

Método das Direções Alternadas para Multiplicadores com Aplicações em Problemas de Suporte Vetorial Distribuído

Por *Caio Vinícius Dadauto* 164556

Orientador *Prof. Dr. Paulo José da Silva e Silva*

*Defesa de dissertação junto
ao programa de Mestrado em
Matemática Aplicada da UNICAMP*



Universidade Estadual de Campinas - UNICAMP
Instituto de Matemática, Estatística e Computação Científica - IMECC



19 de março de 2018

Introdução

Otimização

SVM

SVM Distribuído

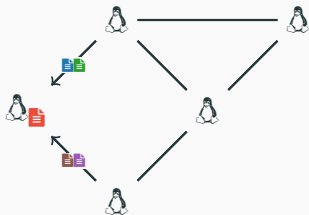
Simulações Numéricas

Introdução

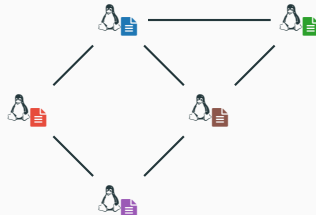
- Motivação
- Contexto Teórico
- Objetivo

Problema

Aprendizado supervisionado para a classificação binária com **conjunto de treino distribuído** entre diferentes nós (agentes) passíveis de comunicação através de uma **rede conexa e descentralizada**.

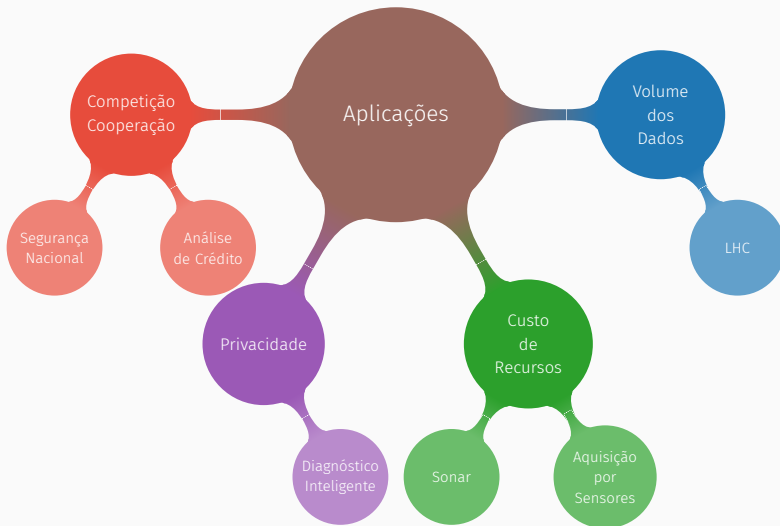


Centralizado



Descentralizado

❓ Há aplicações reais que são descritas por este problema?



O SVM (Máquina de Suporte Vetorial) é um algoritmo de aprendizagem de máquina supervisionado. Determina um modelo preditivo a partir de um conjunto de treino devidamente classificado.

❓ Por que utilizar o SVM?

- ✓ Extremamente bem sucedida;
- ✓ Flexível à casos lineares, não lineares, de classificação binárias ou não e de regressão;
- ✓ Problema naturalmente convexo;

Problema

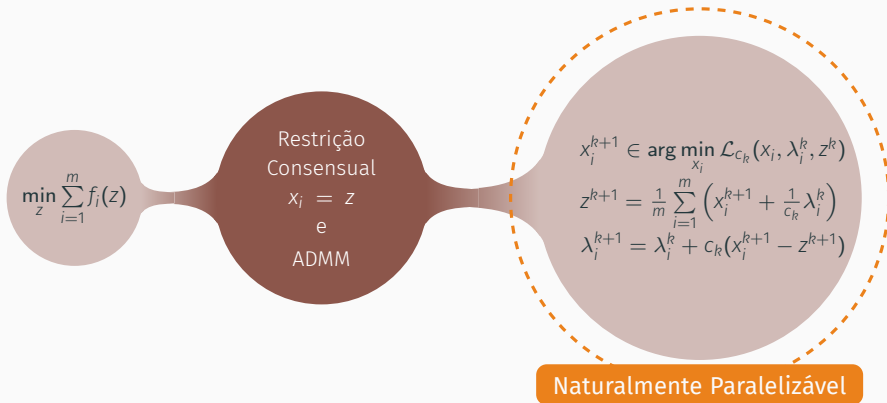
Solução canônica considera dados centralizados

