Aula 1

Prof. Davi Michel Valladão

PUC-Rio
IND2076
Eng. Industrial
16/03/2023

Objetivos. 1) Revisar conceitos e ferramentas chave aos tópicos de convexidade e otimização. 2) Mostrar equivalência entre Regressão Lasso e Otimização Robusta.

1 Revisão de Otimização

O objetivo da revisão é dar uma base comum a todos na prática de modelagem de pesquisa operacional. Vamos primeiro estudar um case prático como motivação:

1.1 Motivação Prática

Nosso objetivo nessa parte é aplicar o modelo de regressão linear para previsão da quantidade horária de bicicletas alugadas com o dataset *Bike Sharing Dataset*, disponível em `https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset`. Será utilizada a linguagem de programação Julia.

O conjunto de dados disponível no arquivo *hour.csv* possui 17.389 observações (cada uma representa 1 hora de um dia), 15 variáveis explicativas e 1 variável representando a quantidade de alugueis de bicicleta no período.

Podemos resolver essa tarefa computacionalmente de 3 maneiras:

1. Pacote GLM
2. Solução "analítica" dos mínimos quadrados
3. Solução do problema de otimização de mínimo erro quadrático

A seguir está o código (simplificado) relacionado a cada maneira:

GLM

```julia
# import package GLM
using GLM
```

```
# apply linear model
glm_coef = lm(X, y)
```

Solução analítica

```
# least squares formula
form_coef = (X'X)^(-1)*X*y
```

Otimização

```
# import packages JuMP and solver
using JuMP
using Ipopt

# get matrix size
n, p = size(X)

# initialize model
m = Model(Ipopt.Optimizer)
# define decision variables
@variable(m, b[1:p])
# define objective function (to be minimized)
@objective(m, Min, sum((X*b - y).^2))

# run solve step
optimize!(m)

# get solution
opt_coef = value.(b)
```

Os 3 métodos chegam, a príncipio, no mesmo resultado de coeficientes para a regressão linear.

Vamos agora fazer uma pequena modificação: reduzir o número de observações (linhas das matrizes) disponíveis:

```
n = 1000
X = X[:n, :]
y = y[:n]
```

Em nosso experimento, o segundo método não funcionou. O motivo disso é que a etapa de inversão da matriz $X'X$ não é possível para um número menor de observações. A primeira técnica, usando o pacote *GLM* ainda funciona e chega no mesmo resultado que a otimização via *PuLP*, pois a biblioteca possui métodos mais "inteligentes" para buscar uma solução.

Vamos agora fazer outra modificação: queremos resolver a regressão minimizando o erro absoluto, não o erro quadrático. Essa modificação é muito útil para casos em que há presença de *outliers* fortes na amostra de treino (referência: `https://towardsdatascience.com/comparing-robustness-of-mae-mse-and-rmse-6d69da870828`). Para essa tarefa, entre as opções anteriores, nos resta seguir na modelagem do problema de otimização com o pacote *JuMP*:

```julia
# import packages JuMP and solver
using JuMP
using Ipopt

# get matrix size
n, p = size(X)

# initialize model
m = Model(Ipopt.Optimizer)
# define decision variables
@variable(m, b[1:p])
@variable(m, err[1:n])

# constraint: err
@constraint(m, err .>= X*b - y)
@constraint(m, err .>= -(X*b - y))

# define objective function (to be minimized)
@objective(m, Min, err)

# run solve step
optimize!(m)

# get solution
opt_coef = value.(b)
```

Importante notar que a função módulo, que não é uma função linear, pôde ser implementada dentro do problema utilizando o **epígrafo** da função, por meio da criação de uma variável auxiliar e algumas restrições. Executando o código e avaliando os parâmetros gerados, temos sucesso em nossa tarefa.

Nessa modelagem, não há variáveis ou expressões não lineares, ou seja, estamos lidando com um problema de **otimização linear**, não mais **otimização convexa**, como era o caso do erro quadrático. Para esse caso, solvers desenvolvidos para lidar com esse tipo de problema poderão chegar a uma solução com mais eficiência e precisão. Logo, faz sentido que mudemos o solver utilizado.

Espero que os experimentos mostrados sirvam como motivação para que nos aprofundemos em alguns conceitos:

1. **Ferramental de otimização** para que possamos resolver uma gama de problemas com facilidade.
2. **Conceitos de otimização** para que possamos identificar corretamente o tipo e as características de problemas de otimização diversos e tomar as decisões certas ao tratá-los.

1.2 Otimização convexa

Vamos, a seguir, trazer definições importantes para otimização convexa. Recomenda-se o uso do livro **Convex Optimization [Boyd and Wandenberghe]**, que pode ser estudado ainda com o apoio das aulas gravadas do professor Stephen Boyd em Stanford (`https://www.youtube.com/playlist?list=PL3940DD956CDF0622`, até a parte de otimização dual).

