temos que por ser o último, este tipo de gap é a classificação para todos os gaps com o tamanho estritamente acima de 17280.

- **gap_train_points**: vetor de números naturais contendo a quantidade de dados de treino utilizados na imputação de seu respectivo tipo de gap ao qual foi designado. Por exemplo, para a imputação de um gap classificado como sendo do tipo gap_types(3) serão utilizados gap_train_points(3) = 300 dados de treino. Estes 300 dados de treino serão obtidos através de uma vizinhança do gap para que se possa treinar o modelo vigente no momento e em seguida realizar a imputação do respectivo gap com o modelo otimizado.

- **gap_permitted**: número natural que indica a quantidade de tipos de gaps iniciais que serão efetivamente utilizados na seleção de modelos para o primeiro nível de imputação (imputação dos gaps considerados pequenos). Essa constante possui fins para debug. Por exemplo, caso gap_permitted= 3, a seleção de modelos irá considerar apenas os três primeiros tipos de gap.

- **LONG_GAP_THRESHOLD_**: número natural que representa o limiar entre gaps curtos e gaps longos. Por exemplo, um gap de tamanho menor ou igual a LONG_GAP_THRESHOLD_ é considerado pequeno, caso contrário é considerado um gap longo.

- **long_gap_types**: vetor de números naturais que indicam os tipos de gap para serem utilizados no segundo nível de imputação destinado aos gaps considerados longos.

- **long_gap_train_points**: vetor de números naturais contendo a quantidade de dados de treino utilizados na imputação de seu respectivo tipo de gap longo.

- **T1**: número real positivo que representa o primeiro limiar para a classificação dos trechos da série temporal baseado em sua taxa de outliers. Por exemplo, dado um trecho T de uma série temporal, se taxa_de_outliers $\leq$ T1, então T pode ser classificado ou como 0 ou como 1. Caso contrário, T pode ser classificado como 2 ou 3.

- **T2**: número real positivo que representa o segundo limiar para a classificação dos trechos da série temporal baseado em sua taxa de outliers. Por exemplo, dado um trecho T de uma série temporal, se taxa_de_outliers $\leq$ T2, então T pode ser classificado ou como 0, ou como 1, ou como 2. Caso contrário, T é classificado 3.

Tabela 3.1: Configurações das constantes internas.

Constantes internas	Valores utilizados
gap_types	[4, 12, 24, 48, 96, 288, 672, 2880, 17280, 35040]
gap_train_points	[200, 250, 300, 400, 500, 600, 1000, 1000, 1000, 1000]
gap_permitted	6
LONG_GAP_THRESHOLD_	288
long_gap_types	[672, 2880, 17280, 35040]
long_gap_train_points	[1000, 1000, 1000, 1000]
T1	0.15
T2	0.4

Opções e Pilha de Modelos - Caso de uso

O usuário tem a liberdade de inicializar algumas variáveis que determinam o comportamento do algoritmo, como é o caso da variáveis *database* e *opt*. Ele também pode configurar os modelos na *Pilha de modelos* para serem considerados tanto na seleção de modelos, quanto na imputação dos outliers. Na Fig. 3.5 temos o caso de uso representado por um diagrama. Na Tabela 3.2 é definida brevemente cada uma destas variáveis passíveis de configuração pelo usuário final. Abaixo temos uma descrição mais profunda sobre cada uma dessas variáveis:

- **database**: variável contendo o caminho para o diretório onde se encontra a base de dados.
- **opt.n_steps**: vetor de números naturais contendo a quantidade de cenários gerada na seleção de modelos para um determinado tipo de gap. Por exemplo, caso opt.n_steps(5) = 120, então serão gerados 120 cenários para o tipo de gap gap_type(5) = 96.
- **opt.border**: número natural, limitado entre 0 e 50, indicando a quantidade de bins para a borda do sub-trecho na geração de cenários. A importância deste parâmetro é gerar cenários onde o gap sintético nunca estará na borda de um sub-trecho, ou seja, isto força a garantia de que haverá pontos de treino à direita e à esquerda do gap sintético.
- **Stack_of_models**: Vetor de células contendo os modelos configurados pelo usuário. Para realizar esta configuração, um arquivo .m externo deve ser preenchido de forma a conter todos os modelos considerados pelo usuário para uso do pacote de processos gaussiano GPML[11]. Este arquivo gera o vetor de células Stack_of_models. Um modelo nesta pilha carrega as seguintes informações:
  - ID: Identificação do modelo. Geralmente é a posição no vetor de células Stack_of_models.
  - Mean function: Configuração da função de média escolhida. Geralmente este campo é nulo.
  - Covariance function: Configuração da função de covariância definida pelo usuário.
  - Likelihood function: Função de verossimilhança. Nesta dissertação utilizamos apenas a função Gaussiana.
  - Inference method: Método de inferência escolhido. Nesta dissertação apenas abordamos o método exato.
  - Number of optimization steps: Número natural que indica a quantidade de passos de otimização para a adaptação do modelo aos dados de treino.
  - Hyperparameters: Estrutura contendo todos os hiperparâmetros iniciais definidos pelo usuário.
  - Gap types: vetor de números inteiros representando os tipos de gap aos quais o modelo em questão será candidato para sua respectiva imputação. Assim, para os tipos de gap determinados aqui, o modelo fará parte do processo de seleção para a escolha do melhor modelo para cada tipo de gap ao qual ele foi considerado. Por exemplo, o vetor [4 12] indica que o modelo será candidato ao ranking envolvendo todos os modelos que foram destinados à imputação de tipos de gap 4 e 12.

Tabela 3.2: Caso de Uso

Variável	Tipo	Descrição
database	String	Caminho para a localização da base de dados.
opt.n_steps	$\mathbb{N}_*^8$	Vetor com número de cenários a serem gerados para cada tipo de gap.
opt.border	$\mathbb{N}$	Quantidade de bins para a borda dos sub-trechos na geração de cenário.
Stack_of_models	Vetor de células	Vetor contendo os modelos configurados pelo usuário.

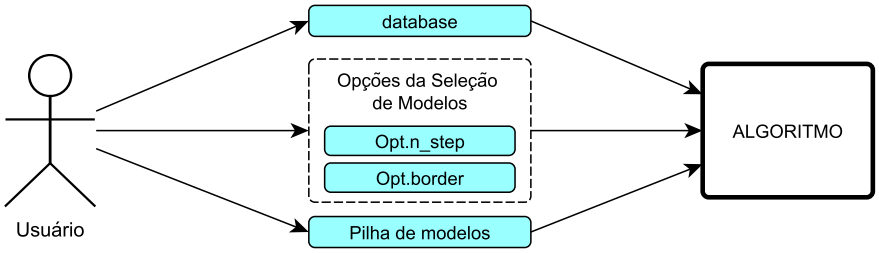


Figura 3.5: Caso de uso.

3.2.3 Saídas - Análise de funcionalidades

A análise de funcionalidades indica tudo aquilo que o algoritmo é capaz de realizar. Assim, na Fig. 3.6 podemos observar todas as saídas geradas pelo algoritmo, além dos resultados que já podíamos conferir no esquema da Fig. 3.4, na pág. 38. Quatro saídas são geradas:

- **Matriz de contagem volumétrica imputada**: O objetivo do algoritmo é estimar valores para os bins considerados outliers. Neste caso, ao final desta tarefa, o algoritmo gera um arquivo .csv contendo os registros de volume de tráfego baseado na estrutura de um arquivo de entrada como "freq_eq_XXX_sent_C_pnct.csv". A nomenclatura padrão para este novo arquivo é "imput_eq_XXX_sent_Y_pnct.csv". Neste caso XXX é a nomeação do equipamento de contagem e Y é o sentido da rodovia considerado, podendo ser C para crescente como D para decrescente.
- **Vetor de dias válidos**: Vetor coluna binário para classificar cada dia da matriz de frequências imputada. Neste caso, 0 indica um dia válido que pode ser utilizado futuramente. Caso contrário emprega-se 1 para os dias inválidos. Dias válidos possuem esta denominação apenas para indicar os dias que não são considerados partes de um gap longo. Este vetor é utilizado em procedimentos futuros fora do escopo desta dissertação.
- **Variáveis relevantes salvas**: Um arquivo .mat é gerado para armazenar todas as variáveis relevantes utilizadas no decorrer do algoritmo. Sua denominação é "PNCT_REGRESSION_RESULTS.mat". Neste arquivo são salvas todas as variáveis necessárias para reproduzir novamente os resultados atingidos, bem como informações gerais como, por exemplo, o tempo gasto pelo algoritmo.
- **Mensagens de erro**: Mensagens de erro são produzidas para comportamentos inesperados do algoritmo.

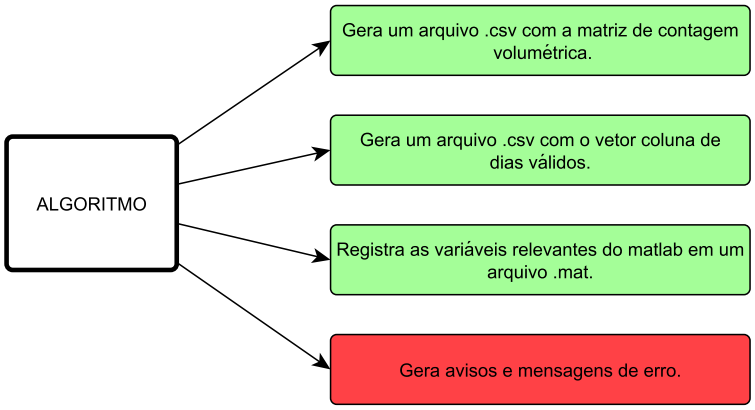


Figura 3.6: Análise de funcionalidades do algoritmo.

3.2.4 A estrutura do algoritmo

Podemos verificar a principal estrutura do algoritmo no pseudo-código descrito no Algoritmo 1. O esquema da Fig. 3.7 auxilia no entendimento do fluxo de funcionamento do Algoritmo 1, mas neste caso não temos a distinção entre os sentidos Crescente ou Decrescente que um equipamento pode nos fornecer.

Com as entradas da Subseção 3.2.2 estabelecidas, primeiro gera-se o que denominaremos "estrutura_do_equipamento" (*eq_str*), Alg. 1-linha 4. Quatro estruturas principais são criadas e adicionadas à *eq_str*: a estrutura *TS* contendo a série temporal , a estrutura *TSWO* contendo a série temporal sem outliers , a estrutura *outl* contendo a marcação de outliers e o vetor de datas *date* onde se encontram as datas de cada dia na matriz T, definida na Seção 3.1. Outros objetos de interesse, mas que não entrarão em evidência nesta dissertação, são as informações sobre os caminhos de diretórios e as variáveis de segurança que ajudam a cuidar das exceções do algoritmo, caso ocorra alguma.

Das seções anteriores constata-se que podemos fazer a seguinte classificação para os dados na matriz T. Um bin ou é um outlier ou é um possível ponto de treino. "Possível", neste caso, porque não sabemos a priori se ele será ou não utilizado pelo método. Uma definição que utilizaremos bastante é a de gap. Como visto anteriormente, um *gap* é um trecho de bins onde todos eles são classificados como outliers.