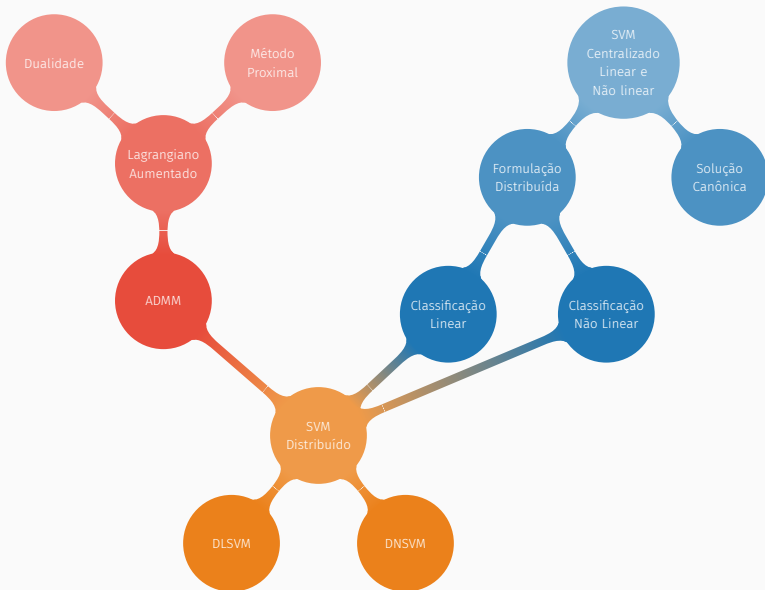
O método de direções alternadas para multiplicadores (ADMM) é uma versão radical do lagrangiano aumentado inexato no qual se faz, por iteração, uma minimização em cada variável separadamente seguida, imediatamente, por uma atualização dos multiplicadores.

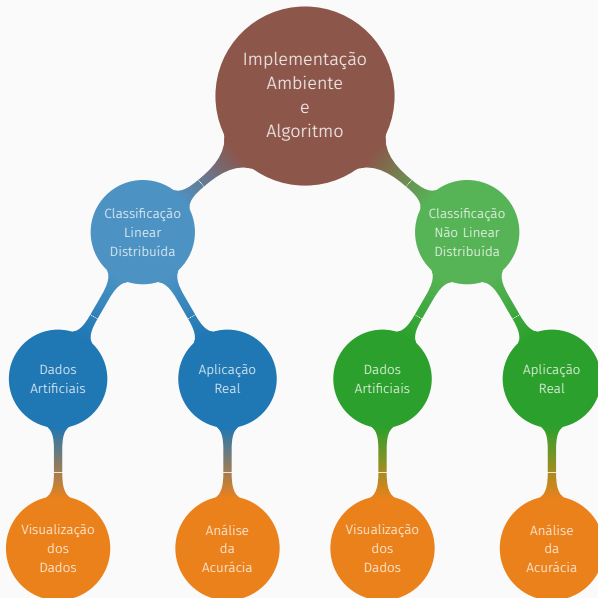
O ADMM apresenta um algoritmo naturalmente paralelizável quando aplicado à problemas separáveis em que são impostas **restrições consensuais**.

Restrição Consensual Força o consenso entre os processos, ou seja, garante que os processos concordem sobre uma mesma solução.

Exemplo simples de aplicação,







Introdução

Otimização

SVM

SVM Distribuído

Simulações Numéricas

Otimização

- Dualidade
- Método Proximal
- Lagrangiano Aumentado
- Método ADMM

Seja $f : \mathbb{R}^n \mapsto \mathbb{R}$, $g : \mathbb{R}^n \mapsto \mathbb{R}^m$ e $h : \mathbb{R}^n \mapsto \mathbb{R}^l$ funções contínuas, toma-se

Primal

$$\begin{array}{ll} \min & f(x) \\ \text{s.a.} & h(x) = 0 \\ & g(x) \leq 0 \\ & x \in \mathbb{R}^n \end{array}$$

Dual

$$\begin{array}{ll} \max & q(\mu, \lambda) \\ \text{s.a.} & \mu \geq 0 \\ & \lambda \in \mathbb{R}^l \end{array}$$

em que $q(\mu, \lambda) = \min_{x \in \mathbb{R}^n} \{\mathcal{L}(x, \mu, \lambda)\}$, com os conjuntos de soluções primal (X^*) e dual (M^*) não vazios e

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \lambda^T h(x) + \mu^T g(x)$$

em que $\mu \in \mathbb{R}^m$ e $\lambda \in \mathbb{R}^l$.

