Tema
Tendências em Matemática Aplicada e Computacional, **21**, N. 1 (2020), 117-131

www.scielo.br/tema
doi: 10.5540/tema.2020.021.01.00117

Combinando Métodos de Aprendizado Supervisionado para a Melhoria da Previsão do *Redshift* de Galáxias

M. MUSETTI[1] e R. IZBICKI[2*]

Recebido em Março 22, 2019 / Aceito em Novembro 11, 2019

RESUMO. Um problema fundamental em cosmologia é estimar *redshifts* de galáxias com base em dados fotométricos. Por exemplo a *Sloan Digital Sky Survey* (SDSS) já coletou dados fotométricos relativos a cerca de um bilhão de objetos para os quais é necessário estimar os respectivos *redshifts*. Tradicionalmente, essa tarefa é resolvida utilizando-se métodos de aprendizado de máquina. Neste trabalho, mostramos como métodos existentes podem ser combinados de forma a se obter estimativas ainda mais precisas para os *redshifts* de galáxias. Abordamos este problema sob duas óticas: (i) estimação da regressão do *redshift* y nas covariáveis fotométricas $\mathbf{x}$, $\mathbf{E}[Y|\mathbf{x}]$, e (ii) estimação da função densidade condicional $f(y|\mathbf{x})$. Aplicamos as técnicas propostas para um banco de dados provenientes do SDSS e concluímos que as predições combinadas são de fato mais precisas que os métodos individuais.

Palavras-chave: aprendizado de máquina, *stacking*, funções densidades condicionais, cosmologia.

1 INTRODUÇÃO

O *redshift* de uma galáxia é essencialmente uma medida da distância entre essa galáxia e a Terra. Estimar essa quantidade com alta precisão é um problema chave em cosmologia: Isso ocorre pois, como a luz necessita de tempo para percorrer uma dada distância, imagens de objetos que estão mais distantes em nosso universo refletem como o universo era há muitos anos atrás. Quanto mais distante o objeto, mais antiga é a imagem obtida. Assim, estimar o *redshift* de um objeto é fundamental para que seja possível fazer inferências precisas sobre a evolução do universo [2].

Existem duas formas de estimar o *redshift* de uma galáxia: a espectroscopia e a fotometria. A espectroscopia é baseada na decomposição da luz emitida por um objeto. Avaliando-se a localização das linhas de absorção no seu espectro, é possível estimar o *redshift* da galáxia com grande acurácia. Infelizmente, essa é uma técnica extremamente cara e lenta, de modo que não

*Autor correspondente: Rafael Izbicki – E-mail: rafaelizbicki@gmail.com
[1]Departamento de Estatística, UFSCar - Univ Federal de São Carlos, 13565-905 São Carlos, SP, Brasil. E-mail: marcelamusetti@hotmail.com https://orcid.org/0000-0001-7008-7445
[2]Departamento de Estatística, UFSCar - Univ Federal de São Carlos, 13565-905 São Carlos, SP, Brasil. E-mail: rafaelizbicki@gmail.com https://orcid.org/0000-0003-0379-9690

é possível aplicá-la a muitos objetos. No entanto, a fotometria é uma técnica de medição mais rápida e barata, que é baseada na observação de imagens de uma galáxia em diferentes bandas fotométricas. Infelizmente, a fotometria produz estimativas de baixa resolução. Como há bilhões de galáxias em nosso universo, não é possível aplicar a espectroscopia a todas elas. Assim, é importante criar métodos que sejam capazes de utilizar a fotometria da melhor forma possível para conseguir estimativas precisas dos *redshifts*, de modo que não seja necessário aplicar a espectroscopia para todas as galáxias de interesse.

Na prática, a fotometria consiste na medição de diversas cores de uma galáxia a partir de sua imagem. Para que essas cores sejam utilizadas para estimar *redshifts*, coleta-se um conjunto de dados em que são conhecidos tantos os dados fotométricos (i.e., as cores das galáxias) quanto o *redshift* de cada uma delas, sendo esses obtidos através de espectroscopia. A partir dessas informações, utiliza-se métodos de aprendizado de máquina para fazer a predição do *redshift* em novas galáxias que não foram submetidas ao processo de espectroscopia [2, 8, 13, 14].

Alguns dos métodos tradicionalmente utilizados para resolver esse problema de predição são os k-vizinhos mais próximos [6], redes neurais artificiais [25], máquinas de vetores de suporte [23], processos gaussianos [1] entre outros. Formalmente, esses métodos visam estimar o valor esperado do *redshift* de uma galáxia Y dadas as covariáveis fotométricas $\mathbf{x}$, i.e., $\mathbf{E}[Y|\mathbf{x}]$. Além dessa abordagem, há um interesse crescente na estimação da *função de densidade condicional* $f(y|\mathbf{x})$, uma vez que essa tem mais informação que seu valor esperado e é capaz de lidar com multimodalidades e assimetrias na distribuição do rótulo [7, 12, 14, 16, 18, 19, 24].

Neste trabalho, mostramos que métodos existentes de predição de *redshifts* podem ser *combinados* a fim de obter estimativas ainda mais precisas para cada galáxia. Isto é, mostramos que, ao invés de simplesmente selecionar o melhor método de predição, é vantajoso combinar as predições fornecidas por cada modelo. Em particular, aplicamos métodos de *stacking* usuais em aprendizado de máquina para executar essa tarefa [5, 28]. Também apresentamos um novo método de *stacking* que visa combinar funções de densidades condicionais. Ainda que tenhamos aplicado as técnicas aqui apresentadas para um problema de astronomia, elas são muito mais gerais e podem ser aplicadas a diferentes contextos, em particular em situações nas quais é muito caro de se obter o rótulo real.

