Projeto Final - Algoritmos e Incerteza - Algoritmos Projection-Free

Guilherme Bodin

Sumário

Algoritmos Projection-Free

O que é uma projeção?

- Projeção em nosso contexto é projeção de um ponto em um conjunto convexo.
- A projeção de um ponto $y \in \mathbb{R}^n$ em um conjunto convexo $\mathcal{C} \subset \mathbb{R}^n$ é basicamente o ponto mais próximo de y em $\mathcal{C}$ no sentindo de norma euclidiana.

$$P_{\mathcal{C}}(y) \triangleq \arg\min_{x \in \mathcal{C}} ||x - y||$$

O que é uma projeção?

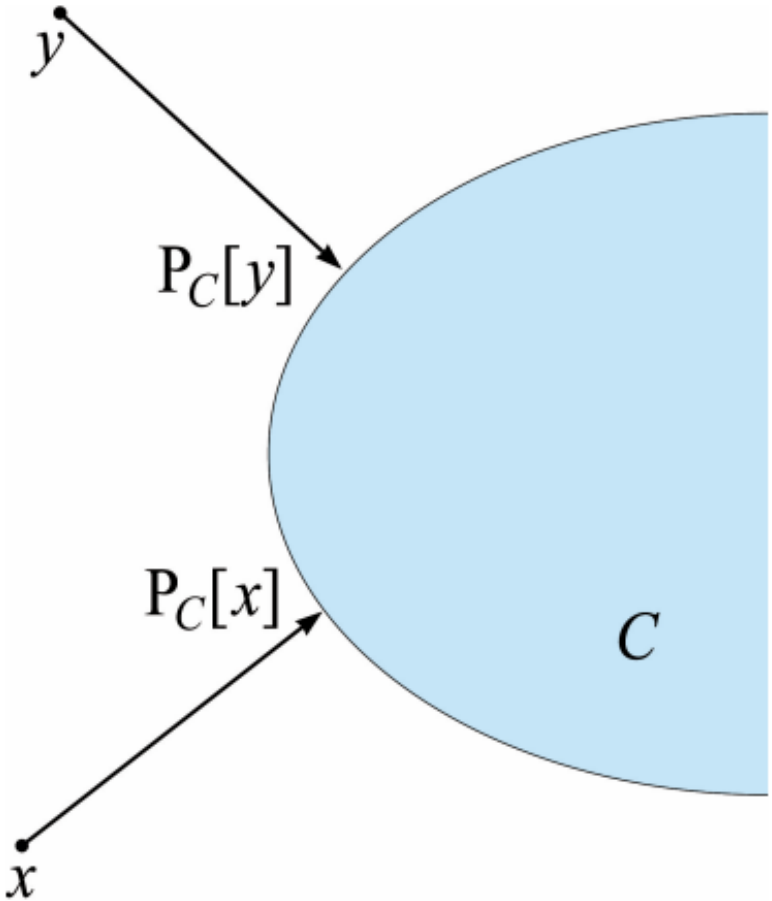

Figura: Exemplo de lecture notes da Angelia Nedić

Onde usamos projeções?

- Gradient Descent!

Algorithm 2 Basic gradient descent

1: Input: f, T, initial point $\mathbf{x}_1 \in \mathcal{K}$, sequence of step sizes $\{\eta_t\}$
2: **for** $t = 1$ to T **do**
3: Let $\mathbf{y}_{t+1} = \mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)$, $\mathbf{x}_{t+1} = \Pi_{\mathcal{K}}(\mathbf{y}_{t+1})$
4: **end for**
5: **return** $\mathbf{x}_{T+1}$

Figura: Algoritmo Gradient Descent do livro do Hazan

- Método de Newton
- Outros algoritmos podem usar outras versões de projeção (ninguém disse que o ponto mais próximo precisa ser no sentido da norma euclidiana)

Projeções são caras?

- Depende do problema. Vou tentar responder daqui a pouco.