Dualidade Fraca

Seja o primal e o dual dados, tem-se que $q^* \leq f^*$, em que $f^* = f(x^*)$ e $q^* = q(\mu^*, \lambda^*)$ com $x^* \in X^*$ e $(\lambda^*, \mu^*) \in M^*$.

Dualidade Forte

Seja f e g convexas com $h(x) = Ax - b$ linear. Ainda, assume-se válida a condição de Slater, ou seja, existe $\bar{x} \in \{x \mid Ax - b = 0\}$ tal que $g(\bar{x}) < 0$. Então, $q^* = f^*$.

Seja o seguinte problema primal

$$\begin{array}{ll} \min & f_1(x) + f_2(Ax) \\ \text{s.a.} & x \in \mathbb{R}^n \end{array}$$

em que $f_1 : \mathbb{R}^n \rightarrow \mathbb{R}$, $f_2 : \mathbb{R}^m \rightarrow \mathbb{R}$ e $A \in \mathbb{R}^{m \times n}$. É possível reescrevê-lo como

Primal

$$\begin{array}{ll} \min & f_1(x_1) + f_2(x_2) \\ \text{s.a.} & x_2 = Ax_1 \\ & x_1 \in \mathbb{R}^n \\ & x_2 \in \mathbb{R}^m \end{array}$$

Dual

$$\begin{array}{ll} \min & f_1^*(A^T \lambda) + f_2^*(-\lambda) \\ \text{s.a.} & \lambda \in \mathbb{R}^m \end{array}$$

em que $f^*(y) = \max_x \{y^T x - f(x)\}$ é a função conjugada de f .

Dualidade de Fenchel

Seja $f^* = q^*$ e (x^*, λ^*) solução ótima primal e dual. Então,

$$x^* \in \arg \min_{x \in \mathbb{R}^n} \{f_1(x) - x^T A^T \lambda^*\} \quad Ax^* \in \arg \min_{z \in \mathbb{R}^m} \{f_2(z) + z^T \lambda^*\}$$

Algoritmo

Seja $\min_x f(x)$. Inicializando x , faz-se

$$x_{k+1} \in \arg \min_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2c_k} \|x - x_k\|^2 \right\}$$

É possível mostrar que o custo $f(x_k)$ e a distância a qualquer minimizador $\|x_k - x^*\|$ de f são **não crescentes** a cada iteração.

Convergência

Seja $\{x_k\}$ gerado pelo método proximal. Então, se $\sum_{i=1}^{\infty} c_k = \infty$

$$f(x_k) \rightarrow f^*$$

e se X^* não vazio, $\lim_{k \rightarrow \infty} x_k \in X^*$

O problema $\min_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2c_k} \|x - x_k\|^2 \right\}$ pode ser reescrito de forma a se enquadra na classe de problemas de interesse à dualidade de Fenchel.

Primal

$$\begin{array}{ll} \min & f_1(x_1) + f_2(x_2) \\ \text{s.a.} & x_2 = x_1 \\ & x_1, x_2 \in \mathbb{R}^n \end{array}$$

Dual

$$\begin{array}{ll} \min & f^*(\lambda) - \lambda^T x_k + \frac{c_k}{2} \|\lambda\|^2 \\ \text{s.a.} & \lambda \in \mathbb{R}^n \end{array}$$

em que $x_1 = x$, $x_2 = x_1$, $f_1(x) = f(x)$, $f_2(x) = \frac{1}{2c_k} \|x - x_k\|^2$ e $A = I$.

A partir da Dualidade de Fenchel, tem-se que

Algoritmo

Inicializando λ e x , encontra-se λ_{k+1} a partir de x_k , a saber

$$\begin{aligned}\lambda_{k+1} &\in \arg \min_{\lambda \in \mathbb{R}^n} \left\{ f^*(\lambda) - \lambda^T x_k + \frac{c_k}{2} \|\lambda\|^2 \right\} \\ x_{k+1} &= x_k - c_k \lambda_{k+1}\end{aligned}$$

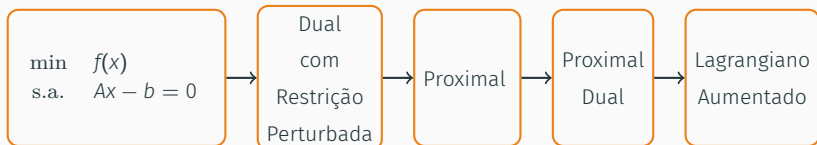
A convergência é garantida pelo método proximal primal, uma vez que a sequência primal gerada pelo proximal dual é a mesma gerada pelo método proximal primal.