O restante do artigo está dividido da seguinte maneira. A Seção 2 apresenta os dados analisados, assim como os métodos usados para predizer o *redshift* de cada galáxia. Em particular, apresentamos os métodos que visam combinar os resultados de técnicas já existentes. A Seção 3 apresenta os resultados das técnicas descritas ao banco de dados. Finalmente, a Seção 4 conclui o artigo com considerações finais.

2 METODOLOGIA

2.1 Dados

Os dados fotométricos e espectroscópicos utilizados fazem parte do *Sloan Digital Sky Survey* (SDSS) [26], que contém imagens de mais de 200 milhões de galáxias. Todas foram medidas utilizando a fotometria e cerca de um milhão foram também medidas por espectroscopia. Neste trabalho, foi utilizado um subconjunto com 300 mil observações, que foi aleatoriamente separado em treinamento dos algoritmos base[1] (150 mil), validação dos algoritmos base (20 mil), treinamento dos algoritmos de *stacking* (100 mil), validação dos algoritmos de *stacking* (20 mil) e teste[2] (10 mil) (vide a Seção 2.3.2 para mais detalhes). A escolha desses números foi motivada levando-se em conta que (i) o conjunto para treinar os algoritmos base deve ser grande (pois esses métodos são complexos), (ii) a validação (i.e., escolha de *tuning parameters*) em geral é um processo mais simples, de modo que apenas 20 mil observações são suficientes para isso, (iii) o treinamento dos algoritmos de *stacking* também é mais simples que o treinamento dos regressores base (pois há poucos regressores base), (iv) o estimador do risco é apenas uma média, de modo que 10 mil observações são suficientes para isso.

O SDSS faz medições de cinco magnitudes, capturando imagens através de cinco filtros correspondentes: o verde (g), vermelho (r), ultravioleta (u) e dois comprimentos de onda infravermelhos (i e z). A partir dessas magnitudes, são calculadas as cores de cada galáxia através da subtração de magnitudes consecutivas: $u-g$, $g-r$, $r-i$ e assim por diante. Cada magnitude também está associada a um erro medição. Esses erros também são medidos e denotados por e_g, e_r e assim sucessivamente. A partir dessas medidas, define-se o vetor de covariáveis associadas à i-ésima galáxia, $\mathbf{x}_i$, como o vetor de valores de medição das cores ($u-g$, $g-r$, $r-i$, etc.) obtidos através da fotometria juntamente com os erros de medição associados a cada uma das magnitudes. Além disso, a variável resposta associada à i-ésima galáxia, y_i, é o valor do *redshift* desta unidade amostral obtido utilizando espectroscopia. O erro de medição da espectroscopia é negligenciável [8], de modo que tratamos y_i como sendo o *redshift* real da i-ésima galáxia.

2.2 Métodos de predição de *redshifts*

Nesta seção, descrevemos os métodos utilizados neste artigo para obter predições para o *redshift* de uma galáxia. Posteriormente (Seção 2.3), esses métodos são combinados de forma a ser obter predições ainda melhores.

Formalmente, deseja-se utilizar a amostra $(\mathbf{X}_1,Y_1),...,(\mathbf{X}_n,Y_n) \in \mathbb{R}^d \times \mathscr{Y}$, para encontrar uma função $g : \mathbb{R}^d \to \mathscr{Y}$ que faça predições precisas para o *redshift* de novas galáxias a partir de suas covariáveis. Para quantificar o quão boas são as predições produzidas pela função g, será utilizada a função de risco quadrática, $R(g) = E\left[(g(\mathbf{X})-Y)^2\right]$, uma vez que Y é uma variável contínua. Note que tal escolha implica que a melhor função de predição é a função de regressão, $\mathbf{E}[Y|\mathbf{x}]$ [9, 17].

[1] Isto é, os modelos base a serem combinados.

[2] Isto é, o conjunto usado para estimar o risco de cada modelo ajustado.

2.2.1 FlexCode

O FlexCode é um método que visa estimar a função densidade condicional $f(y|\mathbf{x})$. Para tanto, sua ideia chave (vide [13] para mais detalhes) é expandir $f(y|\mathbf{x})$ em uma base ortonormal $(\phi_i(y))_{i\in\mathbb{N}}$ (como, por exemplo, a base de Fourier):

$$f(y|\mathbf{x}) = \sum_{j\in\mathbb{N}} \beta_j(\mathbf{x})\phi_j(y).$$

Devido à ortogonalidade, $\beta_j(\mathbf{x}) = E[\phi_j(Y)|\mathbf{x}]$. Dessa forma, os β_j's são estimados por regressão: $\hat{\beta}_j(\mathbf{x}) = \hat{E}[\phi_j(Y)|\mathbf{x}]$, isto é, regredindo cada $\phi_j(Y)$ em $\mathbf{x}$. Por fim, o estimador FlexCode é definido por

$$\hat{f}(y|\mathbf{x}) = \sum_{j=1}^{J} \hat{\beta}_j(\mathbf{x})\phi_j(y),$$

em que J é escolhido com validação cruzada. Note que diferentes métodos de regressão levam a diferentes estimativas da função densidade condicional e o método escolhido pode ter uma grande influência no desempenho do FlexCode. Para maximizar o ganho dos métodos de *stacking*, as escolhas nesse artigo foram feitas de modo que cada método de regressão tivesse uma natureza bastante distinta um do outro. Mais especificamente, os métodos de regressão considerados neste trabalho são florestas aleatórias, lasso, KNN, XGBoost [9, 11], de modo que quatro versões do FlexCode serão contempladas.