Definições importantes:

1. **Conjunto convexo**: Um conjunto $\mathbf{C}$ é convexo se, e somente se:
$$(1-t)x_1 + tx_2 \in \mathbf{C} \ \forall \ x_1, x_2 \in \mathbf{C}, t \in [0,1]$$
Ou seja, um conjunto é convexo se qualquer **combinação afim** (`https://en.wikipedia.org/wiki/Affine_combination`) de quaisquer dois elementos do conjunto também pertença a ele.
2. A partir da definição acima, podemos listar algumas das **operações que mantém convexidade em conjuntos** (Os exemplos abaixo consideram $\mathbf{C}, \mathbf{D} \subseteq \mathbf{R^n}, \beta \in \mathbf{R^n}, \alpha \in \mathbf{R}, A \in \mathbf{R^{mXn}}, b \in \mathbf{R}^m$:
– Intersecção: $\mathbf{C} \cap \mathbf{D}$
– Transformação afim: $A\mathbf{C} + b$ (abaixo exemplos de transformações afim)
– Translação: $\mathbf{C} + \beta$
– Produto escalar: $\alpha\mathbf{C}$
– Soma: $\mathbf{C} + \mathbf{D}$
3. Função convexa: Seja $X \in \mathbf{R^n}$ um conjunto convexo e $f : X \to \mathbf{R}$, f é uma função convexa se, e somente se:
$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2) \ \forall \ x_1, x_2 \in X, t \in [0,1]$$
Em outras palavras, em uma função convexa, se traçarmos o segmento de reta que liga dois pontos da função, todos os pontos do segmento tem valor maior (ou estão "acima") do que a função avaliada no elemento x correspondente. (Função côncava pode ser definida mudando o sinal da desigualdade.)
4. Funções duplamente deriváveis, ou seja, que a segunda derivada existe, e que tenha segunda derivada não negativas, são convexas.
5. Da mesma forma que anteriormente, podemos estudar operações que preservam convexidade de funções:
– **Média ponderada**: $\sum_{i=1}^{n} \alpha_i f_i, \sum_{i=1}^{n} \alpha_i = 1$
– **Composição linear**: Seja $f : X \to \mathbf{R}$ e $h(x) = Ax + b$, $f(h(x))$
– **Máximo** de funções convexas
– **Composição de funções**, a depender de algumas características. Ao invés de decorar quais composições resultam em convexidade, vale entender como demonstrar (assumindo dupla diferenciabilidade nas funções estudadas). Seja $f : X \to \mathbf{R}$, $g : Y \to \mathbf{R}$:

$$h(x) = f(g(x))$$

A partir da regra da cadeia:

$$h'(x) = f'(g(x))g'(x)$$

$$h''(x) = f''(g(x))g'(x)^2 + f'(g(x))g''(x)$$

$$h''(x) \geq 0 \rightarrow f''(g(x)) \geq 0,\ f'(g(x))g''(x) \geq 0$$

Ou seja, precisamos que, das duas, uma:
– f **convexa e não decrescente,** g **convexa**
– f **convexa e não crescente,** g **côncava**

Na prática, podemos utilizar esses conceitos para entender, a partir da função objetivo e restrições do nosso problema de otimização, em qual categoria ele está (linear, convexo, não convexo, entre outros) e tomar decisões corretas sobre como modelá-lo da melhor forma e selecionar um solver adequado.

2 Regressão Lasso como Otimização Robusta

Antes de apresentar o problema em questão, precisamos apresentar algumas definições acerca do conceito de **norma**.

2.1 Norma ($\|x\|_p$)

Considerando x um vetor em $\mathbf{R^n}$, a função norma **p** pode ser definida como:

$$\|x\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$$

onde $|x_i|$ é o valor absoluto do escalar x_i.

As normas que mais vamos nos deparar em problemas são as normas 1, 2 e ∞. Essa última pode não ser tão claramente interpretável a partir da fórmula apresentada, mas pode ser avaliada como a função:

$$\|x\|_\infty = max_i x_i$$

Uma intuição por trás dessa definição é que, a medida que o expoente na definição inicial se aproxima de infinito, o termo de maior valor absoluto no vetor resulta em um potência de valor muito alto, deixando os outros termos irrelevantes quando se eleva a soma ao inverso da potência p.