Abordagem em contexto generalizado

- ✗ Convergência exige $c_k \rightarrow \infty$;
- ✗ Condição $c_k \rightarrow \infty$ pode tornar a hessiana mal condicionada para k grande;

Abordagem em programação convexa

- ✓ Segue naturalmente do método proximal;
- ✓ Basta que $\sum_{i=0}^{\infty} c_k = \infty$;
- ✓ Convergência determinada a partir do método proximal;



Algoritmo

Inicializando o multiplicador λ , faz-se

$$\begin{aligned}x_{k+1} &\in \min_{x \in \mathbb{R}^n} \mathcal{L}_{c_k}(x, \lambda_k) \\ \lambda_{k+1} &= \lambda_k + c_k(Ax_{k+1} - b)\end{aligned}$$

em que $\mathcal{L}_{c_k}(x, \lambda_k) = f(x) + (Ax - b)^T \lambda_k + \frac{c_k}{2} \|Ax - b\|^2$.

Todo ponto de acumulação de $\{x_k\}$ é solução do primal, segue da teoria do método proximal.

Primal

$$\begin{array}{ll} \min & f(x) + g(z) \\ \text{s.a.} & Ax + Bz - d = 0 \end{array}$$

Dual

$$\begin{array}{ll} \max & v(\lambda) + w(\lambda) \\ \text{s.a.} & \lambda \in \mathbb{R}^r \end{array}$$

em que $v = \min_{x \in \mathbb{R}^n} \{f(x) + \lambda^T Ax\}$ e $w = \min_{z \in \mathbb{R}^m} \{g(z) + \lambda^T (Bz - d)\}$.

Algoritmo

Inicializando o multiplicador λ , faz-se

$$\begin{aligned} x_{k+1} &\in \arg \min_{x \in \mathbb{R}^n} \{\mathcal{L}_c(x, z_k, \lambda_k)\} \\ z_{k+1} &\in \arg \min_{z \in \mathbb{R}^m} \{\mathcal{L}_c(x_{k+1}, z, \lambda_k)\} \\ \lambda_{k+1} &= \lambda_k + c(Ax_{k+1} + Bz_{k+1} - d) \end{aligned}$$

Convergência

Seja X^* não vazio. Então, todo ponto de acumulação $(\bar{x}, \bar{z}, \bar{\lambda})$ da sequência gerada pelo método ADMM é tal que (\bar{x}, \bar{z}) é minimizador do problema primal e $\bar{\lambda}$ é minimizador do problema dual.

Introdução

Otimização

SVM

SVM Distribuído

Simulações Numéricas

SVM

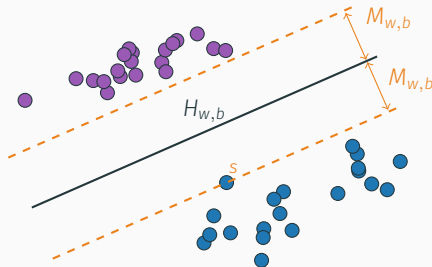
- Classificação Linear
- Classificação Não Linear

Tradicionalmente, o SVM é utilizado para o aprendizado supervisionado aplicado a classificação binária. Para tanto busca o melhor hiperplano de separação dos dados de treino.

Margem Menor distância relativa ao hiperplano.

Objetivo Determinar o plano que maximiza a margem.

Vetores de Suporte (s) Distam exatamente uma margem do hiperplano.



Dado um hiperplano $H_{w,b} := \{x \mid w^T x + b = 0\}$ é possível mensurar a distância relativa ao hiperplano como

$$f_M(x, y, w, b) = \frac{y}{\|w\|} (w^T x + b)$$

em que $x \in \mathbb{R}^n$ e $y \in \{-1, 1\}$.