Para criar estimadores pontuais a partir da função densidade estimada $\widehat{f}(y|\mathbf{x})$, utilizamos três resumos de tal função densidade: a média, a moda e a mediana.

2.2.2 GPZ

O método baseado em processos gaussianos esparsos heterocedásticos (GPZ, [1]) retorna um estimador pontual de $\mathbf{E}[Y|\mathbf{x}]$, e não a função densidade como o FlexCode. O processo é definido por uma forma semiparamétrica construída a partir de pesos. Mais especificamente, assume-se que Y_i é gerado por uma combinação linear de m funções não lineares de $\mathbf{x}_i$: $\phi(\mathbf{x}_i) = [\phi_1(\mathbf{x}_i), \ldots, \phi_m(\mathbf{x}_i)] \in \mathbb{R}^m$.

$$Y_i = \phi(\mathbf{x}_i)\mathbf{w} + \varepsilon_i,$$

em que $\varepsilon_i \sim N(0, \beta^{-1}(\mathbf{x}_i))$, $\mathbf{w}$ é o vetor de parâmetros a ser estimado e $\phi_j(\mathbf{x}_i)$ é dado por

$$\phi_j(\mathbf{x}_i) = \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{p}_j)^t \Gamma_j^t \Gamma_j (\mathbf{x}_i - \mathbf{p}_j)}{2}\right\},$$

em que $\mathbf{p}_j$ são conjuntos de vetores de base associados às funções de base, $\Gamma_j^t\Gamma_j$ são matrizes de precisão sob medida associadas a cada função de base. Além disso, assume-se que $\beta(\mathbf{x}) = \exp(\phi(\mathbf{x})u + b)$. Os parâmetros desse processo, incluindo u e b, são estimados com um método bayesiano. Para mais detalhes, vide [1].

2.3 Combinando preditores pontuais

Nesta seção, descrevemos os métodos utilizados para criar uma função de predição combinada (aqui denotada por G) a partir das funções já construídas utilizando os métodos descritos na seção anterior (aqui denotadas por $g_i = g_i(\mathbf{x}), i = 1, \ldots, B$).

2.3.1 Média simples e mediana

A forma mais simples de combinar $g_1, g_2, ..., g_B$ é utilizando a média dos valores previstos ou seja,

$$G(\mathbf{x}) := \frac{1}{B}\sum_{b=1}^{B} g_b(\mathbf{x}).$$

Alternativamente, pode-se utilizar a mediana dessas predições.

2.3.2 *Stacking*

A ideia chave do método de *stacking* [11, 28] é utilizar as predições $g_1(\mathbf{x}), \ldots, g_B(\mathbf{x})$ como entradas para algoritmos de aprendizado supervisionado com a finalidade de obter predições combinadas.

Formalmente, após obter as funções de predição $g_i : \mathbb{R}^d \to \mathbb{R}, \quad i = 1, ..., B$, utilizando-se os algoritmos descritos nas outras seções, aplica-se cada uma delas a um conjunto de dados não utilizados para seu treinamento, $(\tilde{\mathbf{X}}_1, \tilde{Y}_1), ..., (\tilde{\mathbf{X}}_{\tilde{n}}, \tilde{Y}_{\tilde{n}}) \in \mathbb{R}^d \times \mathscr{Y}$. Seja $\tilde{\mathbf{w}}_i := (g_1(\tilde{\mathbf{x}}_i), ..., g_B(\tilde{\mathbf{x}}_i)) \in \mathbb{R}^B$ o vetor que contém o valor de cada função de predição aplicada na i-ésima unidade amostral $\mathbf{x}_i$. Aplica-se então um método de aprendizado supervisionado (como regressão linear, redes neurais ou florestas aleatórias [9]) ao conjunto $(\tilde{\mathbf{w}}_1, \tilde{Y}_1), ..., (\tilde{\mathbf{w}}_{\tilde{n}}, \tilde{Y}_{\tilde{n}})$, com a finalidade de se obter uma função de predição $h : \mathbb{R}^B \to \mathbb{R}$. Feito isso, define-se $G(\mathbf{x}) := h(g_1(\mathbf{x}), ..., g_B(\mathbf{x}))$ como a função resultante que combina os B métodos de predição já criados. Neste trabalho, criamos h a partir dos seguintes métodos de aprendizado supervisionado: florestas aleatórias [3], KNN [27], lasso [20] e XGBoost [4].

2.4 Combinando funções densidades condicionais

Além das técnicas de combinação de preditores pontuais, também investigamos duas formas de combinar os estimadores da função densidade condicional $f(y|\mathbf{x})$.

2.4.1 Média simples

Uma maneira de combinar as estimativas $\widehat{f_i}(y|\mathbf{x}),\ i = 1, \ldots, B$ é calcular a média simples entre elas, ou seja,

$$\hat{f}(y|\mathbf{x}) = \frac{1}{B}\sum_{i=1}^{B} \hat{f}_i(y|\mathbf{x}).$$

Como cada $\widehat{f_i}$ é uma função densidade, $\widehat{f}$ também o é, pois necessariamente é positiva e integra um em y.