Projection-Free

- Frank-Wolfe (Conditional Gradient)

Algorithm 22 Conditional gradient

Input: step sizes $\{\eta_t \in (0,1),\ t \in [T]\}$, initial point $\mathbf{x}_1 \in \mathcal{K}$.
for $t = 1$ to T **do**
  $\mathbf{v}_t \leftarrow \arg\min_{\mathbf{x}\in\mathcal{K}} \left\{\mathbf{x}^\top \nabla f(\mathbf{x}_t)\right\}$.
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \eta_t(\mathbf{v}_t - \mathbf{x}_t)$.
end for

Figura: Algoritmo Gradient Descent do livro do Hazan

Revisão de álgebra

Valores singulares de uma matriz

Seja a matriz $X \in \mathbb{R}^{n \times m}$. O número não negativo $\sigma \in \mathbb{R}_+$ é dito valor singular da matriz X se existirem dois vetores $u \in \mathbb{R}^n$ e $v \in \mathbb{R}^m$ tal que:

$$X^T u = \sigma v \quad , \quad Xv = \sigma u$$

Os vetores u e v são chamados vetor singular à esquerda e vetor singular à direita respectivamente.

Decomposição singular de uma matriz (SVD)

A matriz X pode ser escrita como

$$X = U\Sigma V^T \quad , \quad U \in \mathbb{R}^{n\times\rho} \quad , \quad V \in \mathbb{R}^{\rho\times m}$$

onde $\rho = \min\{n, m\}$, U é a base ortogonal dos vetores singulares à esquerda, V é a base ortogonal dos vetores singulares à direita, e Σ é a matriz diagonal com os respectivos valores singulares.

Posto, norma nuclear e produto interno

O posto (em inglês rank) de uma matriz é o menor $k < \rho$ tal que a matriz pode ser escrita como pode ser escrita como:

$$X = UV \quad , \quad U \in \mathbb{R}^{n \times k} \quad , \quad V \in \mathbb{R}^{k \times m}$$

A norma nuclear de X é a norma ℓ_1 dos seus valores singulares

$$||X||_* = \sum_{i=1}^{\rho} \sigma_i$$

Denota-se $A \bullet B$ o produto interno de matrizes como vetores em $\mathbb{R}^{n \times m}$, ou seja,

$$A \bullet B = \sum_{i=1}^{n} \sum_{j=1}^{m} A_{ij} B_{ij} = \mathrm{Tr}(AB^T)$$

Low-rank matrix completion problem

Formulação

- Imagine que uma exista uma matriz de opiniões $M \in \{0, 1, *\}^{n \times m}$ sobre filmes/séries onde:

$$M_{ij} = \begin{cases} 0 & \text{se pessoa } i \text{ não gosta do filme } j \\ 1 & \text{se pessoa } i \text{ gosta do filme } j \\ * & \text{se pessoa } i \text{ não opinou/viu sobre } j \end{cases}$$

- O objetivo é completar essa matriz com 1's e 0's nas entradas faltantes.
- A premissa dessa formulação é assumir que a matriz tem rank baixo, ou seja, a matriz M "verdadeira"pode ser escrita como :

$$M = UV \quad , \quad U \in \mathbb{R}^{n \times k} \quad , \quad V \in \mathbb{R}^{k \times m}$$

intuitivamente isso significa que apenas k fatores (diretor, gênero, atores, etc) descrevem as entradas de M.

Formulação

Vamos chamar de $||\cdot||_{OB}$ a norma euclidiana das entradas observadas da matriz, ou seja,

$$||M||_{OB}^2 = \sum_{M_{ij} \neq *} M_{ij}^2$$

Gostaríamos de minimizar essa norma $||\cdot||_{OB}^2$ de forma que a matriz resultante tenha rank baixo.

$$\min_{X \in \mathbb{R}^{n \times m}} ||X - M||_{OB}^2$$

$$\text{s.a. } \text{rank}(X) \leq k$$

Problema: Não sei o que significa essa restrição em termos de programação matemática. $\text{rank}(X) \leq k$. O padrão é substituir rank pela $||\cdot||_*$, isso é uma relaxação.

Formulação

Com as alterações o problema fica:

$$\min_{X \in \mathbb{R}^{n \times m}} ||X - M||_{OB}^2$$

$$\text{s.a. } ||X||_* \leq k$$

E pode-se mostrar que:

$$||X||_* = \text{Tr}\left(\sqrt{XX^T}\right)$$

o que nos leva ao problema:

$$\min_{X \in \mathbb{R}^{n \times m}} ||X - M||_{OB}^2$$

$$\text{s.a. } \text{Tr}\left(\sqrt{XX^T}\right) \leq k$$

Como e porque usar para matrix completion

Como?