- ❗ $f_M(x, y, w, b)$ é invariante por múltiplos escalares de (w, b) .
- ❗ $f_M(x, y, w, b)$ é empregada como mecanismo para averiguar se a predição apontada pelo modelo $H_{w,b}$ para um dado de teste (x_{teste}, y_{teste}) está correta.
- ❗ Naturalmente, $w^T x + b$ é utilizada para classificar dados ainda não rotulados.

$$\max_{(w,b) \in \mathbb{R}^{n+1}} \min_{(x,y) \in S_{\text{treino}}} f_M(x, y, w, b)$$

$$M_{w,b} = \min_{(x,y) \in S_{\text{treino}}} f_M(x, y, w, b)$$

$$\begin{aligned} \max \quad & M_{w,b} \\ \text{s.a.} \quad & M_{w,b} - \frac{y_i}{\|w\|} (w^T x_i + b) \leq 0 \quad i = 1, \dots, m \\ & (w, b) \in \mathbb{R}^{n+1} \end{aligned}$$

Escala tal que $\|w\| = \frac{1}{M_{w,b}}$

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|^2 \\ \text{s.a.} \quad & 1 - y_i (w^T x_i + b) \leq 0 \quad i = 1, \dots, m \\ & (w, b) \in \mathbb{R}^{n+1} \end{aligned}$$

Primal

Flexibilização da Margem

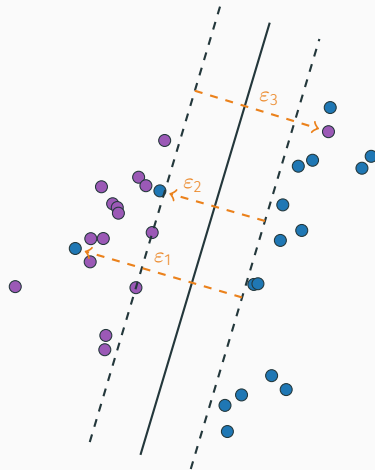
$$\frac{y_i}{\|w\|} (w^T x_i + b) \geq M_{w,b} (1 - \varepsilon_i)$$

Adição de Regularização l_1

$$\sum_{i=0}^m \varepsilon_i$$

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=0}^m \varepsilon_i \\ \text{s.a.} \quad & 1 - \varepsilon_i - y_i (w^T x_i + b) \leq 0 \quad i = 1, \dots, m \\ & -\varepsilon_i \leq 0 \quad i = 1, \dots, m \\ & (w, b) \in \mathbb{R}^{n+1} \end{aligned}$$

Primal



$$q(\mu, \eta) = \min_{(w, b, \varepsilon) \in \mathbb{R}^{n+2}} \mathcal{L}(w, b, \varepsilon, \mu, \eta)$$

$$\begin{array}{ll} \max & q(\mu, \eta) \\ \text{s.a.} & (\mu, \eta) \in \mathbb{R}^{2m} \end{array}$$

Lagrangiano convexo

$$\begin{array}{ll} \max & \sum_{i=0}^m \mu_i - \sum_{i=0}^m \sum_{j=0}^m \mu_i \mu_j y_i y_j x_i^T x_j \\ \text{s.a.} & \sum_{i=0}^m \mu_i y_i = 0 \\ & 0 \leq \mu_i \leq C \quad i = 1, \dots, m \end{array}$$

$$w = \sum_{i=0}^m \mu_i y_i x_i$$

Problema de otimização quadrática, o qual pode ser solucionado por diversos métodos.

Normalmente, há a exigência de que os dados estejam centralizados.

Procura determinar um espaço de *Hilbert* \mathcal{F} com dimensão $M \in [1, \infty]$ e um mapeamento $\phi(\cdot) : \mathbb{R}^n \rightarrow \mathcal{F}$ de forma que os dados do conjunto $\bar{S} = \{(\phi(x_i), y_i), i = 1, \dots, m\}$ sejam separados segundo a classe y de forma acurada por um hiperplano em \mathcal{F} .

- ❗ O mapa ϕ muitas vezes não é conhecido.
- ❗ Quando conhecido, resolver o SVM apenas aplicando ϕ é impraticável devido ao crescimento demasiado da complexidade computacional introduzida pela alta dimensionalidade de \mathcal{F} .