2.4.2 Média ponderada

Uma alternativa à média simples é utilizar uma média ponderada, uma vez que cada método pode ter um desempenho diferente. Para isso, consideramos a seguinte função de risco para estimadores de função densidade condicional:

$$\begin{aligned} R(\hat{f},f) &= \int\int \left[\left(\hat{f}(y|\mathbf{x}) - f(y|\mathbf{x})\right)^2\right] dP(\mathbf{x})dy \\ &= \int\int \hat{f}^2(y|\mathbf{x})dP(\mathbf{x})dy - \int\int 2\hat{f}(y|\mathbf{x})f(y|\mathbf{x})dP(\mathbf{x})dy + \int\int f^2(y|\mathbf{x})dP(\mathbf{x})dy. \end{aligned}$$

Denotando por $(\mathbf{X}'_1, Y'_1), ..., (\mathbf{X}'_n, Y'_n)$ um conjunto de dados não utilizados para estimar a função densidade, pode-se estimar esse risco (a menos de uma constante) por [13]

$$\hat{R}(\hat{f},f) := \frac{1}{n}\sum_{i=1}^{n}\int \hat{f}^2(y|\mathbf{x}'_i)dy - \frac{2}{n}\sum_{i=1}^{n}\hat{f}(y'_i|\mathbf{x}'_i).$$

A ideia chave para criar um estimador da função densidade condicional utilizando-se média ponderada, $\hat{f}^{\alpha}(y|\mathbf{x}) = \sum_{i=1}^{B}\alpha_i f(y|\mathbf{x})$, é buscar os valores de $\alpha_1, \ldots, \alpha_B$ que minimizam o risco estimado, sujeito à restrição de que tais valores sejam de fato pesos. Isto é, busca-se por

$$\arg \min_{\alpha:\alpha_i>0, \sum_{i=1}^{B}\alpha_i=1} \widehat{R}(\widehat{f^{\alpha}}, f).$$

Como

$$\widehat{R}(\widehat{f^{\alpha}}, f) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k,l=1}^{B}\int \hat{f}_k(y|\mathbf{x}'_i)\hat{f}_l(y|\mathbf{x}'_i)dy - \frac{2}{n}\sum_{i=1}^{n}\sum_{k=1}^{B}\hat{f}_k(y'_i|\mathbf{x}'_i),$$

a escolha ótima é dada pela solução do problema

$$\arg \min_{\alpha:\alpha_i>0, \sum_{i=1}^{B}\alpha_i=1} \alpha^{\mathbf{t}}\mathbb{B}\alpha - 2\alpha^{\mathbf{t}}\mathbf{b}, \tag{2.1}$$

em que $\mathbb{B}$ é a matriz

$$\mathbb{B} = \left[\frac{1}{n}\sum_{i=1}^{n}\int f_b(y|\mathbf{x}'_i)f_{b'}(y|\mathbf{x}'_i)dy\right]_{b,b'=1}^{B}$$

e o vetor $\mathbf{b}$ é dado pela equação

$$\mathbf{b} = \left[\frac{1}{n}\sum_{i=1}^{n}\hat{f}_b(y'_i|\mathbf{x}'_i)\right]_{b=1}^{B}.$$

Tal problema de otimização pode ser resolvido numericamente utilizando métodos de programação quadrática [21]. Neste artigo, utilizamos a função `solve.QP` do pacote `quadprog` para otimizar essa função objetivo [22].

3 RESULTADOS

O FlexCode foi ajustado atráves do pacote em R [15]. Já o GPZ foi ajustado com o código em Python disponível em `https://github.com/OxfordML/GPz`. A Tabela 1 mostra os riscos quadráticos estimados de cada um dos modelos de predição pontual investigados. Os métodos combinados por *stacking* dominaram uniformemente todos os métodos individuais. Por outro lado, combinar métodos individuais utilizando média ou mediana foi subótimo. Isso possivelmente se deve ao fato de que, nessas abordagens, todos os modelos individuais contribuem igualmente para a predição combinada. Isto é, essas abordagens não levam em conta que métodos diferentes podem possuir desempenhos distintos.

Tabela 1: Risco estimado e tempo de ajuste de cada preditor pontual. Para os métodos baseados em combinação/*stacking*, o tempo se refere apenas para a combinação dos modelos, e portanto não leva em conta os ajustes dos classificadores base.

Método	**Variação**	**Risco**	**Tempo computacional**
STACKING	Lasso	2,64	1s
STACKING	XGBoost	2,66	7min
STACKING	Floresta	2,67	20min24s
STACKING	KNN	2,71	43s
FLEXCODE XGBoost	Mediana	2,76	18min1s
GPZ Heterocedástico	-	2,81	4h
FLEXCODE XGBoost	Moda	3,59	18min1s
FLEXCODE XGBoost	Média	3,72	18min1s
COMBINAÇÃO	Mediana	3,91	$<$ 1s
FLEXCODE Lasso	Mediana	4,46	5min36s
COMBINAÇÃO	Média	4,52	$<$ 1s
FLEXCODE Lasso	Média	4,72	5min36s
FLEXCODE Floresta	Média	5,98	60min29s
FLEXCODE Floresta	Mediana	6,19	60min29s
FLEXCODE Lasso	Moda	8,08	5min36s
FLEXCODE Floresta	Moda	10,13	60min29s
FLEXCODE KNN	Média	12,87	43s
FLEXCODE KNN	Moda	14,14	43s
FLEXCODE KNN	Mediana	13,83	43s