Vamos reestruturar o problema chamando

$$\nabla f(X^t) = \nabla_t = (X^t - M)_{OB} \begin{cases} (X^t_{ij} - M_{ij}), & (i,j) \in OB \\ 0, & \text{caso contrário} \end{cases}$$

Assumindo (como no livro) que a matriz X é quadrada e simétrica, nesse caso norma nuclear e traço são iguais. O problema vira.

$$\min_{X \in \mathbb{R}^{n \times n}} X \bullet \nabla_t$$

$$\text{s.a. } \operatorname{Tr} X \leq k$$

e é possível mostrar (exercício do livro) que isso é equivalente a:

$$\min_{x \in \mathbb{R}^n} x^T \nabla_t x$$

$$\text{s.a. } ||x||_2^2 \leq k$$

que é equivalente a calcular o autovetor associado ao maior autovalor de ∇_t (direção que o gradiente mais cresce).

Como?

Chamando de $v_{max}(\nabla_t)$ essa conta temos o algoritmos conditional gradient para esse problema:

Algorithm 23 Conditional gradient for matrix completion

1: Let X^1 be an arbitrary matrix of trace k in $\mathcal{K}$.
2: **for** $t = 1$ to T **do**
3: $\quad \mathbf{v}_t = \sqrt{k} \cdot v_{\max}(-\nabla_t)$.
4: $\quad X^{t+1} = X^t + \eta_t(\mathbf{v}_t\mathbf{v}_t^\top - X^t)$ for $\eta_t \in (0, 1)$.
5: **end for**

Por que?

In this setting, the convex set $\mathcal{K}$ is the set of bounded nuclear norm matrices. Projecting a matrix onto this set amounts to calculating the SVD of the matrix, which is similar in computational complexity to algorithms for matrix diagonalization or inversion. The best known algorithms for matrix diagonalization are superlinear in the matrices' size, and thus impractical for large datasets that are common in applications.

In contrast, the CG method does not require projections at all, and replaces them with linear optimization steps over the convex set, which we have observed to amount to singular vector computations. The latter can be implemented to take linear time via the power method (or Lanczos algorithm, see bibliography).

Figura: Paragrafo do capítulo 7 do livro do Hazan

Online Conditional Gradient e seu regret

Online Conditional Gradient

Algorithm 24 Online conditional gradient

1: Input: convex set $\mathcal{K}$, T, $\mathbf{x}_1 \in \mathcal{K}$, parameters η, $\{\sigma_t\}$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Play $\mathbf{x}_t$ and observe f_t.
4: Let $F_t(\mathbf{x}) = \eta \sum_{\tau=1}^{t-1} \nabla_\tau^\top \mathbf{x} + \|\mathbf{x} - \mathbf{x}_1\|^2$.
5: Compute $\mathbf{v}_t = \arg\min_{\mathbf{x} \in \mathcal{K}} \{\nabla F_t(\mathbf{x}_t) \cdot \mathbf{x}\}$.
6: Set $\mathbf{x}_{t+1} = (1 - \sigma_t)\mathbf{x}_t + \sigma_t \mathbf{v}_t$.
7: **end for**

Figura: Algoritmo Online Conditional Gradient do livro do Hazan

Regret

Theorem 7.2. Online conditional gradient (Algorithm 24) with parameters $\eta = \frac{G}{DT^{3/4}}, \sigma_t = \min\{1, \frac{2}{t^{1/2}}\}$, attains the following guarantee

$$\text{regret}_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}^\star \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}^\star) \ \leq 8DGT^{3/4}$$

Lemma 7.3. Assume that the parameters η, σ_t are chosen such that $\eta G \sqrt{h_{t+1}} \leq \frac{D^2}{2}\sigma_t^2$. Then the iterates $\mathbf{x}_t$ of Algorithm 24 satisfy for all $t \geq 1$

$$h_t \leq 2D^2\sigma_t.$$

Regret

Theorem 5.1. The RFTL Algorithm 10 attains for every $\mathbf{u} \in \mathcal{K}$ the following bound on the regret:

$$\text{regret}_T \leq 2\eta \sum_{t=1}^{T} \|\nabla_t\|_t^{*2} + \frac{R(\mathbf{u}) - R(\mathbf{x}_1)}{\eta}.$$

If an upper bound on the local norms is known, i.e. $\|\nabla_t\|_t^* \leq G_R$ for all times t, then we can further optimize over the choice of η to obtain

$$\text{regret}_T \leq 2D_R G_R \sqrt{2T}.$$

OBRIGADO

Guilherme Bodin

13 de Novembro de 2018