Os dois níveis de imputação

Como podemos ver mais claramente no esquema da Fig. 3.7, dois níveis de imputação são considerados no Alg. 1. O primeiro nível é destinado à recuperação de gaps curtos e o segundo nível à recuperação de gaps longos. Podemos ver que consideramos vários tipos de gaps em gap_types, porém cada nível só terá acesso à alguns tipos de gap em particular, de acordo com o que foi estabelecido internamente para gaps curtos e gaps longos. Os testes realizados deste algoritmo utilizaram o delimitador LONG_GAP_THRESHOLD_ = 288 bins. Portanto os gaps curtos são: 1h, 3h, 6h, 12h, 1d e 3d. Todos os outros são considerados gaps longos. Ocorre também que a série temporal com os gaps curtos imputados é *reutilizada* na reconstrução de gaps longos como parte da estratégia de imputação. Na Fig. 3.8 observa-se o segundo nível de imputação, onde são destacados o pós-processamento dos resultados obtidos pela aplicação do Núcleo.

Núcleo - as instruções para a imputação

Tendo estabelecido a estrutura *eq_str*, o primeiro nível de imputação é o próximo passo. Denominaremos *Núcleo* como o conjunto interno de instruções para realizar uma imputação. Seu esquema é apresentado na Fig. 3.9. Podemos ver na Fig. 3.7 que o Núcleo aparece nos dois níveis de imputação da mesma

Algorithm 1 Model selection and imputation for several long time series

```
- Initialize the stack of models and constants;
- Configure options for model selection;
for eq = 1 : number of equipments do
    eq_struct(eq) ⟵ get_equipment(eq)
    for direction = 1 : 2 do
        for level = 1 : 2 do
            if level == 2 then
                TSwO, Outl ⟵ update(TSwO, Outl)
            end if
            strip_basis, gaps ⟵ gaps_and_strips(constants, eq_structure)
            best_models ⟵ model_selection(constants,
                                          eq_structure,
                                          strip_basis,
                                          options,
                                          stack_of_models)
            gaps ⟵ add_models(gaps, best_models)
            gaps ⟵ vector_of_structure(gaps)
            eq_struct(eq) ⟵ imputation(TSwO,
                                       constants,
                                       gaps,
                                       stack_of_models)
        end for
        - Store imputed volumetric counting matrix
        - Store valid days vector
        - Save matlab variables
    end for
end for
```

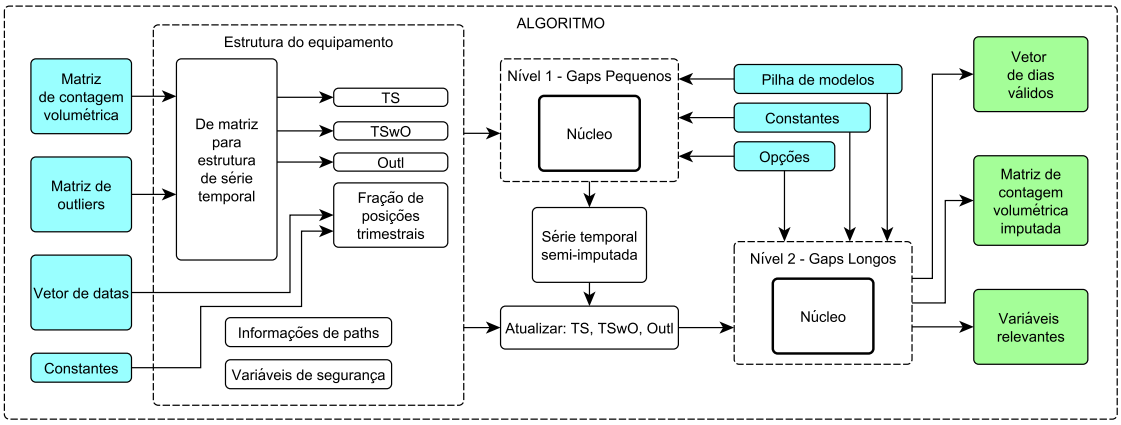


Figura 3.7: Esquema interno do algoritmo.

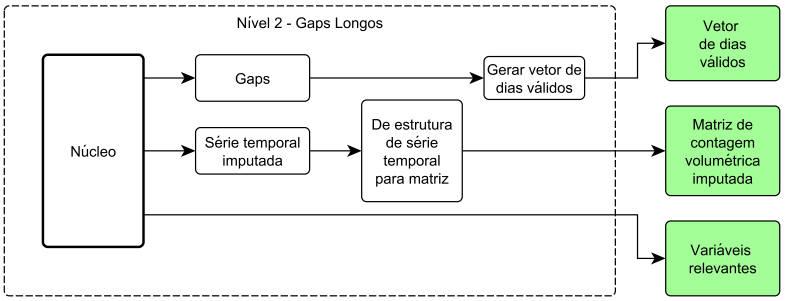


Figura 3.8: Segundo nível de imputação.

maneira. A única mudança significativa entre os dois níveis de imputação ocorre nas entradas do Núcleo. Sua composição principal é dada por três rotinas essenciais para a imputação: a Geração da matriz de validação dos trechos e da estrutura *Gaps*, a Seleção de modelos e a Imputação. Estas três rotinas serão discutidas mais adiante. Como saídas do Núcleo, temos: *A estrutura Gaps*, onde contêm todas as informações relativas aos gaps imputados no nível em questão, *as variáveis relevantes*, onde se encontram os resultados de algumas rotinas e por fim *a série temporal imputada*.

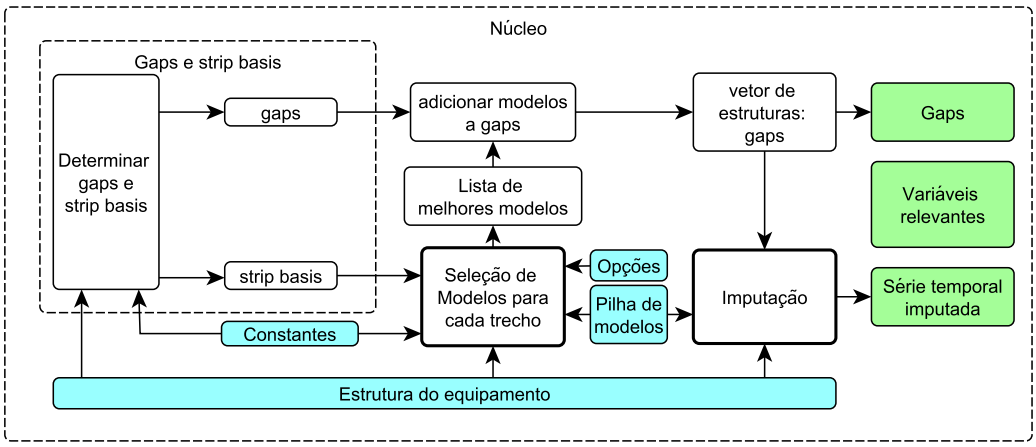


Figura 3.9: Núcleo. O primeiro nível de imputação.

A geração da matriz de validação dos trechos - strip basis

Como vimos no capítulo anterior, a técnica de processos Gaussianos possui uma complexidade numérica de $O(n^3)$ operações e isto é um gargalo computacional que torna o uso de grandes quantidades de dados para

treino quase impraticável, tomando um tempo considerável para os cálculos computacionais. Após alguns testes, foi constatado empiricamente que acima de 1000 bins para treino, ∼10 dias, o tempo computacional começa a aumentar fortemente no dispositivo utilizado para testes. Com esta grande restrição e também com a incerteza sobre o comportamento dos dados ao longo do tempo, a série temporal a ser tratada é dividida em trechos de meio trimestre, cerca de 45 dias, como mostrado na Fig. 3.10 através de barras verticais. Note que o primeiro e último trechos são um pouco maiores. Isto é resultado da aglomeração dos trechos da borda com seus vizinhos devido ao tamanho ser menor que 45 dias. Os trechos são baseados no vetor de datas, onde os trimestres anuais considerados são: 01/Jan, 01/Abr, 01/Jul e 01/Out.

Em cada trecho será determinado um conjunto de modelos a serem utilizados na imputação de seus gaps. Gaps que estiverem simultaneamente em dois trechos terão o seu trecho determinado através da localização de seu centro e no caso deste se encontrar exatamente entre dois trechos, o trecho anterior será escolhido. Entretanto na Fig. 3.10, podemos verificar trechos onde 100% de dados são considerados outliers. Para estes trechos não é possível estabelecer um conjunto de modelos, pois não há dados a serem utilizados pelo processo de seleção de modelos. Desta maneira é necessário validar antecipadamente cada trecho para enviá-los à seleção de modelos, afim de estabelecê-los adequadamente.

A validação de cada trecho é realizada com relação à sua taxa de outliers. A *taxa de outliers* é uma proporção gerada entre as quantidades de outliers e de bins totais do trecho na qual está sendo calculada. Assim são considerados quatro casos baseados em dois delimitadores: $T_1 = 0.15$ e $T_2 = 0.4$, representando, respectivamente, 15% e 40% de outliers:

- Caso 0 | taxa_de_outliers $= 0$
  A proporção de outliers é ótima. Não há gaps a serem imputados, ou seja, não precisamos selecionar modelos para este trecho.

- Caso 1 | taxa_de_outliers $\leq T_1$
  A proporção de outliers é boa e o trecho pode ser utilizado para treino na seleção de modelos.

- Caso 2 | $T_1 <$ taxa_de_outliers $\leq T_2$
  A proporção de outliers não é boa e o trecho precisa ser investigado. Procura-se então um sub-trecho, através de uma busca binária[2] dentro do trecho original, que possua uma proporção de outliers menor que a metade da do trecho original. O tamanho mínimo aceitável do sub-trecho é de 1000 bins. Caso este não seja encontrado, o trecho original é mantido.

- Caso 3 | $T_2 <$ taxa_de_outliers
  A proporção de outliers é extremamente inadequada e o trecho não pode ser utilizado na escolha de modelos a priori. Ainda assim é feita uma busca "trinária"[3] para encontrar um sub-trecho com uma proporção menor ou igual que o terço do original.

Um exemplo desta validação ocorrendo pode ser visto na Fig. 3.11.