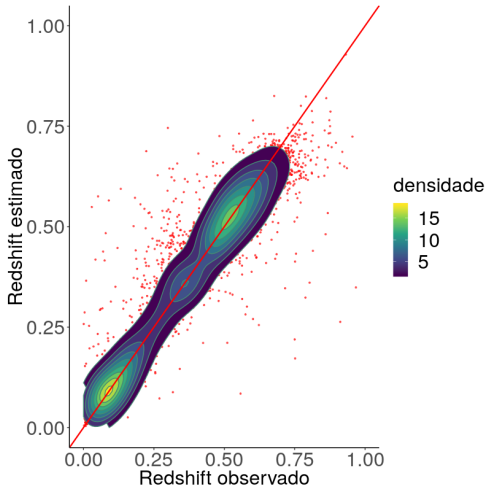
Observa-se também que os riscos do método GPZ e do FlexCode estimado via regressão por XGBoost (utilizando a mediana) produzem os menores riscos dentre os modelos individuais. Por fim, nota-se que todas as predições feitas através do KNN possui resultados inferiores aos demais métodos.

Com a finalidade de compreender o modelo ajustado pelo *stacking* via XGBoost, calculamos o ganho médio (isto é, o quanto cada covariável contribui para reduzir o erro quadrático médio do modelo, [4]) de cada modelo individual (Tabela 2). Quanto maior o valor dessa medida, maior é a contribuição deste modelo individual na diminuição do erro de predição do modelo conjunto. Os métodos mais importantes são o GPZ e o FlexCode XGBoost com a variação mediana. De fato, esses foram justamente os melhores métodos individuais segundo a Tabela 1. Além disso, os demais apresentam um ganho nulo ou muito próximo de zero, o que indica que esses métodos não são relevantes para o cálculo das predições do *stacking* via XGBoost. Do ponto de vista prático, pode ser vantajoso remover modelos com ganho nulo. De fato, essa remoção não deve afetar o desempenho estatístico (i.e., preditivo) do ajuste resultante, e ela também fará com que haja mais velocidade na hora de calcular as predições para novas observações.

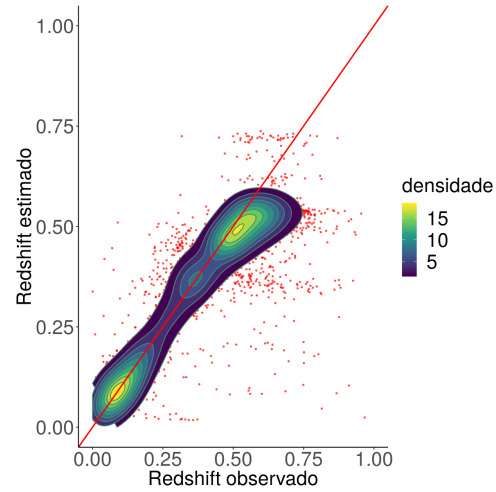
Tabela 2: Importância de cada covariável no modelo combinado ajustado via XGBoost.

Método	**Variação**	**Ganho** $(\times 10^{-2})$
FLEXCODE XGBoost	Mediana	83,7
GPZ	–	15,1
FLEXCODE XGBoost	Moda	0,2
FLEXCODE Floresta	Mediana	0,2
FLEXCODE XGBoost	Média	0,2
FLEXCODE Floresta	Média	0,1
FLEXCODE KNN	Moda	0,1
FLEXCODE Lasso	Média	0,1
FLEXCODE KNN	Mediana	0,1
FLEXCODE KNN	Média	0,1
FLEXCODE Lasso	Moda	0,1
FLEXCODE Lasso	Mediana	0,1
FLEXCODE Floresta	Moda	0,0

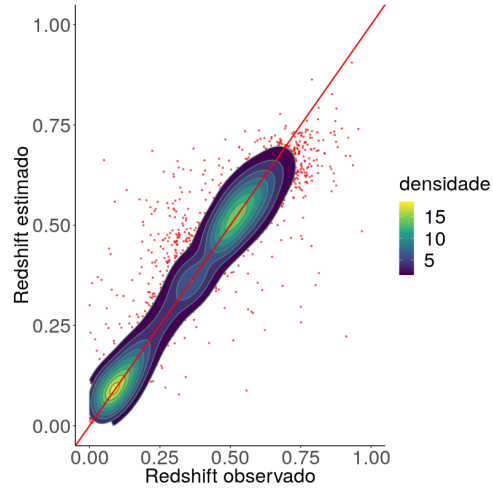
A Tabela 3 mostra o modelo de *stacking* ajustado utilizando-se o lasso. Os únicos métodos selecionados pelo modelo foram o GPZ e o FlexCode XGBoost com a variação mediana e média. Este resultado é coerente com o observado no *stacking* via XGBoost (Tabela 2).



(a) GPZ



(b) FlexCode XGBoost mediana



(c) *Stacking* via lasso

Figura 1: Gráficos de dispersão entre os *redshifts* observados e os *redshifts* estimados pelos melhores métodos individuais e pelo *stacking* via lasso.

