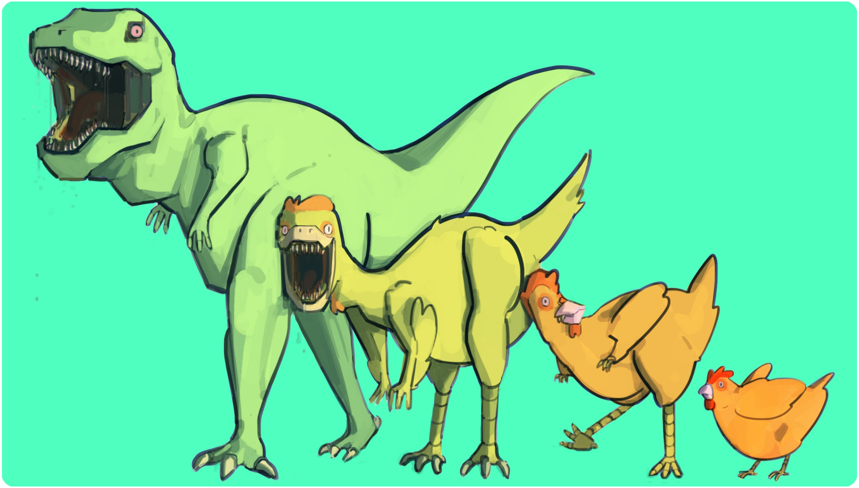
Dataset Shift: Covariate Shift



Dataset shift

No paradigma de aprendizado supervisionado temos um vetor aleatório $(X_1, X_2, \cdots, X_n, Y)$ e uma relação do tipo

$$Y \sim f(X_1, X_2, \cdots, X_n) + \varepsilon.$$

O objetivo é estimar a função f a partir de uma amostra de observações do vetor aleatório.

Mas o que acontece se a distribuição do vetor aleatório muda?

- **Concept Shift**: mudança em f
- **Covariate Shift**: mudança nas distribuições de X_i

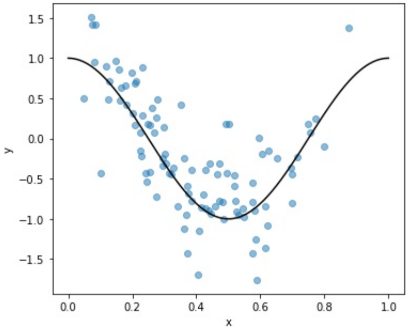
Podemos garantir que a estimação de f continua fazendo um bom trabalho?