A geração da estrutura *Gaps*

As séries temporais apresentam uma gama variada de gaps, desde gaps de 1 bin até gaps de ∼30 mil bins, aproximadamente 1 ano de perda de dados. Assim, cada gap é classificado de acordo com o seu tamanho. Os tipos de gaps considerados são aqueles entre os delimitantes dados por gap_types= $[4; 12; 24; 48; 96; 288; 672; 2880; 17280; 35040]$. Convertendo para a escala temporal, os delimitantes são respectivamente: $1h, 3h, 6h, 12h, 1d, 3d, 1s, 1m, 6m, 1a$. Então dado um gap $\mathbb{G}$ e seu tamanho $L(\mathbb{G})$, segue que:

$$\text{type}(\mathbb{G}) = \min\{i \in \mathbb{N} : L(\mathbb{G}) \leq \text{gap\_types}(i)\} \tag{3.3}$$

[2] A busca binária é uma técnica de busca recursiva baseada no algoritmo de *dividir e conquistar*. No caso de um trecho, este é dividido ao meio e verifica-se a taxa de outliers em cada parte. Caso aceite um dos trechos, a busca termina. Caso contrário, os trechos são divididos novamente em duas partes e o procedimento anterior é refeito. A busca continua até achar algum trecho ou atingir algum critério de parada.

[3] O mesmo que a busca binária, porém no caso de um trecho, este é dividido em três partes, ao invés de duas.

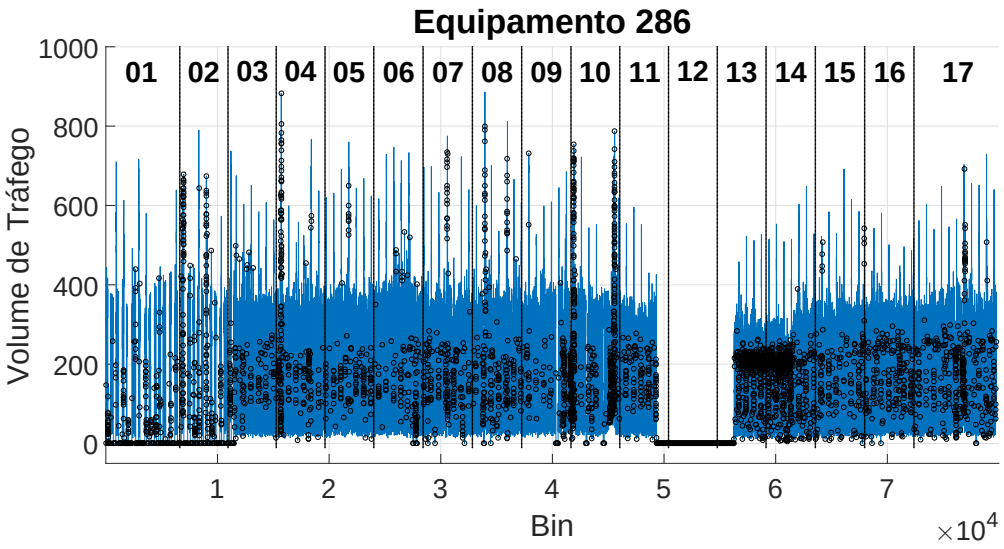


Figura 3.10: A série temporal completa em azul relativa ao equipamento 286 com a marcação dos outliers em círculos pretos. Os índices superiores indicam os trechos de meio-trimestre, 45 dias, delimitados pelas barras verticais em preto.

Trechos	Matriz de validação dos trechos – Equipamento 286 - SC						
01	[1]	[6623]	[0.617962264150943]	[3]	[]	[]	[]
02	[6625]	[4319]	[0.614441101596853]	[3]	[]	[]	[]
03	[10945]	[4319]	[0.11201110853969]	[1]	[]	[]	[]
04	[15265]	[4367]	[0.0512703135729]	[1]	[]	[]	[]
05	[19633]	[4367]	[0.029755092698558]	[1]	[]	[]	[]
06	[24001]	[4415]	[0.0584106859859633]	[1]	[]	[]	[]
07	[28417]	[4415]	[0.045958795562599]	[1]	[]	[]	[]
08	[32833]	[4415]	[0.0552411138781979]	[1]	[]	[]	[]
09	[37249]	[4415]	[0.0762961285940684]	[1]	[]	[]	[]
10	[41665]	[4367]	[0.140993362325475]	[1]	[]	[]	[]
11	[46033]	[4367]	[0.277180132753491]	[2]	[46033]	[2183]	[0.0297619047619048]
12	[50401]	[4367]	[1]	[3]	[]	[]	[]
13	[54769]	[4367]	[0.419089036392767]	[3]	[56953]	[2183]	[0.1364468864468886]
14	[59137]	[4415]	[0.128367670364501]	[1]	[]	[]	[]
15	[63553]	[4415]	[0.0371292732623953]	[1]	[]	[]	[]
16	[67969]	[4415]	[0.0477699796241793]	[1]	[]	[]	[]
17	[72385]	[7391]	[0.0626352813852814]	[1]	[]	[]	[]

Figura 3.11: Matriz de validação dos trechos para o equipamento 286 no sentido SC. A visualização dos trechos pode ser feita na Fig. 3.10. Cada linha representa um trecho. A primeira coluna da matriz de validação representa a posição do bin inicial do trecho. A segunda coluna representa o tamanho do trecho em bins. A terceira coluna armazena os resultados da taxa de outliers de cada trecho. A quarta coluna é a classificação que o trecho recebeu. As três últimas colunas possuem as mesmas designações que as três primeiras, porém para os sub-trechos encontrados. Note que os dois primeiros trechos não possuem sub-trechos com uma taxa de outliers melhor e são desconsiderados. Por outro lado, os trechos 11 e 13, que abordam o início e o fim de um gap longo, conseguem melhorar bastante a taxa de outliers. Já para o trecho 12 não há o que ser feito e este é sumariamente descartado da etapa de seleção de modelos.

Assim, cada gap na série temporal TS possui uma classificação. Além do mais estarão associados aos tipos de gap a quantidade necessária de pontos de treino para a imputação do respectivo gap que será

predefinida pelo seguinte vetor gap_train_points = [200, 250, 300, 400, 500, 600, 1000, 1000, 1000, 1000].

A estrutura Gaps armazena as seguinte informações:

- **Transição**: Caso o gap contenha uma transição de trechos (posição do bin onde as barras verticais em preto estão designadas na Fig. 3.10), esta variável é configurada como 1. Caso contrário, 0.
- **Posição inicial**: Posição na série temporal *TS* do primeiro bin pertencente ao gap em questão.
- **Tamanho do gap**: Quantidade de bins contidos no gap.
- **Pontos de imputação**: Todas as posições de cada bin contido no gap para serem imputadas mais tarde.
- **Pontos de treino**: Posições dos pontos de treino em *TSWO* contidos em uma vizinhança de raio r centrada na mediana do gap.
- **Mediana**: Posição do bin mais próximo ao centro do gap.
- **Raio de treino**: Raio baseado na quantidade de pontos de treino determinada por gap_train_points. Neste caso $r = \frac{1}{2}$gap_train_points(i), para algum i determinado pela classificação do tipo de gap.

A seleção de modelos

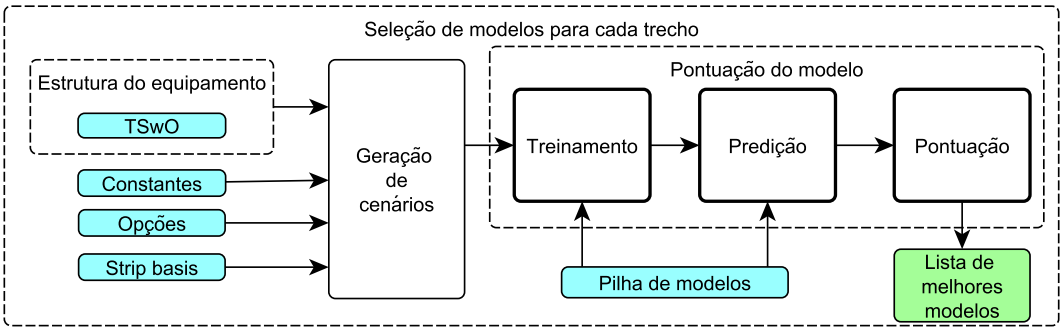


Figura 3.12: Seleção de modelos.

Para reconstruirmos os gaps da série temporal, precisamos saber previamente qual o modelo e quais pontos de treino utilizaremos para aquele determinado tipo de gap. Cada trecho de 45 dias possuirá sua lista de melhores modelos correspondentes aos tipos de gap especificados em gap_types. A quantidade respectiva de pontos de treino para cada tipo de gap também é especificada por um vetor, a saber: gap_train_points = [200, 250, 300, 400, 500, 600, 1000, 1000, 1000, 1000].

Podemos observar no esquema da Fig. 3.12 que a seleção de modelos consiste em duas partes primordiais: a geração de cenários e a pontuação dos modelos. Assim duas variáveis de configuração serão levadas em conta: a pilha de modelos e as opções de configuração. A pilha de modelos é proveniente de um arquivo externo onde são configurados os modelos que serão escolhidos pela metodologia, como explicado na Subseção 3.2.2. As opções de configuração atuam somente na geração de cenários, onde

serão especificadas a quantidade de cenários gerados por tipo de gap através do vetor *opt.n_steps* e o tamanho da borda dos sub-trechos gerados sintéticamente através da variável *opt.border*.

Nesta etapa *não utilizaremos bins considerados outliers*, portanto os bins utilizados para a seleção de modelos são proveniente da intersecção do trecho, ao qual se está selecionando modelos, com $TSWO$. Designaremos o trecho utilizado aqui como trecho_$tswo$.

Além do mais, na etapa de pontuação dos modelos serão levados em consideração para a comparação de modelos apenas dados com bins entre 08:00h e 20:00h por dois motivos: a importância dos dados nesse período para a área de transporte e pelo fato de escolhermos mais adiante o erro relativo como nossa função de erro, pois no período da madrugada obtemos volumes de tráfego muito baixos, o que faz com que perturbações relativamente pequenas obtenham erros muito maiores que os obtidos em perturbações em dados diurnos.

- A geração de cenários

Um cenário corresponde a um sub-trecho de trecho_$tswo$ onde será gerado sinteticamente um gap. O tamanho deste sub-trecho será baseado no tipo de gap a ser gerado, na quantidade respectiva de pontos de treino a ser utilizada em sua imputação e nas bordas, onde estas existirão para que tenhamos uma configuração sempre com pontos de treino ao redor do gap. Assim, sendo i o tipo de gap a ser gerado:

$$L(\text{sub\_trecho}) = \text{border} + \text{gap\_train\_points}(i) + \text{gap\_types}(i) + \text{border} \tag{3.4}$$

Com este tamanho definido, precisamos apenas escolher o ponto de início do sub-trecho em trecho_$tswo$ e esta escolha é feita aleatoriamente através de uma distribuição uniforme aplicada sobre o intervalo $[0, 1]$ e redimensionada para $[0, L(\text{trecho\_}tswo) - L(\text{sub\_trecho})]$, onde 0 é a posição do bin inicial de trecho_$tswo$. Uma vez gerado o sub-trecho, geramos de forma análoga um gap na região *entre* suas bordas, de acordo com a opção border.