Algumas propriedades importantes da norma:

1. $\|x\|_p = 0 \rightarrow x = 0$
2. $\|\alpha x\|_p = \alpha \|x\|_p$
3. $\|x_1 + x_2\|_p \leq \|x_1\|_p + \|x_2\|_p$ (Desigualdade triangular)

2.2 Norma dual

Definimos $G(x) = \|x\|_p$. A norma dual pode ser escrita da seguinte forma:

$$G_*(x) = max_{\|v\|_p=1}\ v'x$$

/ Para as normas mais utilizadas:

1. A norma dual da norma 1 é a norma infinita
2. A norma dual da norma infinita é a norma 1
3. A norma dual da norma 2 é a própria norma 2
4. A norma dual da norma dual de norma p é a própria norma p

Por fim, podemos definir uma função $h_p : \mathbf{R^n} \to \mathbf{R^n}$:

$$h_p(x) = v,\ v^T x = \|x\|_p$$

2.3 Regressão Lasso

Nós queremos mostrar a equivalência entre regressão Lasso, que possui a seguinte forma:

$$min_\beta\ \|y - X\beta\|_p + \lambda\|\beta\|_q$$

e regressão robusta. Vamos partir do formato da regressão robustam que é a busca pelo β que minimiza o caso com uma pertubação Δ que maximiza o erro absoluto. Abrindo o produto dentro do módulo:

$$min_\beta\ \{max_{\Delta \in \mathbf{U}}\|y - (X + \Delta)\beta\|_p\}$$

Abrindo o produto dentro do módulo:

$$min_\beta\ \{max_{\Delta \in \mathbf{U}}\|(y - X\beta) - \Delta\beta\|_p\}$$

e substituindo o termo $(y - X\beta)$ por $\varepsilon(\beta)$:

$$min_\beta\ \{max_{\Delta \in \mathbf{U}}\|\varepsilon(\beta) - \Delta\beta\|_p\}$$

Aqui, vamos relembrar e aplicar uma propriedade de norma que foi apresentada anteriormente, a desigualdade triangular:

$$\|\varepsilon(\beta) - \Delta\beta\|_p \leq \|\varepsilon(\beta)\|_p + \|\Delta\beta\|_p$$

Ou seja, a norma da soma dos dois fatores de erro está limitada pela soma das normas dos fatores. Essa limitação vem da direção do vetor $\Delta\beta$, ou seja, o caso que maximiza o lado direito da equação e iguala os dois é o que possui os dois vetores na mesma direção:

$$\frac{\varepsilon(\beta)}{\|\varepsilon(\beta)\|} = \frac{\Delta\beta}{\|\Delta\beta\|}$$

Para encontrar um Δ, adicionaremos o conceito de *Norma Induzida*. Para um λ fixo:

$$InducedNorm(p, r) = \{\Delta \mid max_\beta \frac{\|\Delta\beta\|_p}{\|\beta\|_r} \leq \lambda\}$$

Utilizando essa função, podemos encontrar um Δ e garantir que $\|\Delta\beta\|_p \leq \lambda\|\beta\|_p$. Vamos colocar as desigualdades em sequência para observar o limite superior do nosso erro (que o problema busca maximizar em Δ):

$$\|\varepsilon(\beta) - \Delta\beta\|_p \leq \|\varepsilon(\beta)\|_p + \|\Delta\beta\|_p \leq \|\varepsilon(\beta)\|_p + \lambda\|\beta\|_p$$

Finalmente, para maximizar o lado esquerdo da desigualdade, precisamos de um Δ que:

1. $\Delta\beta$ tenha a mesma direção que $\varepsilon(\beta)$
2. $\|\Delta\beta\|_p = \lambda\|\beta\|_p$

Aqui, vamos resgatar a função $h_p(x) = \{v,\ v^T x = \|x\|_p\}$. Com ela, podemos encontrar nosso valor de Δ que maximiza o erro para um β:

$$\Delta^* = \lambda \frac{\epsilon(\beta)}{\|\epsilon(\beta)\|} h_p(\beta)$$

Repare que esse delta satisfaz as duas condições, fazendo com que o erro seja máximo, pois:

$\Delta^*\beta = \lambda \frac{\epsilon(\beta)}{\|\epsilon(\beta)\|} h_p(\beta)\beta = \lambda \frac{\epsilon(\beta)}{\|\epsilon(\beta)\|} \|\beta\|_p$

Esse produto: 1) possui a mesma direção de $\varepsilon(\beta)$ (todas os outros fatores são escalares) e 2) possui norma p igual a $\lambda\|\beta\|_p$, pois o termo fracionário tem norma 1. Logo:

$$max_{\Delta \in \mathbf{U}} \|\varepsilon(\beta) - \Delta\beta\|_p \ = \|\varepsilon(\beta) - \Delta^*\beta\|_p = \|\varepsilon(\beta)\|_p + \lambda\|\beta\|_p$$

Finalmente, partindo do problema de otimização robusta:

$$min_\beta \ \{max_{\Delta \in \mathbf{U}} \|\varepsilon(\beta) - \Delta\beta\|_p\} = min_\beta \ \|\varepsilon(\beta)\|_p + \lambda\|\beta\|_p$$

nós chegamos na Regressão Lasso!

Isso significa que a Regressão Lasso é uma técnica de robustez, que busca parâmetros robustos a perturbações na variáveis explicativas, e não necessariamente uma regularização que busca zerar parâmetros relacionados a variáveis pouco importantes, como normalmente é usado. Futuramente, estudaremos mais a fundo por que essa consequência acaba acontecendo na prática também.