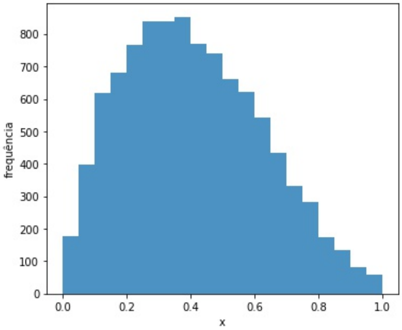
Exemplo numérico para o Covariate Shift

$$(X, Y) \sim \mathcal{D}^{\text{old}}$$

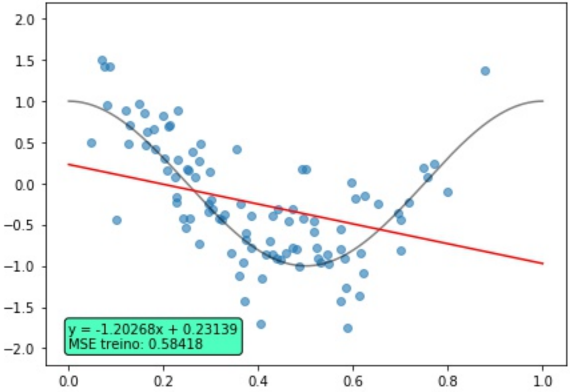
$$X \sim \mathrm{Beta}(2, 3) + \xi$$

$$Y \sim \cos(2\pi X) + \mathcal{N}(0, 0.25)$$





Exemplo numérico para o Covariate Shift

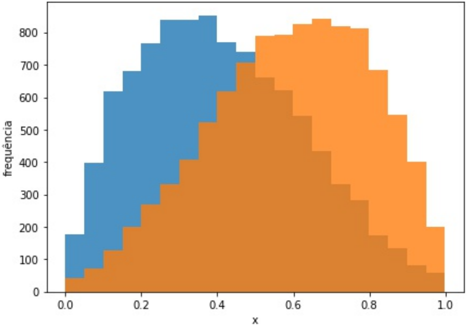


Exemplo numérico para o Covariate Shift

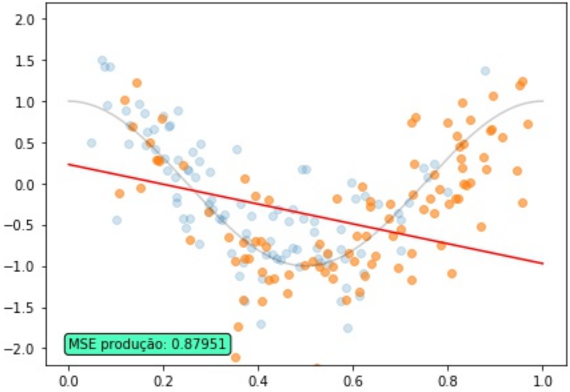
$$(X, Y) \sim \mathcal{D}^{\text{new}}$$

$$X \sim \mathrm{Beta}(3, 2) + \xi$$

$$Y \sim \cos(2\pi X) + \mathcal{N}(0, 0.25)$$



Exemplo numérico para o Covariate Shift



Identificando Covariate Shift

Precisamos identificar se a distribuição das covariáveis no treinamento é a mesma das covariáveis em produção olhando apenas para amostras.

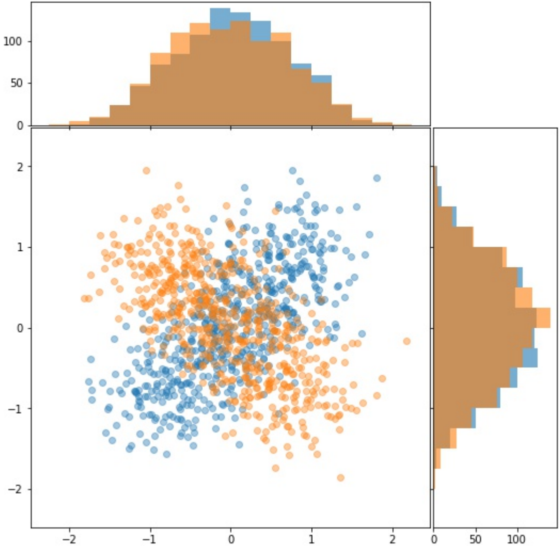
Técnicas clássicas para problemas univariados (componente por componente)

- *Teste de hipótese para comparação de médias*
- *Teste Kolmogorov-Smirnov*
- *Divergência de Kullback-leibler*
- *Área da interseção de histogramas*
- *QQ-plot*

Se sei qual componente do meu vetor aleatório está problemática posso entender o motivo.

Sabendo a variável com shift podemos usar alguma técnica que tenta **aproximar as distribuições** (drop na coluna ou undersample).

Identificando Covariate Shift



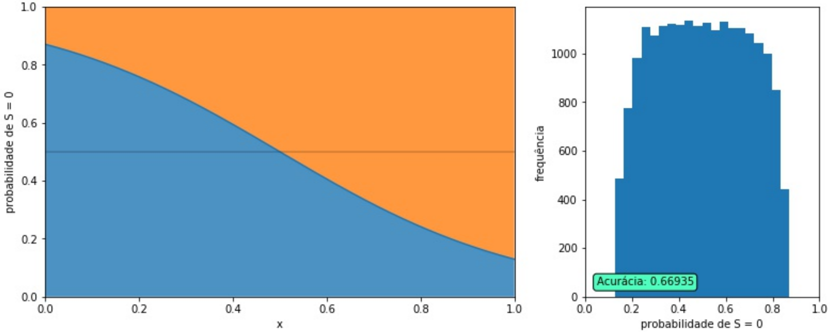
Identificando Covariate Shift

Ideia: treinar um classificador binário que recebe o vetor com todas as covariáveis e tenta prever ser é do treino ($S = 0$) ou da produção ($S = 1$).