Como temos a restrição de utilizarmos apenas dados entre 08:00h e 20:00h, então um processo de sorteio ocorre de forma que desconsideramos gaps que possuam um coeficiente de rejeição[4] maior que 30% de sua quantidade de bins, ou seja, gaps que possuam a maioria de seus bins fora do intervalo de tempo estabelecido. Caso o sorteio não seja efetivo em 100 passos, aumentamos a tolerância em 10%. Esta metodologia ocorre desta forma, aumentando a tolerância até um máximo de 70% de bins rejeitados. Desta maneira, é possível sempre sortear um gap em condições mínimas para a comparação de modelos.

Desta maneira constrói-se uma *pilha de cenários* onde a quantidade de cenários gerados por tipo de gap é respectivamente dada pela variável *opt.n_steps*, como por exemplo *opt.n_steps* $= [80, 80, 56, 56, 28, 28, 28, 28]$.

Logo, um cenário apresentará o seguinte formato:

$$\text{cenário } = [P_1 \quad L_1 \quad P_2 \quad L_2] \tag{3.5}$$

onde P_1 é a posição inicial do sub-trecho no trecho_$tswo$, L_1 é o tamanho do sub-trecho, P_2 é a posição inicial do gap sintético no sub-trecho e L_2 é o tamanho do gap considerado. A Fig. 3.13 apresenta um exemplo de uma pilha de cenários gerada para *opt.n_steps* $= [1, 1, 1, 1, 1, 1, 1, 1]$ e *opt.border* $= 20$.

- A pontuação dos modelos

Um cenário implica em um sub-trecho de TSWO com dois tipos de bin: aquele pertencente ao gap sintético e aquele que não. Chamaremos de pontos de teste aqueles que pertencem ao gap sintético, caso contrário denominaremos pontos de treino. Assim, um modelo aplicado a um cenário será otimizado com os pontos de treino aplicando o método da máxima verossimilhança tipo-II através da minimização por gradientes conjugados [8, 11]. Com o modelo otimizado, imputa-se nos pontos de teste utilizando o pacote GPML [11] para os Processos Gaussianos e com o resultado avaliamos o quanto o modelo errou com relação às observações reais em TSWO.

Cada tipo de gap terá um conjunto de modelos na pilha de modelos para serem avaliados. Assim cada modelo será aplicado aos seus correspondentes cenários segundo o tipo de gap ao qual foi especificado,

[4] O *coeficiente de rejeição* indica a porcentagem de bins do gap sintético que se encontram fora do intervalo de tempo diário de 08:00h as 20:00h.

como indicado na Pilha de Modelos na Subseção 3.2.2. A função de erro utilizada calcula o erro relativo entre o valor estimado e o valor real:

$$\text{Erro} = \left| \frac{y_{imputado} - y_{real}}{y_{real}} \right| \tag{3.6}$$

Bins pertencentes ao gap e que *não* se encontrem entre 8:00h e 20:00h serão descartados dos pontos de teste. Assim, cada tipo de gap terá um ranking de modelos, o que nos permitirá produzir, para o trecho da série temporal em questão, um conjunto de modelos para realizar a imputação, um para cada tipo de gap. Vale ressaltar que um dos gargalos computacionais do código ocorre neste momento. A solução empregada para isto é paralelizar [9] a pontuação dos modelos em relação as suas respectivas quantidade de cenários.

```
2367        204        158          4
3370        262         31         12
1089        324         61         24
3005        448         89         48
1216        596        270         96
1901        888        394        288
```

Figura 3.13: Pilha de cenários gerada com variáveis de configuração $opt.n\_steps = [1, 1, 1, 1, 1, 1, 1]$ e $opt.border = 20$.

A imputação dos dados

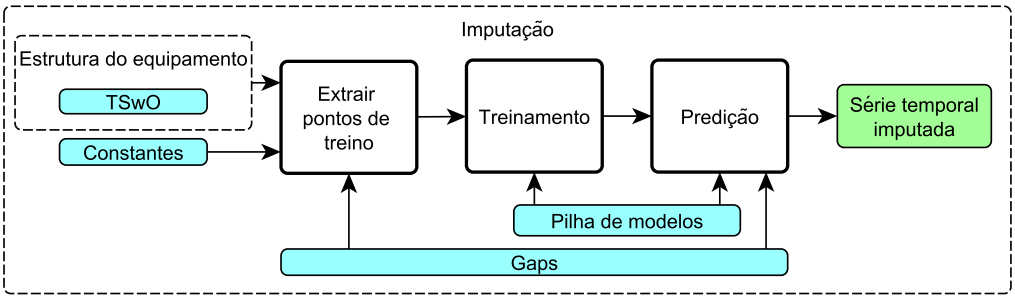


Figura 3.14: Imputação.

Com o conjunto de modelos de cada trecho da série temporal especificados, cada gap detectado na série temporal será imputado independentemente. Para isto todos os gaps terão que ser atualizados com seu respectivo modelo associado baseando-se no trecho ao qual ele pertence. Na Fig. 3.14 são mostradas três

fases principais para a imputação de um gap: a extração dos pontos de treino, o treinamento do modelo em questão e a predição dos bins pertencentes ao gap corrente. Os modelos serão otimizados na fase de treinamento utilizando-se uma vizinhança do gap contendo seus respectivos pontos de treino obtidos em sua fase de extração. A imputação ocorrerá em seguida através da aplicação destes modelos otimizados. Este é mais um gargalo do algoritmo que é resolvido com a paralelização do processo de imputação dos gaps. Na predição de gaps considerados longos, ao invés de imputarmos os outliers utilizando a média da distribuição de probabilidade preditiva como visto anteriormente, sorteamos uma *amostra* desta mesma distribuição e utilizamos seus valores na imputação de seus respectivos outliers. Uma vez imputados os gaps, a nova série temporal é armazenada.

3.2.5 Código fonte

O código fonte escrito em MATLAB possui como arquivo principal ***PNCT_regression.m*** e está disponibilizado em:

- `https://www.dropbox.com/sh/gof9yhx83zrn8he/AADn4rEIn56M9EYOjkTUv2NZa?dl=0`

.

3.3 Aplicação da metodologia

Trataremos nesta Seção de pontos pertinentes à aplicação da metodologia proposta.

3.3.1 Equipamentos e sistemas computacionais

Dois equipamentos foram utilizados: um para o desenvolvimento e fase de testes, e outro para a aplicação.

Especificações de Hardware

- Hardware de desenvolvimento e fase de testes:
  - Fabricante: Micro-Star International Co. (MSI)
  - Modelo: GT60-2OD
  - Processador: 1x Intel(R) Core i7-4700MQ CPU @ 2.40GHz
  - Núcleos físicos: 4
  - Nº de threads: 8
  - Memória cache: 6MB SmartCache
  - Mémoria: 16.0GB DDR3
- Hardware para aplicação
  - Processador: 4x Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
  - Núcleos físicos: 8
  - Nº de threads: 16
  - Memória cache: 20MB SmartCache
  - Mémoria: 66.0GB

Especificação do Sistema Operacional

- S.O. utilizado nas fases de desenvolvimento e de testes:
  - Sistema: Windows
  - Versão: 8.1 x86_64
- S.O. para a fase de aplicação
  - Sistema: Linux
  - Versão: 2.6.32-642.el6.x86_64
  - Distribuição: CentOS 6.8

Ambiente de Desenvolvimento

- Software de desenvolvimento e fase de testes:
  - MATLAB R2016a
- Software para aplicação
  - MATLAB R2015a

Pacotes Externos

Para a aplicação da técnica de processos Gaussianos, foi utilizado o pacote GPML 4.0 que pode ser encontrado em [31]. A escolha por este pacote foi apoiada na flexibilização que ele possui para a manipulação de todos os integrantes necessários para efetuar uma regressão via processos gaussianos de uma função unidimensional. Em especial a facilidade com que ele nos permite manipular a construção de funções de covariância torna este pacote em uma ferramenta poderosa para a modelagem de dados.

3.3.2 Configuração das opções de entrada

Na Tabela 3.3 se encontram as configurações para a variável de opções utilizadas na seleção de modelos. Esta configurações foram estabelecidas para a geração dos resultados mais adiante.

Tabela 3.3: Variáveis e constantes com suas respectivas configurações.

Variáveis	Valores utilizados
opt.n_steps	$[160, 160, 112, 112, 56, 56, 56, 56]$
opt.border	20

3.3.3 Modelos considerados

Nesta subseção encontra-se a configuração da pilha de modelos utilizada. Um modelo é totalmente descrito pelos seguintes itens:

- **Função de covariância**: Aqui podemos configurar a nossa função de covariância que determinará os hiperparâmetros a serem otimizados durante o algoritmo. Os modelos considerados combinam as seguintes funções de covariância: isotropic SE, isotropic RQ e Matern-1/2.
- **Função de verossimilhança**: Para o uso do pacote GPML, consideramos que os dados observados são i.i.d. Assim, a função de verossimilhança torna-se fatorável:

$$p(\mathbf{y}|\mathbf{f}, \sigma_n) = \prod_{i=1}^{n} p_i(y_i|f_i, \sigma_n), \text{ onde } f_i = f(x_i) \tag{3.7}$$

Assim este pacote nos permite escolher uma dentre várias funções de verossimilhança $p_i(y_i|f(\mathbf{x}_i), \sigma_n)$ voltadas para o problema de regressão. Na aplicação aqui realizada, todos os modelos considerados utilizam a distribuição Gaussiana.

- **Método de inferência**: Os métodos de inferência nos permitem calcular, exata ou aproximadamente, as distribuições a posteriori, o logaritmo negativo da verossimilhança marginal e suas derivadas parciais em relação aos hiperparâmetros. Vários métodos de inferência são proposto, porém o aplicado aqui é a inferência exata.

- **Passos de otimização**: Para realizarmos a adaptação dos hiperparâmetros através da maximização da sua verossimilhança, é utilizado um otimizador baseado no método dos gradientes conjugados que necessita da quantidade de passos de otimização como configuração. Cada modelo especifica a sua quantidade de passos de otimização.

- **Tipos de gap**: Cada modelo terá seu conjunto de tipos de gap aos quais ele concorrerá na seleção de modelos. Por exemplo, os modelos 1 e 2 foram considerados para a reconstrução de gaps acima de 48 bins, caso contrário aplica-se os modelos 3, 4 e 5.

Para esta aplicação foram considerados cinco modelos baseados nas propriedade do tipo de série temporal aqui estudada. Nas Tabelas 3.4 a 3.8 apresentamos as configurações para cada um dos modelos propostos.

Tabela 3.4: **Modelo 1**.

Função de Covariância	**Hiperparâmetros**					
Per(Mater-1, 1d) + Per(Mater-1, 1w)	σ_1	l_1	p_1	σ_2	l_2	p_2
	40	96/4	96	40	96	$7*96$
F. de Verossimilhança	**Hiperparâmetro**					
Gaussiana	σ_n					10
Método de Inferência	Exata					
Passos de Otimização	600					
Tipos de Gap	48, 96, 288, 672, 2880					

Tabela 3.5: **Modelo 2**.