A Figura 1 mostra gráficos de dispersão entre os *redshifts* observados e os *redshifts* estimados pelos melhores métodos individuais e pelo *stacking* via lasso. Pode-se notar que a maior parte das observações encontram-se em torno da reta identidade, o que indica que há um bom desempenho preditivo. Contudo, há uma grande quantidade de galáxias *outliers* (sinalizadas em vermelho). Essas são galáxias para as quais os métodos preditivos forneceram resultados ruins. Como estes pontos estão presentes tanto no ajuste do GPZ quanto no XGBoost mediana, que são de natureza bastante diferente, há portanto um indicativo de que as covariáveis possivelmente não são infor-

Tabela 3: Coeficientes do modelo combinado ajustado utilizando-se o lasso.

Método	**Variação**	**Coeficiente** $(\times 10^{-2})$
FLEXCODE XGBoost	Mediana	42,6
GPZ	–	54,0
FLEXCODE XGBoost	Média	2,4

mativas o suficiente para se fazer a predição com grande acurácia para algumas galáxias. Este fato faz com que os métodos de estimação de função densidade condicional, cujos ajustes estão descritos na seção que segue, sejam de grande valia para descrever as incertezas existentes nas predições.

3.1 Combinando estimadores de funções densidades condicionais

A Tabela 4 mostra o risco estimado de cada método de estimação da função densidade condicional. A combinação usando a média ponderada de fato leva a um melhor resultado. A média simples, por outro lado, tem um resultado inferior ao FlexCode-XGBoost. Isso possivelmente se deve ao fato de que, nessa abordagem, todos os modelos individuais contribuem igualmente para a predição combinada.

Entre os modelos individuais, o método que apresenta o menor risco é o FlexCode-XGBoost, enquanto que os piores resultados são dados pelo FlexCode-Lasso.

Tabela 4: Risco estimado de cada estimador da função densidade condicional $f(y|\mathbf{x})$.

Método	**Risco**
Combinação - Média Ponderada	-10,78
FlexCode-XGBoost	-10,72
Combinação - Média Simples	-9,05
FlexCode-Floresta Aleatórias	-8,35
FlexCode-KNN	-5,43
FlexCode-Lasso	-4,18

A Tabela 5 mostra os valores ótimos dos pesos encontrados segundo a Equação 2.1. Note que valores de α_i diferentes de zero estão associados aos métodos de estimação de função densidade que tiveram um bom desempenho preditivo segundo a Tabela 4. Por outro lado, os métodos que obtiveram riscos mais altos tiveram peso nulo na ponderação. Observe que o FlexCode-

XGBoost, o método que apresenta o melhor poder preditivo, tem peso associado muito maior do que os demais.

Tabela 5: Pesos ótimos α para a combinação ponderada das funções densidades condicionais.

	Pesos	**Densidades**
α_1	0,12	Floresta
α_2	0,00	KNN
α_3	0,00	Lasso
α_4	0,88	XGBoost

A Figura 2 mostra as funções densidades estimadas utilizando-se ponderação para quatro galáxias escolhidas aleatoriamente. A linha vertical representa o *redshift* observado. Observa-se que as estivas das funções densidades para cada galáxia têm natureza bastante distinta em cada caso. Em geral, elas são assimétrica e multimodais, o que corrobora que essas funções densidades possuem mais informação que preditores pontuais.

4 CONCLUSÕES

A estimação mais precisa do *redshift* de galáxias permite fazer inferência mais precisa sobre modelos cosmológicos. Assim, ela possibilita identificar quais teorias sobre a evolução do universo são mais adequadas, o que por sua vez é fundamental para prever com maior precisão o comportamento futuro de nosso universo. Neste trabalho, mostramos que combinar métodos já existentes de predição do *redshift* de uma galáxia pode levar a resultados melhores do que considerar apenas algoritmos de aprendizado individuais. Para o caso de regressão (i.e., estimação pontual de y), vimos que o método que levou a predições mais precisas foi o *stacking* via lasso. Tal método utilizou uma combinação linear entre FlexCode-XGBoost mediana, FlexCode-XGBoost média e GPZ. Tal combinação deu mais pesos ao FlexCode-XGBoost mediana, que foi justamente o melhor preditor individual. Ainda que este combinação tenha elevado o poder preditivo dos modelos individuais, vimos que as predições são bastante ruins para algumas galáxias (Figura 1), o que indica que as covariáveis de fato não podem determinar o *redshift* de cada galáxia com alta precisão. Assim, torna-se valioso conseguir estimar bem a função densidade condicional $f(y|\mathbf{x})$. O método que levou a estimativas mais precisas de $f(y|\mathbf{x})$ foi o método que combina linearmente as estimativas individuais com pesos diferentes. Quase todo o peso foi dado ao FlexCode-XGBoost, que demonstrou ter um bom ajuste.

Assim, concluímos que os métodos de *stacking* levaram a um aumento no poder preditivo de cada um dos estimadores base. Em troca, esses procedimentos são computacionalmente mais intensivos na hora de calcular predições, visto que é necessário calcular as predições de cada um dos estimadores base para então combiná-las.