Se o classificador tem uma acurácia alta, quer dizer que a distribuição das variáveis explicativas no treino e em produção são diferentes.

x	y	S
0.0679785	0.456452	0
0.266631	-0.517171	0
0.509772	-0.822024	0
⋮	⋮	⋮
0.737744	?	1
0.969182	?	1
0.398244	?	1
0.979168	?	1

Identificando Covariate Shift



Minimizando o risco empírico

Tomemos como exemplo a regressão linear, como aplicamos no exemplo numérico. Nela estamos interessados em encontrar uma relação do tipo:

$$h_{a,\,b}(x) = ax + b$$

reduzindo o erro quadrático médio nos dados de treinamento ($\mathcal{D}^{\mathrm{old}}$).

Nossa idealização é minimizar o **MSE na distribuição**

$$\mathrm{MSE}(h_{a,\,b}, \mathcal{D}^{\mathrm{old}}) = \mathbb{E}_{(x,y)\sim\mathcal{D}^{\mathrm{old}}} \left[(h_{a,\,b}(x) - y)^2 \right].$$

mas não conhecemos a distribuição.

A alternativa é minimizar o **erro empírico** (no nosso conjunto de dados de treino)

$$\mathrm{MSE}(h_{a,\,b}, \mathrm{sample\ old}) = \frac{1}{N} \sum_{i=1}^{N} (h_{a,\,b}(x_i) - y_i)^2 .$$

Unsupervised Domain Adaptation

Se quisermos um bom resultado em $\mathcal{D}^{\text{new}}$, precisamos achar a e b que minimizem o **MSE na distribuição nova**, isto é

$$
\begin{aligned}
\mathrm{MSE}(h_{a,\,b}, \mathcal{D}^{\text{new}}) &= \mathbb{E}_{(x,y)\sim\mathcal{D}^{\text{new}}}\left[\left(h_{a,\,b}(x) - y\right)^2\right] = \\
&= \sum_{(x,y)} \mathcal{D}^{\text{new}}(x,y)\left(h_{a,\,b}(x) - y\right)^2 = \\
&= \sum_{(x,y)} \mathcal{D}^{\text{new}}(x,y)\frac{\mathcal{D}^{\text{old}}(x,y)}{\mathcal{D}^{\text{old}}(x,y)}\left(h_{a,\,b}(x) - y\right)^2 = \\
&= \sum_{(x,y)} \mathcal{D}^{\text{old}}(x,y)\left(\frac{\mathcal{D}^{\text{new}}(x,y)}{\mathcal{D}^{\text{old}}(x,y)}\left(h_{a,\,b}(x) - y\right)^2\right) = \\
&= \mathbb{E}_{(x,y)\sim\mathcal{D}^{\text{old}}}\left[\frac{\mathcal{D}^{\text{new}}(x,y)}{\mathcal{D}^{\text{old}}(x,y)}\left(h_{a,\,b}(x) - y\right)^2\right].
\end{aligned}
$$

Unsupervised Domain Adaptation

Portanto, para um bom resultado nos dados novos, devemos escolher a e b de forma que minimizamos uma **variação do MSE**:

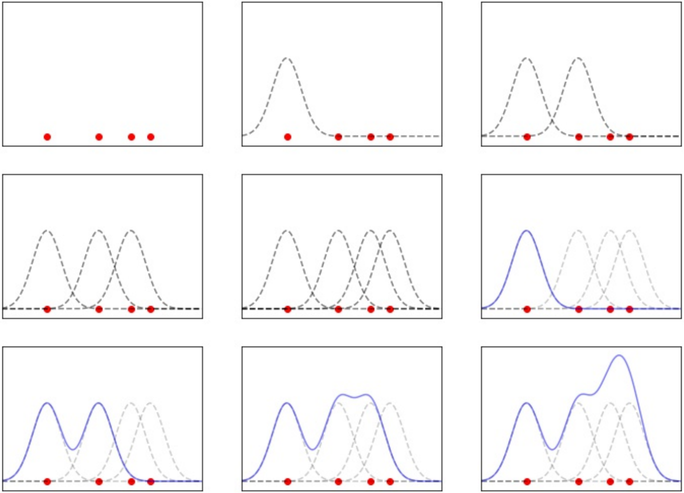
$$\overline{\mathrm{MSE}}(h_{a,\,b}, \text{sample old}) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mathcal{D}^{\mathrm{new}}(x_i, y_i)}{\mathcal{D}^{\mathrm{old}}(x_i, y_i)}\left(h_{a,\,b}(x_i) - y_i\right)^2\right).$$