Função de Covariância	**Hiperparâmetros**					
Per(Mater-1, 1d)$_1$ + Per(SE, 1w)$_2$	σ_1	l_1	p_1	σ_2	l_2	p_2
	40	96/4	96	40	96/4	$7*96$
F. de Verossimilhança	**Hiperparâmetro**					
Gaussiana	σ_n				10	
Método de Inferência	Exata					
Passos de Otimização	600					
Tipos de Gap	48, 96, 288, 672, 2880					

Tabela 3.6: **Modelo 3**.

Função de Covariância	**Hiperparâmetros**								
Per(Mater-1, 1d) + Per(Mater-1, 1w) + RQ	σ_1	l_1	p_1	σ_2	l_2	p_2	l_3	σ_3	α
	40	96/4	96	40	96	7*96	5	30	3
F. de Verossimilhança	**Hiperparâmetro**								
Gaussiana	σ_n					10			
Método de Inferência	Exata								
Passos de Otimização	600								
Tipos de Gap	4, 12, 24								

Tabela 3.7: **Modelo 4**.

Função de Covariância	**Hiperparâmetros**								
$\text{Per(SE, 1d)}_1 \times \text{RQ}_2 + \text{Per(SE, 1w)}_3 \times \text{RQ}_4 + \text{RQ}_5$	σ_1	l_1	p_1	σ_2	l_2	α_2	σ_3	l_3	p_3
	5	96/4	96	5	5	3	5	96/4	$2*96$
	σ_4	l_4	α_4	σ_5	l_5	α_5	-		
	5	5	3	30	5	3			
F. de Verossimilhança	**Hiperparâmetro**								
Gaussiana	σ_n					10			
Método de Inferência	Exata								
Passos de Otimização	600								
Tipos de Gap	4, 12, 24								

Tabela 3.8: **Modelo 5**.

Função de Covariância	**Hiperparâmetros**								
$\text{Per(SE, 1d)}_1 \times \text{RQ}_2 + \text{Per(SE, 1w)}_3 \times \text{RQ}_4 + \text{SE}_5$	σ_1	l_1	p_1	σ_2	l_2	α_2	σ_3	l_3	p_3
	5	96/4	96	5	5	3	5	96/4	$2*96$
	σ_4	l_4	α_4	σ_5	l_5	-			
	5	5	3	30	10				
F. de Verossimilhança	**Hiperparâmetro**								
Gaussiana	σ_n					15			
Método de Inferência	Exata								
Passos de Otimização	600								
Tipos de Gap	4, 12, **24**								

3.4 Resultados da aplicação

Um dentre todos os equipamentos considerados para reconstrução é o equipamento 232. Apresentamos aqui os resultados para o sentido decrescente SD deste equipamento. O tempo de processamento necessário para realizar a imputação das séries temporais nos dois sentidos, SC e SD, do equipamento 232 foi de 24h:27m:50s utilizando 28 processadores em paralelo para os gargalos computacionais descritos na Subseção 3.2.4.

3.4.1 Equipamento 232 - SD

Série temporal original

Na Fig. 3.15 encontra-se o gráfico da série temporal original gerada utilizando os registros do equipamento 232 SD. Também são exibidas as divisões por trechos de 45 dias e suas marcações de outliers.

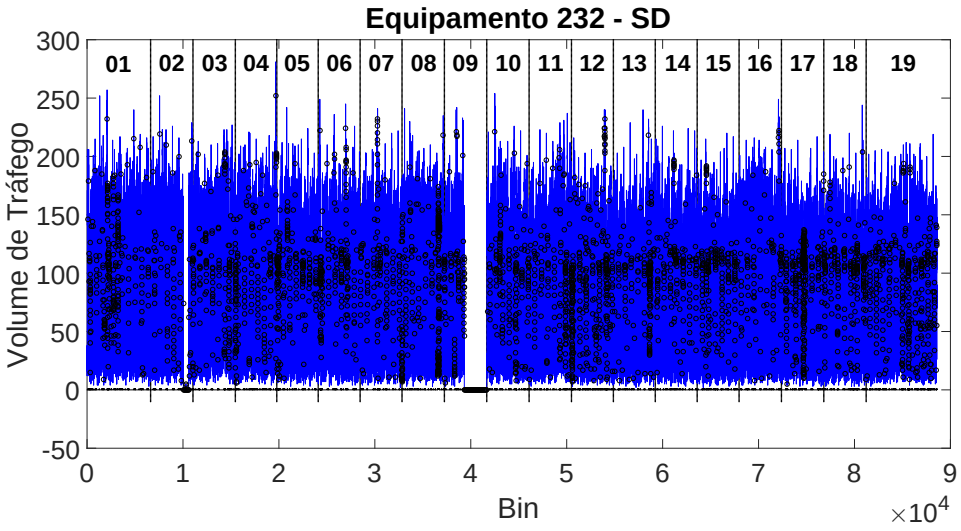


Figura 3.15: Em azul encontra-se o gráfico da série temporal original do equipamento 232 no sentido SD. Os outliers estão representados pelos círculos de cor preta. As barras verticais em preto delimitam os trechos de 45 dias.

Matriz de validação dos trechos

Observe na Fig. 3.16 que os trechos 02 e 09 apresentam suas taxas de outliers, estabelecidas na terceira coluna da matriz de validação de trechos, bem maiores que os demais trechos. Isto se deve ao fato de termos nestes trechos a aparição de gaps longos, como podemos verificar na Fig. 3.15.

Trechos	Matriz de validação dos trechos – Equipamento 232 - SD						
01	[1]	[6623]	[0.04030188679245 28]	[1]	[]	[]	[]
02	[6625]	[4415]	[0.144668326918723]	[1]	[]	[]	[]
03	[11041]	[4415]	[0.0357708852162101]	[1]	[]	[]	[]
04	[15457]	[4319]	[0.0416570238370748]	[1]	[]	[]	[]
05	[19777]	[4319]	[0.04420273084934 04]	[1]	[]	[]	[]
06	[24097]	[4367]	[0.0407415884641794]	[1]	[]	[]	[]
07	[28465]	[4367]	[0.0331883726253147]	[1]	[]	[]	[]
08	[32833]	[4415]	[0.0624858501245189]	[1]	[]	[]	[]
09	[37249]	[4415]	[0.549467964681911]	[3]	[37249]	[2207]	[0.0991847826086956]
10	[41665]	[4415]	[0.0425628254471361]	[1]	[]	[]	[]
11	[46081]	[4415]	[0.0359972832239076]	[1]	[]	[]	[]
12	[50497]	[4367]	[0.0748455024032959]	[1]	[]	[]	[]
13	[54865]	[4367]	[0.0375371938658732]	[1]	[]	[]	[]
14	[59233]	[4367]	[0.0347905699244678]	[1]	[]	[]	[]
15	[63601]	[4367]	[0.0418860151064317]	[1]	[]	[]	[]
16	[67969]	[4415]	[0.0235453928005434]	[1]	[]	[]	[]
17	[72385]	[4415]	[0.066334616255377]	[1]	[]	[]	[]
18	[76801]	[4415]	[0.0513923477473398]	[1]	[]	[]	[]
19	[81217]	[7391]	[0.0472132034632035]	[1]	[]	[]	[]

Figura 3.16: Matriz de validação dos trechos para o equipamento 232 no sentido SD.

Histograma de gaps encontrados

A Fig. 3.17 representa o histograma de gaps existentes na série temporal do equipamento 232 SD. Assim é possível termos uma noção de quais tipos de gaps são mais recorrentes nesta série temporal.

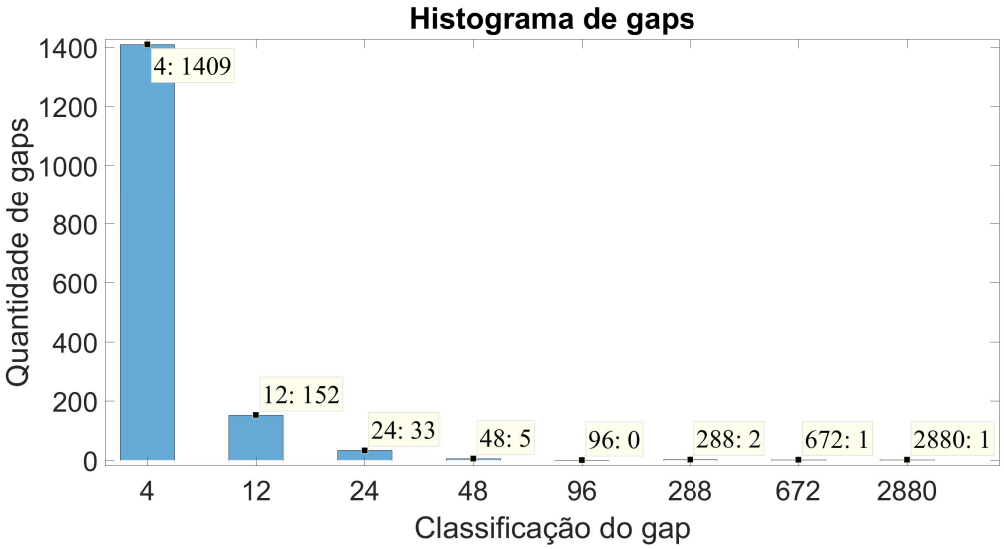


Figura 3.17: Histograma apresentando a quantidade de gaps por tipo de gap presentes na série temporal do equipamento 232.

Resultados da seleção de modelos

Os resultados da seleção de modelos para os dois níveis de imputação podem ser vistos nas tabelas 3.9 a 3.28. Como visto anteriormente, foram considerados 5 modelos para esta aplicação aos quais aqui denominamos m1, ..., m5. A primeira linha "gaps"das Tabelas indica os respectivos tamanhos dos gaps considerados. Em seguida temos a linha "cenários"com as quantidades de cenários geradas por tipo de gap. Da terceira à sétima linha temos os resultados de cada um dos modelos aplicados aos seus respectivos tipos de gap. Tome como exemplo o modelo m2. Os gaps para os quais ele foi considerado são os de 12h, 1d, 3d, 1sem e 1mês. Todos os outros gaps ficam de fora da etapa de seleção para o modelo m2, por este motivo que há os traços em 1h, 3h e 6h. Cada valor respectivo aos modelos da tabela é resultante da média da pontuação gerada ao se aplicar o modelo em questão ao respectivo gap na quantidade de cenários determinada, como explicado na Subseção 3.2.4. Como exemplo, observe a Tabela 3.9 resultante da seleção de modelos para o trecho 01 da Fig. 3.15. O modelo m2 imputa 112 vezes gaps de 12h gerados sinteticamente. Isto resulta em um vetor de 112 entradas contendo o erro relativo de cada imputação realizada. O valor 0.1194 é a média desse vetor de erros relativos. Finalmente na última linha "seleção"estão contidos os melhores modelos, ou seja, aqueles que apresentaram as menores médias de erro relativo referente a cada tipo de gap. No caso da Tabela 3.9, vemos que gaps de tipo 1h no Trecho 01 serão imputados utilizando o modelo m3, já gaps de tipo 12h neste mesmo Trecho 01 serão imputados utilizando o modelo m2. Assim, ao final destes resultados temos grupos de modelos respectivamente determinados para a imputação de gaps em cada Trecho da série temporal no primeiro nível de inferência. Para o segundo nível de inferência, que concerne a imputação de gaps longos, tratamos a série temporal inteira como um único Trecho resultando então em apenas um grupo de modelos, como podemos verificar pela Tabela 3.28.