Considera \mathcal{F} um espaço de *Hilbert* reproduzido por núcleos (RKHS) e toma o mapa $\phi(x)$ como sendo o núcleo associado a \mathcal{F} $K(\cdot, x)$, em que K é único. Dessa forma,

$$\langle \phi(x), \phi(y) \rangle_{\mathcal{F}} = K(x, y)$$

$$\begin{aligned} \max \quad & \sum_{i=0}^m \mu_i - \sum_{i=0}^m \sum_{j=0}^m \mu_i \mu_j y_i y_j K(x_i, x_j) \\ \text{s.a.} \quad & \sum_{i=0}^m \mu_i y_i = 0 \\ & 0 \leq \mu_i \leq C \quad i = 1, \dots, m \end{aligned}$$



$$w = \sum_{i=0}^m \mu_i y_i \phi(x_i)$$



Função para a classificação

$$\text{sign} \left(\sum_{i=0}^m \mu_i y_i K(x_i, x) + b \right)$$

Alguns núcleos de interesse prático são,

Linear $k(x_i, x_j) = x_i^T x_j$

Polinomial $k(x_i, x_j) = (x_i^T x_j + c)^d \quad d \in \mathbb{N} \text{ e } c \geq 0$

RBF $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad \gamma > 0$

Introdução

Otimização

SVM

SVM Distribuído

Simulações Numéricas

SVM Distribuído

- Classificação Linear Distribuída
- Classificação Não Linear Distribuída

Seja uma rede conexa com N nós em que cada nó possui parte dos dados, a saber o nó i possui m_i dados.

$$\begin{aligned}
 \min \quad & \frac{1}{2} \sum_{i=1}^N \|w_i\|^2 + C \sum_{i=1}^N \sum_{j=1}^{m_i} \varepsilon_{ij} \\
 \text{s.a.} \quad & 1 - y_{ij}(x_{ij}^T w_i + b_i) - \varepsilon_{ij} \leq 0 \quad i = 1, \dots, N \quad j = 1, \dots, m_i \\
 & -\varepsilon_{ij} \leq 0 \quad i = 1, \dots, N \quad j = 1, \dots, m_i \\
 & w_i - w_k = 0 \quad i = 1, \dots, N \quad k \in \mathbb{K}_i \\
 & b_i - b_k = 0 \quad i = 1, \dots, N \quad k \in \mathbb{K}_i \\
 & (w_i, b_i) \in \mathbb{R}^{n+1} \quad i = 1, \dots, N,
 \end{aligned}$$

Pois a
rede é conexa

em que \mathbb{K}_i é o conjunto de vizinhos do nó i e os dados são rotulados por j . Note que as restrições consensuais garantem que a solução seja única entre os nós.

Definindo-se,

$$\begin{aligned} v_i &= [w_i^T \ b_i]^T && \in \mathbb{R}^{n+1} \\ Y_i &= \text{diag}(y_{i1}, \dots, y_{im_i}) && \in \mathbb{R}^{m_i \times m_i} \\ X_i^T &= \begin{bmatrix} x_{i1} & \dots & x_{im_i} \\ 1 & \dots & 1 \end{bmatrix} && \in \mathbb{R}^{n+1 \times m_i} \\ \mathcal{E}_i &= [\varepsilon_{i1} \ \dots \ \varepsilon_{im_i}]^T && \in \mathbb{R}^{m_i} \\ 1_i &= [1 \ \dots \ 1]^T && \in \mathbb{R}^{m_i} \end{aligned}$$

o problema pode ser reescrito como

Desacopla

v_i e v_k

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{i=1}^N v_i^T (I - e_{n+1} e_{n+1}^T) v_i + C \sum_{i=1}^N \mathcal{E}_i^T 1_i \\ \text{s.a.} \quad & 1_i - Y_i X_i v_i - \mathcal{E}_i \leq 0 && i = 1, \dots, N \\ & -\mathcal{E}_i \leq 0 && i = 1, \dots, N \\ & \begin{cases} v_i - u_{ik} = 0 \\ v_k - u_{ik} = 0 \end{cases} && i = 1, \dots, N \quad k \in \mathbb{K}_i \\ & v_i \in \mathbb{R}^{n+1} && i = 1, \dots, N, \end{aligned}$$

$$\{v_k - u_{ik}\}_{\forall i, k \in \mathbb{K}_i}$$

$$\{v_k - u_{ik}\}_{\forall i, k \in \mathbb{K}_i}$$

$$Av - Bu = 0$$

em que

$$v \in \mathbb{R}^{M(n+1)}$$

$$u \in \mathbb{R}^{2E(n+1)}$$

$$A \in \mathbb{R}^{4E(n+1) \times M(n+1)}$$

$$B \in \mathbb{R}^{4E(n+1) \times 2E(n+1)}$$

com E sendo o número de arestas do grafo.