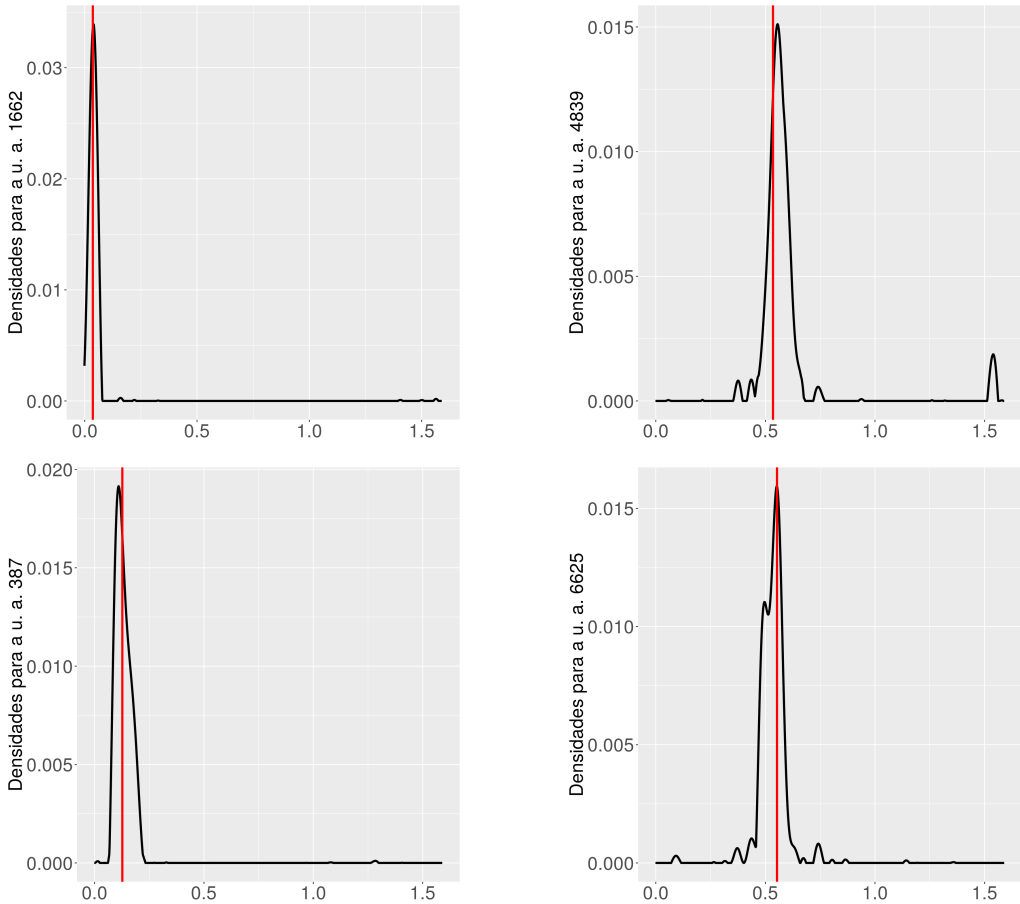


Figura 2: Funções densidades estimadas utilizando-se ponderação para quatro galáxias escolhidas aleatoriamente. A linha vertical representa o *redshift* observado em cada caso.

Em trabalhos futuros pretendemos propor outras formas de combinar as funções densidades obtidas que permitam que o peso recebido por cada componente do *stacking* varie segundo o valor de cada covariável. Outro ponto a ser estudado é que os métodos existentes possuem a suposição de que as observações são independentes e identicamente distribuídas. Contudo, a suposição de mesma distribuição não é razoável para esse problema [14], uma vez que as galáxias que são rotuladas geralmente são mais próximas, por conta da espectroscopia funcionar melhor para esses casos. Como se tem interesse em fazer predições para galáxias não rotuladas, o comportamento da sua medição induz um viés de seleção no conjunto rotulado, de modo que o risco pode ser subestimado mesmo utilizando o *data splitting*. Assim, uma extensão deste trabalho consiste em levar esse viés de seleção em conta na hora de combinar as predições. Finalmente, também iremos combinar covariáveis fotométricas com informações sobre a morfologia de cada galáxia [10] de modo a obter estimativas mais precisas para o *redshift* de cada galáxia.

AGRADECIMENTOS

Os autores agradecem a Luís Ernesto Salasar e Gustavo Henrique de Araujo Pereira pelas valiosas sugestões feitas a esse trabalho.

Este projeto contou com o auxílio da FAPESP (2017/03363-8 e 2019/11321-9) e do CNPq (306943/2017-4).

ABSTRACT. A key problem in cosmology is the estimation of the redshifts of galaxies using photometric data. For instance, the *Sloan Digital Sky Survey* (SDSS) has already collected photometric data of about one billion objects, and it is necessary to estimate their redshifts. Typically, this is done by using supervised learning methods. In this work we show that existing redshift prediction methods can be combined in order to obtain more accurate predictions. We tackle this problem under two perspectives: (i) estimation of the regression function of the redshift y on the photometric $\mathbf{x}$, $\mathbf{E}[Y|\mathbf{x}]$, and (ii) estimation of the conditional density $f(y|\mathbf{x})$. We apply the proposed techniques to data from the *Sloan Digital Sky Survey*, and show that the combined predictions are indeed more accurate.

Keywords: machine learning, stacking, conditional densities, cosmology.

REFERÊNCIAS

[1] I.A. Almosallam, M.J. Jarvis & S.J. Roberts. GPZ: non-stationary sparse Gaussian processes for heteroscedastic uncertainty estimation in photometric redshifts. *Monthly Notices of the Royal Astronomical Society*, **462**(1) (2016), 726–739.

[2] G.B. Brammer, P.G. van Dokkum & P. Coppi. EAZY: a fast, public photometric redshift code. *The Astrophysical Journal*, **686**(2) (2008), 1503.