Tabela 3.9

Trecho 1						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	0.1205	**0.1287**	**0.1335**
m2	-	-	-	**0.1194**	0.1602	0.1488
m3	**0.0882**	**0.1045**	**0.1345**	-	-	-
m4	0.0905	0.1141	0.1544	-	-	-
m5	0.0905	0.1139	0.1616	-	-	-
seleção	**m3**	**m3**	**m3**	**m2**	**m1**	**m1**

Tabela 3.10

Trecho 2						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1174**	**0.1186**	**0.1361**
m2	-	-	-	0.1226	0.1213	0.1479
m3	**0.0859**	**0.1086**	**0.1417**	-	-	-
m4	0.0887	0.1124	0.1665	-	-	-
m5	0.0888	0.1121	0.1703	-	-	-
seleção	**m3**	**m3**	**m3**	**m1**	**m1**	**m1**

Tabela 3.11

Trecho 3						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	0.1091	**0.1081**	**0.1173**
m2	-	-	-	**0.1075**	0.1098	0.1217
m3	0.0907	**0.0977**	**0.1311**	-	-	-
m4	0.0894	0.1076	0.1535	-	-	-
m5	**0.0893**	0.1085	0.1646	-	-	-
seleção	m5	m3	m3	m2	m1	m1

Tabela 3.12

Trecho 4						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1126**	0.1313	**0.1237**
m2	-	-	-	0.1140	**0.1230**	0.1290
m3	0.0977	**0.1100**	**0.1257**	-	-	-
m4	**0.0964**	0.1160	0.1552	-	-	-
m5	0.0966	0.1172	0.1620	-	-	-
seleção	**m4**	**m3**	**m3**	**m1**	**m2**	**m1**

Tabela 3.13

Trecho 5						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1217**	0.1212	**0.1330**
m2	-	-	-	0.1229	**0.1188**	0.1497
m3	**0.0978**	**0.1077**	**0.1326**	-	-	-
m4	0.0988	0.1189	0.1638	-	-	-
m5	0.0990	0.1217	0.1718	-	-	-
seleção	m3	m3	m3	m1	m2	m1

Tabela 3.14

Trecho 6						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1171**	**0.1254**	**0.1346**
m2	-	-	-	0.1237	0.1299	0.1419
m3	0.0920	**0.1138**	**0.1392**	-	-	-
m4	**0.0914**	0.1159	0.1645	-	-	-
m5	0.0923	0.1178	0.1759	-	-	-
seleção	**m4**	**m3**	**m3**	**m1**	**m1**	**m1**

Tabela 3.15

Trecho 7						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1132**	0.1301	0.1383
m2	-	-	-	0.1135	**0.1247**	**0.1210**
m3	0.0899	**0.1082**	**0.1447**	-	-	-
m4	**0.0891**	0.1189	0.1599	-	-	-
m5	0.0892	0.1204	0.1646	-	-	-
seleção	**m4**	**m3**	**m3**	**m1**	**m2**	**m2**

Tabela 3.16

Trecho 8						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1135**	**0.1245**	**0.1300**
m2	-	-	-	0.1176	0.1291	0.1460
m3	0.0933	**0.1153**	**0.1353**	-	-	-
m4	**0.0904**	0.1183	0.1620	-	-	-
m5	0.0908	0.1208	0.1654	-	-	-
seleção	**m4**	**m3**	**m3**	**m1**	**m1**	**m1**

Tabela 3.17

Trecho 9						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1124**	**0.1304**	**0.1583**
m2	-	-	-	0.1202	0.1357	0.3678
m3	**0.0828**	**0.1000**	**0.1224**	-	-	-
m4	0.0849	0.1007	0.1437	-	-	-
m5	0.0850	0.1012	0.1511	-	-	-
seleção	**m3**	**m3**	**m3**	**m1**	**m1**	**m1**

Tabela 3.18

Trecho 10						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1197**	**0.1378**	**0.1663**
m2	-	-	-	0.1216	0.1519	0.1736
m3	0.0942	**0.1047**	**0.1354**	-	-	-
m4	**0.0926**	0.1053	0.1523	-	-	-
m5	0.0928	0.1074	0.1573	-	-	-
seleção	**m4**	**m3**	**m3**	**m1**	**m1**	**m1**

Tabela 3.19

Trecho 11						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	0.1137	**0.1128**	**0.1244**
m2	-	-	-	**0.1136**	0.1168	0.1704
m3	**0.0885**	**0.1049**	**0.1204**	-	-	-
m4	0.0895	0.1094	0.1517	-	-	-
m5	0.0902	0.1094	0.1565	-	-	-
seleção	**m3**	**m3**	**m3**	**m2**	**m1**	**m1**

Tabela 3.20

Trecho 12						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1206**	**0.1259**	**0.1307**
m2	-	-	-	0.1245	0.1380	0.1440
m3	**0.0880**	**0.1131**	**0.1310**	-	-	-
m4	0.0888	0.1191	0.1721	-	-	-
m5	0.0897	0.1206	0.1859	-	-	-
seleção	**m3**	**m3**	**m3**	**m1**	**m1**	**m1**

Tabela 3.21

Trecho 13						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1179**	**0.1163**	**0.1323**
m2	-	-	-	0.1276	0.1194	0.1350
m3	**0.0847**	**0.1056**	**0.1298**	-	-	-
m4	0.0850	0.1154	0.1602	-	-	-
m5	0.0858	0.1182	0.1655	-	-	-
seleção	**m3**	**m3**	**m3**	**m1**	**m1**	**m1**

Tabela 3.22

Trecho 14						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1156**	**0.1243**	0.1392
m2	-	-	-	0.1183	0.1371	**0.1385**
m3	0.0953	**0.1045**	**0.1311**	-	-	-
m4	**0.0949**	0.1131	0.1568	-	-	-
m5	0.0952	0.1144	0.1702	-	-	-
seleção	**m4**	**m3**	**m3**	**m1**	**m1**	**m2**

Tabela 3.23

Trecho 15						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1154**	**0.1213**	0.1266
m2	-	-	-	0.1183	0.1249	**0.1221**
m3	0.0954	**0.1131**	**0.1368**	-	-	-
m4	**0.0924**	0.1224	0.1719	-	-	-
m5	0.0927	0.1241	0.1697	-	-	-
seleção	**m4**	**m3**	**m3**	**m1**	**m1**	**m2**

Tabela 3.24

Trecho 16						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1206**	0.1255	**0.1391**
m2	-	-	-	0.1210	**0.1228**	0.1447
m3	**0.0899**	**0.1102**	**0.1293**	-	-	-
m4	0.0914	0.1174	0.1555	-	-	-
m5	0.0915	0.1193	0.1588	-	-	-
seleção	**m3**	**m3**	**m3**	**m1**	**m2**	**m1**

Tabela 3.25

Trecho 17						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1147**	**0.1293**	**0.1323**
m2	-	-	-	0.1185	0.1347	0.1688
m3	0.0938	**0.1098**	**0.1375**	-	-	-
m4	**0.0931**	0.1162	0.1504	-	-	-
m5	0.0932	0.1179	0.1550	-	-	-
seleção	**m4**	**m3**	**m3**	**m1**	**m1**	**m1**

Tabela 3.26

Trecho 18						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1139**	0.1268	**0.1300**
m2	-	-	-	0.1223	**0.1244**	0.1684
m3	0.0986	**0.1068**	**0.1254**	-	-	-
m4	**0.0964**	0.1129	0.1472	-	-	-
m5	0.0969	0.1144	0.1551	-	-	-
seleção	**m4**	**m3**	**m3**	**m1**	**m2**	**m1**

Tabela 3.27

Trecho 19						
gaps	1h	3h	6h	12h	1d	3d
cenários	160	160	112	112	56	56
m1	-	-	-	**0.1155**	**0.1210**	**0.1346**
m2	-	-	-	0.1218	0.1278	0.1588
m3	0.0935	**0.1087**	**0.1240**	-	-	-
m4	**0.0910**	0.1184	0.1500	-	-	-
m5	0.0914	0.1182	0.1535	-	-	-
seleção	**m4**	**m3**	**m3**	**m1**	**m1**	**m1**

Tabela 3.28

Gaps Longos		
gaps	1sem	1mês
cenários	56	56
m1	**0.1238**	**0.1308**
m2	0.1273	0.1377
m3	-	-
m4	-	-
m5	-	-
seleção	**m1**	**m1**

Resultados da reconstrução

Os resultados da reconstrução são obtidos pelo passo de imputação, uma vez que os gaps já possuam seus modelos selecionados. Na Fig. 3.18, temos um exemplo de um gap de 45 min reconstruído no Trecho 06. Este gap é da classe de gaps de 1h e através da Tabela 3.14 podemos observar que o modelo selecionado para imputar tipos de gaps de 1h é o modelo 4.

Na Fig. 3.19 podemos observar a reconstrução de um gap de 1 dia e meio pertencente ao Trecho 08. Este gap é da classe de gaps de 3 dias. Verificando a Tabela 3.16, podemos ver que o modelo 1 é o modelo selecionado para efetuar esta imputação. Podemos verificar a recuperação do comportamento diário através de Processos Gaussianos baseado-se em uma vizinhança deste gap contendo 600 bins utilizados como pontos de treino.

Já na Fig. 3.20 exibimos um exemplo de reconstrução de gap longo. Neste caso a Tabela 3.28 nos indica que todos os gaps considerados longos serão imputados com o modelo 1. Os dados imputados são indicados pela curva em vermelho, que difere da forma como gaps curtos são imputados, pois esta curva é uma amostra aleatória gerada através de um Processo Gaussiano, ao invés de apenas utilizarmos a média deste Processo, como é realizado para gaps curtos. Note a captura do comportamento diário e semanal.