Mas como podemos calcular para os nossos exemplos o valor de

$$\frac{\mathcal{D}^{\mathrm{new}}(x_i, y_i)}{\mathcal{D}^{\mathrm{old}}(x_i, y_i)}\ ?$$

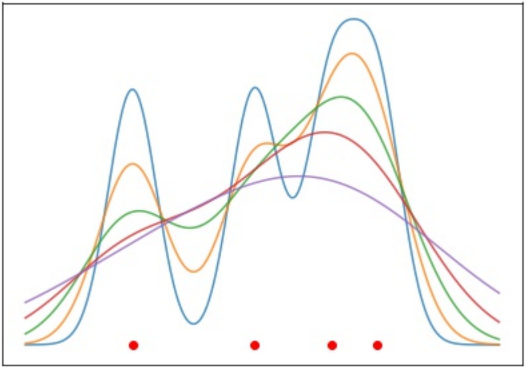
Kernel Density Estimation

$$\widehat{f}(x) = \frac{1}{N}\sum_{i=1}^{N} K(x - x_i)$$



Kernel Density Estimation

$$K(x - x_i) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{x - x_i}{\sigma}\right)^2}$$



Unsupervised Domain Adaptation

Ideia: Todos os exemplos são tirados de alguma distribuição base $\mathcal{D}^{\text{base}}$. Alguns dados são escolhidos para ir para a distribuição antiga $\mathcal{D}^{\text{old}}$ e outros para a distribuição nova $\mathcal{D}^{\text{new}}$.

Decidimos qual exemplo vai pra qual a partir de uma variável de seleção S. A escolha de S depende apenas das covariáveis.

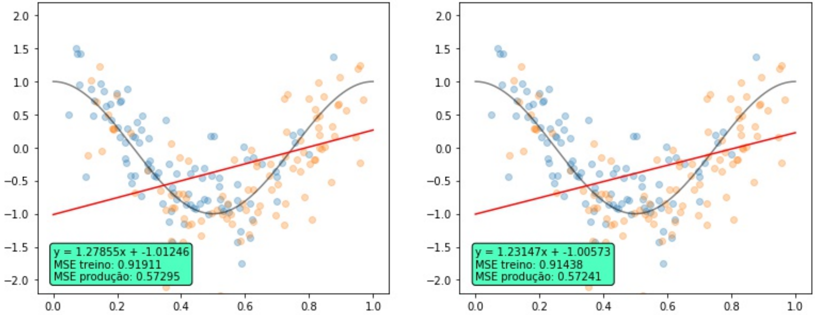
$S = 0 \to$ distribuição antiga e $S = 1 \to$ distribuição nova.

$$\mathcal{D}^{\text{old}}(x, y) = \mathcal{D}^{\text{base}}(x, y)\,\mathbb{P}(S = 0|x),$$
$$\mathcal{D}^{\text{new}}(x, y) = \mathcal{D}^{\text{base}}(x, y)\,\mathbb{P}(S = 1|x).$$

Portanto

$$\frac{\mathcal{D}^{\text{new}}(x_i, y_i)}{\mathcal{D}^{\text{old}}(x_i, y_i)} = \frac{\mathbb{P}(S = 1|x_i)}{\mathbb{P}(S = 0|x_i)} = \frac{1 - \mathbb{P}(S = 0|x_i)}{\mathbb{P}(S = 0|x_i)} = \frac{1}{\mathbb{P}(S = 0|x_i)} - 1.$$

```
lr_weight = LinearRegression().fit(X_past, Y_past,
sample_weight =1/log.predict_proba(X_past)[:,0]-1)
```




```
lr_new = LinearRegression().fit(X_new, Y_new)
```

Referências

- **Unsupervised Domain Adaptation: capítulo 8 - Bias and Fairness do livro A Course in Machine Learning do Hal Daumé III (ciml.info)**
- Principles of Risk Minimization for Learning Theory (Vapnik): shorturl.at/kPQSV
- Introdução ao Dataset Shift (Analytics Vidhya): shorturl.at/nsQ03
- Livro Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation (seções 1.1 - 1.3.4 e 2.1)
- Livro Dataset Shift in Machine Learning (seções 1.1 - 1.4)
- Intro to Kernel Density Estimation: https://youtu.be/x5zLaWT5KPs
- Interseção de histogramas: shorturl.at/ruLQW
- Subsampling a training set to match a test set : shorturl.at/xyzOZ
- Using QQ-plot to compare two samples: shorturl.at/eGHW0