[3] L. Breiman. Random forests. *Machine learning*, **45**(1) (2001), 5–32.

[4] T. Chen & C. Guestrin. XGBoost: A scalable tree boosting system. In "Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining". ACM (2016), pp. 785–794.

[5] V. Coscrato, M.H.d.A. Inácio & R. Izbicki. The NN-Stacking: Feature weighted linear stacking through neural networks. *arXiv preprint arXiv:1906.09735*, (2019).

[6] I. Csabai, T. Budavari, A.J. Connolly, A.S. Szalay, Z. Gyory, N. Benitez, J. Annis, J. Brinkmann, D. Eisenstein, M. Fukugita *et al.* The application of photometric redshifts to the SDSS early data release. *The Astronomical Journal*, **125**(2) (2003), 580.

[7] N. Dalmasso, T. Pospisil, A.B. Lee, R. Izbicki, P.E. Freeman & A.I. Malz. Conditional Density Estimation Tools in Python and R with Applications to Photometric Redshifts and Likelihood-Free Cosmological Inference. *arXiv preprint arXiv:1908.11523*, (2019).

[8] P.E. Freeman, R. Izbicki & A.B. Lee. A unified framework for constructing, tuning and assessing photometric redshift density estimates in a selection bias setting. *Monthly Notices of the Royal Astronomical Society*, **468**(4) (2017), 4556–4565.

[9] T. Hastie, R. Tibshirani & J. Friedman. "The Elements of Statistical Learning". Springer Series in Statistics. Springer New York Inc., New York, NY, USA (2001).

[10] P. Ianishi & R. Izbicki. Classificação Morfológica de Galáxias em Conjuntos de Dados Desbalanceados. *TEMA (São Carlos)*, **18**(1) (2017), 155–172.

[11] R. Izbicki & T.M. dos Santos. Machine Learning sob a ótica estatística (2019). URL `http://www.rizbicki.ufscar.br/sml.pdf`.

[12] R. Izbicki & A.B. Lee. Nonparametric conditional density estimation in a high-dimensional regression setting. *Journal of Computational and Graphical Statistics*, **25**(4) (2016), 1297–1316.

[13] R. Izbicki & A.B. Lee. Converting high-dimensional regression to high-dimensional conditional density estimation. *Electronic Journal of Statistics*, **11**(2) (2017), 2800–2831.

[14] R. Izbicki, A.B. Lee & P.E. Freeman. Photo-z estimation: An example of nonparametric conditional density estimation under selection bias. *The Annals of Applied Statistics*, **11**(2) (2017), 698–724.

[15] R. Izbicki & T. Pospisil. rizbicki/FlexCode v5.9-beta.3 (2019). doi:10.5281/zenodo.3366065. URL `https://doi.org/10.5281/zenodo.3366065`.

[16] R. Izbicki, G.T. Shimizu & R.B. Stern. Distribution-free conditional predictive bands using density estimators. *arXiv preprint arXiv:1910.05575*, (2019).

[17] G. James, D. Witten, T. Hastie & R. Tibshirani. "An Introduction to Statistical Learning: with Applications in R". Springer (2013). URL `https://faculty.marshall.usc.edu/gareth-james/ISL/`.

[18] S. Schmidt, A. Malz, J. Soo, I. Almosallam, M. Brescia, S. Cavuoti, J. Cohen-Tanugi, A. Connolly, J. DeRose, P. Freeman *et al.* Evaluation of probabilistic photometric redshift estimation approaches for LSST. *arXiv preprint arXiv:2001.03621*, (2020).

[19] E.S. Sheldon, C.E. Cunha, R. Mandelbaum, J. Brinkmann & B.A. Weaver. Photometric redshift probability distributions for galaxies in the SDSS DR8. *The Astrophysical Journal Supplement Series*, **201**(2) (2012), 32.

[20] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, **58**(1) (1996), 267–288.

[21] B.A. Turlach & A. Weingessel. quadprog: Functions to solve quadratic programming problems. *CRAN-Package quadprog*, (2007).

[22] B.A. Turlach & A. Weingessel. "quadprog: Functions to solve Quadratic Programming Problems." (2013). URL `https://CRAN.R-project.org/package=quadprog`. R package version 1.5-5.

[23] Y. Wadadekar. Estimating photometric redshifts using support vector machines. *Publications of the Astronomical Society of the Pacific*, **117**(827) (2004), 79.

[24] D. Wittman. What lies beneath: Using p(z) to reduce systematic photometric redshift errors. *The Astrophysical Journal Letters*, **700**(2) (2009), L174.

[25] C. Yèche, P. Petitjean, J. Rich, E. Aubourg, J.C. Hamilton, J.M. Le Goff, I. Paris, S. Peirani, C. Pichon, E. Rollinde *et al.* Artificial neural networks for quasar selection and photometric redshift determination. *Astronomy & Astrophysics*, **523** (2010), A14.

[26] D.G. York, J. Adelman, J.E. Anderson Jr, S.F. Anderson, J. Annis, N.A. Bahcall, J. Bakken, R. Barkhouser, S. Bastian, E. Berman *et al.* The sloan digital sky survey: Technical summary. *The Astronomical Journal*, **120**(3) (2000), 1579.

[27] M.L. Zhang & Z.H. Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition*, **40**(7) (2007), 2038–2048.

[28] Z.H. Zhou. "Ensemble Methods: Foundations and Algorithms". Chapman & Hall/CRC, 1st ed. (2012).