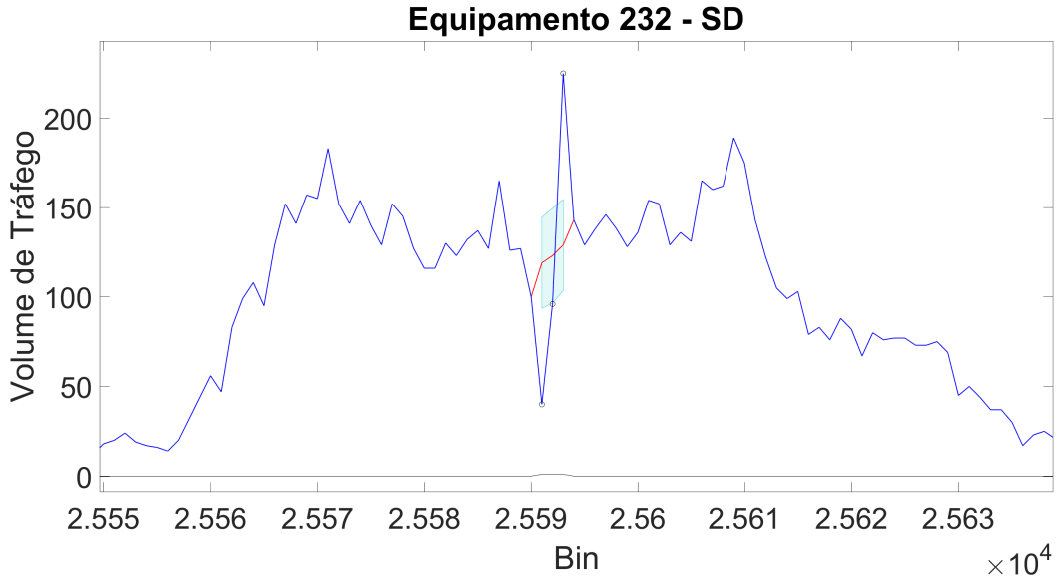


Figura 3.18: Imputação de um gap de 45min, 3 bins, no Trecho 06. O modelo utilizado para a reconstrução deste gap foi o modelo 4.

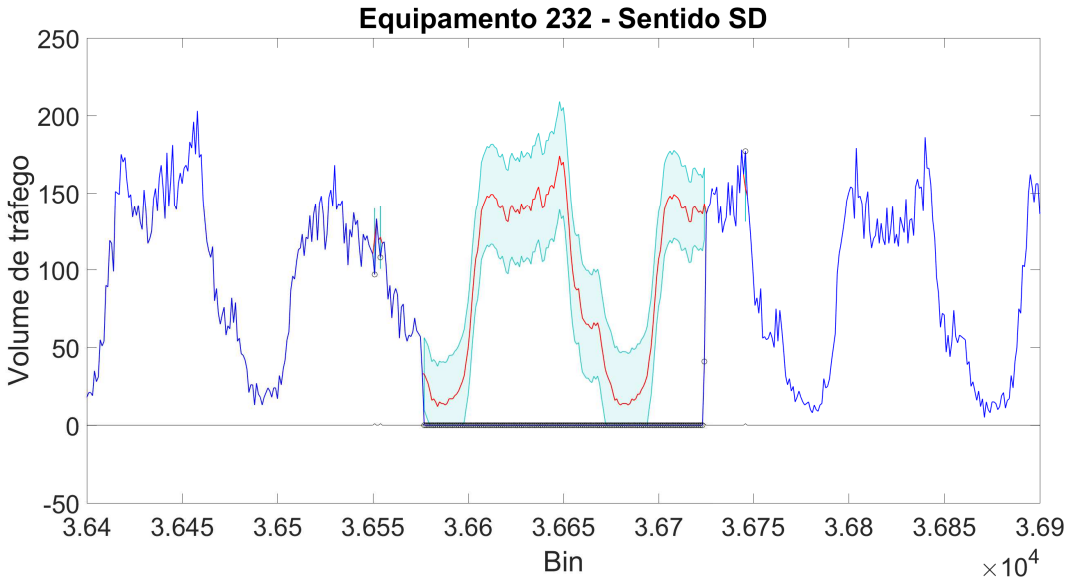


Figura 3.19: Imputação de um gap de ± 1 dia e meio, ∼ 144 bins, no Trecho 08. O modelo utilizado para a reconstrução deste gap foi o modelo 1. Note a captura da sazonalidade diária e das rápidas variações em períodos diurnos.

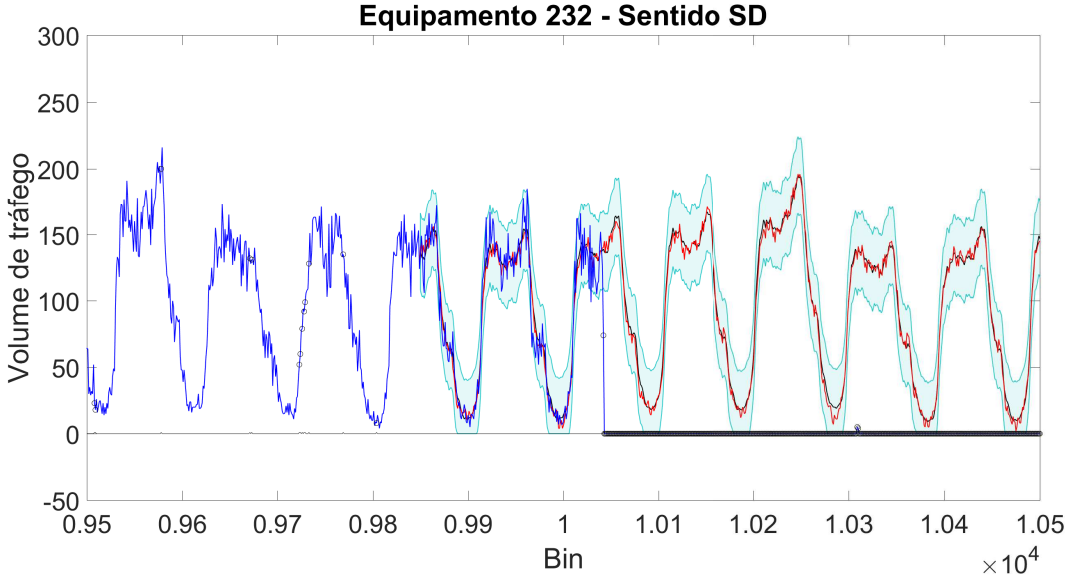


Figura 3.20: Imputação de um gap longo de ± 10 dias, ∼ 1000 bins, no Trecho 02. O modelo utilizado para a reconstrução deste gap foi o modelo 1. A linha preta é a média da distribuição preditiva. O sombreado azul seu intervalo de confiança de 95%. A linha vermelha denota uma amostra aleatória desta mesma dstribuição. Note a captura da sazonalidade semanal e diária.

Capítulo 4

Conclusão

O método do aprendizado de máquina via processos Gaussianos se mostrou uma ferramenta bastante efetiva para abordar o problema de imputação de dados ausentes em séries temporais provenientes do PNCT. Isto se deve pelo fato de os processos Gaussianos serem uma extensão natural da regressão linear Bayesiana para uma classe muito mais flexível de modelos, como foi visto nessa dissertação. A interpretabilidade dos hiperparâmetros nos possibilita uma modelagem mais prática abarcando melhores resultados após o passo de otimização. Entretanto vimos também o alto custo computacional envolvido. A regressão via processos Gaussianos realizada de maneira exata exige a inversão de uma matriz que é diretamente proporcional a quantidade de dados de treino e possui complexidade computacional de $\mathcal{O}(n^3)$, o que torna impraticável a regressão com uma quantidade massiva de dados que é o caso das séries temporais do PNCT. A metodologia proposta além de lidar bem com esta restrição dividindo o problema em vários problemas menores e aliando um alto poder computacional, também nos permitiu abordar bem o problema da falta de conhecimento sobre uma função de covariância, onde ficou claro pelos resultados obtidos ao realizar uma seleção de modelos dentre alguns que possivelmente explicariam bem os dados de alguma maneira.

4.1 Trabalhos futuros

Neste trabalho verificou-se que para gaps pequenos, a metodologia se apresentou mais efetiva do que para gaps longos, mesmo porque o problema de imputação de gaps longos é bastante complicado. Uma solução para este problema parece ser utilizar informação correlacionada proveniente de múltiplos equipamentos de contagem de tráfego próximos [18]. No caso do PNCT, por exemplo, utilizar equipamentos de uma mesma rodovia, ou região, em trechos diferentes. Um exemplo recente é apresentado na fig. 4.1 que mostra a reconstrução da série temporal do equipamento 187 utilizando também a série temporal do equipamento 190, cuja distância euclidiana entre eles é aproximadamente 50 km [32], porém não na mesma rodovia.

Como o custo computacional da regressão via processos Gaussianos é alto, acaba piorando ao utilizar múltiplas séries temporais. Para lidar com este problema, técnicas como aproximações esparsas [1] ou multi-task IVM [22] são mais indicadas. Outra forma de lidar com este problema se encontra em melhor utilizar o poder computacional, como por exemplo realizar uma paralelização com GPUs.

Outro problema que pode ser explorado é o da seleção de dados ativos [17] em que realiza-se uma regressão em tempo real com múltiplas séries temporais onde utilizamos algum critério baseado em um instante à frente no tempo de forma a selecionar o dado mais informativo para a regressão. Desta forma, dependendo de como a regressão está sendo realizada, podemos extrair informações de um conjunto de sensores como, por exemplo, aqueles que são mais informativos que outros.

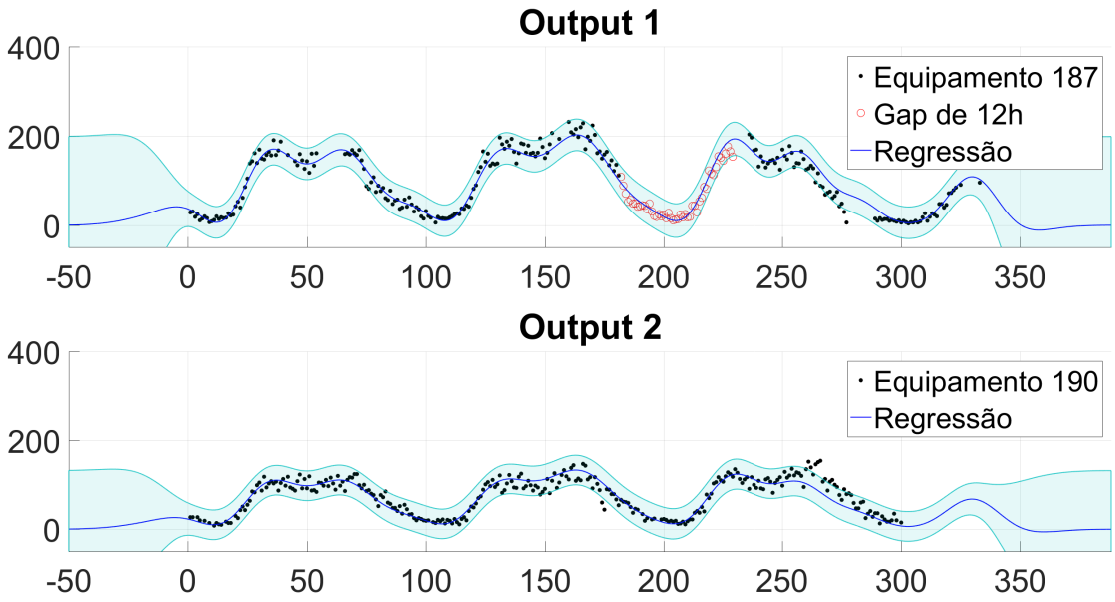


Figura 4.1: Reconstrução de um gap de 12h utilizando informação correlata proveniente de múltiplas séries temporais.

Apêndice A

Identidades Gaussianas

As seguintes identidades são muito úteis para alguns dos desenvolvimentos realizados. Todas se baseiam na distribuição Normal multivariada. Para uma lista mais extensa de identidades, ver [10].

A.1 Função densidade de probabilidade

Seja $\boldsymbol{x}$ uma variável aleatória multivariada m-dimensional. Sua função densidade de probabilidade é da seguinte forma:

$$p(\boldsymbol{x}\,|\,\boldsymbol{\mu},\boldsymbol{A}) = \mathcal{N}(\boldsymbol{x}\,|\,\boldsymbol{\mu},\boldsymbol{A}) = \frac{1}{\sqrt{\det 2\pi\boldsymbol{A}}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{A}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{A.1}$$

onde $\boldsymbol{\mu} \in \mathbb{R}^m$ é sua média e $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ é sua matriz de covariância simétrica positiva definida.

A.2 Identidades

A.2.1 Distribuição marginal e condicionada

Primeiro resultado

Considere a seguinte distribuição de probabilidade conjunta:

$$p(\boldsymbol{x},\boldsymbol{y}\,|\,I) \triangleq \mathcal{N}\left(\left[\begin{array}{c}\boldsymbol{x}\\ \boldsymbol{y}\end{array}\right]\middle|\left[\begin{array}{c}\boldsymbol{\mu}\\ \boldsymbol{\nu}\end{array}\right], \left[\begin{array}{cc}\boldsymbol{A} & \boldsymbol{C}\\ \boldsymbol{C}^\top & \boldsymbol{B}\end{array}\right]\right) \tag{A.2}$$

então suas distribuições marginal e condicionada são respectivamente:

$$p(\boldsymbol{x}\,|\,I) = \mathcal{N}(\boldsymbol{x}\,|\,\boldsymbol{\mu},\boldsymbol{A}) \tag{A.3a}$$

$$p(\boldsymbol{x}\,|\,\boldsymbol{y},I) = \mathcal{N}(\boldsymbol{x}\,|\,\boldsymbol{\mu}+\boldsymbol{C}\boldsymbol{B}^{-1}(\boldsymbol{y}-\boldsymbol{\nu}), \boldsymbol{A}-\boldsymbol{C}\boldsymbol{B}^{-1}\boldsymbol{C}^\top) \tag{A.3b}$$

Segundo resultado

Considere as seguintes distribuições de probabilidade:

$$\begin{aligned} p(\boldsymbol{x}\,|\,I) &\triangleq \mathcal{N}(\boldsymbol{x}\,|\,\boldsymbol{\mu},\boldsymbol{A}) \\ p(\boldsymbol{y}\,|\,\boldsymbol{x},I) &\triangleq \mathcal{N}(\boldsymbol{y}\,|\,\boldsymbol{M}\boldsymbol{x}+\boldsymbol{c},\boldsymbol{L}) \end{aligned} \tag{A.4}$$

então a distribuição conjunta de $\boldsymbol{x}$ e $\boldsymbol{y}$ pode ser escrita como:

$$p(\boldsymbol{x},\boldsymbol{y}|I) \triangleq \mathcal{N}\left(\left[\begin{array}{c}\boldsymbol{x}\\ \boldsymbol{y}\end{array}\right] \middle| \left[\begin{array}{c}\boldsymbol{\mu}\\ \boldsymbol{M\mu}+\boldsymbol{c}\end{array}\right], \left[\begin{array}{cc}\boldsymbol{A} & \boldsymbol{AM}^\top\\ \boldsymbol{MA} & \boldsymbol{L}+\boldsymbol{MAM}^\top\end{array}\right]\right) \tag{A.5}$$

e então, utilizando eq.A.3a e eq.A.3b, temos:

$$p(\boldsymbol{y}|I) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{M\mu}+\boldsymbol{c}, \boldsymbol{L}+\boldsymbol{MAM}^\top) \tag{A.6a}$$
$$p(\boldsymbol{x}|\boldsymbol{y},I) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}+\boldsymbol{\Gamma}(\boldsymbol{y}-\boldsymbol{M\mu}-\boldsymbol{c}), \boldsymbol{A}-\boldsymbol{\Gamma MA}) \tag{A.6b}$$

onde

$$\boldsymbol{\Gamma} = \boldsymbol{AM}^\top(\boldsymbol{L}+\boldsymbol{MAM}^\top)^{-1} \tag{A.7}$$

A.2.2 Produto de duas Gaussianas

O produto de duas distribuições normais multivariadas é proporcional a outra distribuição normal multivariada:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{A})\mathcal{N}(\boldsymbol{x}|\boldsymbol{b},\boldsymbol{B}) = Z\mathcal{N}(\boldsymbol{x}|\boldsymbol{c},\boldsymbol{C}) \tag{A.8}$$

onde a média e a matriz de covariância resultantes são:

$$\boldsymbol{C} = (\boldsymbol{A}^{-1}+\boldsymbol{B}^{-1})^{-1} \quad \text{e} \quad \boldsymbol{c} = \boldsymbol{C}(\boldsymbol{A}^{-1}\boldsymbol{a}+\boldsymbol{B}^{-1}\boldsymbol{b}) \tag{A.9}$$

e a constante de normalização:

$$Z = \mathcal{N}(\boldsymbol{a}|\boldsymbol{b},(\boldsymbol{A}+\boldsymbol{B})) \tag{A.10}$$

Bibliografia

Livros

[1] Rasmussen, C.E., Williams, C.K.I., *Gaussian Processes for Machine Learning.* The MIT Press, 2006.

[2] Bishop, C. M., *Pattern Recognition and Machine Learning.* Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006.

[3] MacKay, D. J. C., *Information Theory, Inference and Learning Algorithms.* Cambridge University Press New York, NY, USA, 2002.

[4] AASHTO. *Guidelines for Traffic Data Programs*, 2nd edition, American Association of State Highway and Transportation Officials (AASHTO), Washington, D.C., 2009.

[5] Scholkopf, B., Smola, A. J., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press Cambridge, MA, USA, 2002.

[6] Wasserman, L., *All of Statistics: a concise course in statistical inference.* Springer, New York, 2004.

[7] Abramowitz, M., Stegun, I. A. *Handobook of Mathematical Functions With Formulas, Graphs and Mathematical Tables.* National Bureau of Standards, Applied Mathematics Series 55, 1972.

[8] Quarteroni, A., Saleri, F., Gervasio, P., *Scientific Computing with MATLAB and Octave.* Texts in computational science and engineering, Springer, 2014.

[9] Altman, Y. M., *Accelerating MATLAB Performance: 1001 tips to speed up MATLAB programs.* Chapman & Hall/CRC, 2014.

[10] Petersen, K. B., Pedersen, M. S., *The Matrix Cookbook.* Technical University of Denmark, 2012. `http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf`

[11] Rasmussen, C.E., Williams, C.K.I., *The GPML Toolbox version 4.0.* 2016. `http://www.gaussianprocess.org/gpml/code/matlab/doc/manual.pdf`

Artigos

Gaussian process

[12] MacKay, D. J. C., *Introduction to Gaussian Processes.* In Bishop, C. M., editor, Neural Networks and Machine Learning. Springer-Verlag, 1998.

[13] MacKay, D. J. C., *Comparison of Approximate Methods for Handling Hyperparameters.* Neural Computation, 11(5):1035-1068, 1999.

[14] MacKay, D. J. C., *Bayesian Interpolation.* Neural Computation, 4(3):415-447, pp. xiii, xvi, 109, 1992b.

[15] Rasmussen, C. E., Ghahramani, Z., *Occam's Razor.* In Advances in Neural Information Processing Systems, 13, 294-300, MIT Press, 2001.

[16] Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., Aigrin, S. *Gaussian processes for timeseries modelling.* Philosophical Transactions of the Royal Society (Part A), 2012.

[17] Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., Aigrin, S. *Active data selection for sensor networks with faults and changepoints.* Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications, 2010.

[18] Álvarez, M.A., Lawrence, N.D., *Computationally efficient convolved multiple output Gaussian Processes.* J. Mach. Learn. Res. No. 12, pp. 1459-1500, 2011.

[19] Neal, R. M., *Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification..* Technical Report 9702, Department of Statistics, University of Toronto, 1997.

[20] Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J. B., Ghahramani, Z., *Structure Discovery in Nonparametric Regression through Compositional Kernel Search.* ARXIV, eprint arXiv:1302.4922, 2013.

[21] Salakhutdinov, R. and Hinton, G., *Using deep belief nets to learn covariance kernels for Gaussian processes.* Advances in Neural information processing systems, 20:1249?1256, 2008.

[22] Lawrence, N. D., Platt, J. C., *Learning to learn with the informative vector machine.* ICML '04 Proceedings of the twenty-first international conference on Machine learning, p.65, Banff, Alberta, Canada, 2004

Transportation

[23] Xie, Y., Zhao, K., Sun, Y., Chen, D., *Gaussian Processes for short-term traffic volume forecasting.* Transportation Research Record: Journal of the Transportation Research Board, No. 2165, 2010, pp. 69-78.

[24] Zhong, M., Lingras, P.J., and Sharma, S.C., *Estimation of missing traffic counts using factor, genetic, neural, and regression techniques.* Transport. Res. Part C Emerg. Technol., No. 12, pp. 139-166, 2004.

[25] Zhong, M., Sharma, S., *A hybrid approach to integrate fuzzy C-means based imputation method with genetic algorithm for missing traffic volume data estimation.* Transport. Res. Part C Emerg. Technol., No. 51, pp. 29-40, 2015.

[26] Ghosh, B., Basu, B., O?Mahony, M. M., *Time-series modeling for forecasting vehicular traffic flow in Dublin.* Proceedings of the 84th Annual Meeting of Transportation Research Board, January, Washington, DC. 2005.

[27] Ramsey, B., Hayden, G., *AutoCounts: a way to analyse automatic traffic count data.* Traffic Eng. Control No. 35 (4), pp. 245-246, 1994.

[28] Redfern, E.J., Waston, S.M., Tight, M.R., Clark, S.D., *A comparative assessment of current and new techniques for detecting outliers and estimating missing values in transport related time series data.* Proceedings of Highways and Planning Summer Annual Meeting, Institute of Science and Technology, University of Manchester, England, 1993.

[29] Williams, B.M., Hoel, L.A., *Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results.* J. Transport. Eng. No. 129 (6), pp. 664?672, 2003.

[30] Qu, L., Li, L., Zhang, Y., Hu, J., *PPCA-based missing data imputation for traffic flow volume: a systematical approach.* IEEE Trans. Intell. Transport. Syst. No. 10 (3), pp. 512?522. 2009.

Páginas da Web

[31] The Gaussian Process Website, `http://www.gaussianprocess.org/`.

[32] Plano Nacional de Contagem de Tráfego, `http://servicos.dnit.gov.br/dadospnct/`. Departamento Nacional de Infraestrutura de Transportes.

[33] Duvenaud, D., The Kernel Cookbook: Advice on Covariance Functions, `http://www.cs.toronto.edu/~duvenaud/cookbook/index.html`.