$$\begin{aligned} \min \quad & \sum_{i=0}^M 1_i \mathcal{E}_i \\ \mathcal{E}_i \geq & 1_i - Y_i X_i v_i \\ \mathcal{E}_i \geq & 0 \end{aligned}$$

$$\mathcal{E}_i = \max\{0, 1_i - Y_i X_i v_i\}$$

Definindo-se

$$\begin{aligned}f_i(v_i) &= \frac{1}{2}v_i^T(I - e_{n+1}e_{n+1}^T)v_i + C1_i^T \max\{0, 1_i - Y_iX_iv_i\} \\F(v) &= \sum_{i=1}^N f_i(v_i),\end{aligned}$$

o problema de suporte vetorial distribuído pode ser reescrito de forma a se enquadrar na classe de problemas de interesse ao ADMM, a saber

$$\begin{aligned}\min \quad & F(v) \\ \text{s.a.} \quad & Av - Bu = 0 \\ & v \in \mathbb{R}^{N(n+1)} \\ & u \in \mathbb{R}^{2E(n+1)},\end{aligned}$$

em que o lagrangiano aumentado é dado por

$$\mathcal{L}_c(v, u, \lambda) = F(v) + (Av - Bu)^T \lambda + \frac{c}{2} \|Av - Bu\|^2$$

Aplicando o ADMM,

$$\begin{aligned} v^{l+1} &\in \arg \min_{v \in \mathbb{R}^{N(n+1)}} \mathcal{L}_c(v, u^l, \lambda^l) \\ u^{l+1} &\in \arg \min_{u \in \mathbb{R}^{2E(n+1)}} \mathcal{L}_c(v^{l+1}, u, \lambda^l) \\ \lambda^{l+1} &= \lambda^l + c(Av^{l+1} - Bu^{l+1}), \end{aligned}$$

Escrevendo a restrição de forma explícita e reintroduzindo a variável \mathcal{E}_i o lagrangiano pode ser reescrito como

$$\mathcal{L}_c((v, \mathcal{E}), u, \lambda) = \sum_{i=1}^N \left\{ f(v_i, \mathcal{E}_i) + \sum_{k \in \mathbb{K}_i} \left[(v_i - u_{ik})^T \lambda_{ik_1} + (v_i - u_{ki})^T \lambda_{ki_2} + \frac{\epsilon}{2} \|v_i - u_{ik}\|^2 + \frac{\epsilon}{2} \|v_i - u_{ki}\|^2 \right] \right\},$$

Somas Separáveis em (v_i, \mathcal{E}_i) e u_{ik}

O ADMM pode ser reescrito como um conjunto de iterações para cada nó i , a saber

$$\begin{aligned} (v_i, \mathcal{E}_i)^{l+1} &\in \arg \min_{(v_i, \mathcal{E}_i) \in \mathbb{H}_i} \mathcal{V}_c \left((v_i, \mathcal{E}_i), \{u_{ik}^l, u_{ki}^l, \lambda_{ik_1}^l, \lambda_{ki_2}^l\}_{k \in \mathbb{K}_i} \right) \\ u_{ik}^{l+1} &\in \arg \min_{u_{ik} \in \mathbb{R}^{n+1}} \mathcal{U}_c \left(v_i^{l+1}, v_k^{l+1}, u_{ik}, \lambda_{ik_1}^l, \lambda_{ki_2}^l \right) \quad \forall k \in \mathbb{K}_i \\ \lambda_{ik_1}^{l+1} &= \lambda_{ik_1}^l + c(v_i^{l+1} - u_{ik}^{l+1}) \quad \forall k \in \mathbb{K}_i \\ \lambda_{ki_2}^{l+1} &= \lambda_{ki_2}^l + c(v_i^{l+1} - u_{ki}^{l+1}) \quad \forall k \in \mathbb{K}_i \end{aligned}$$

em que \mathbb{H}_i contem as restrições, ou seja,

$$\mathbb{H}_i = \{(v_i, \mathcal{E}_i) \mid 1_i - Y_i X_i v_i - \mathcal{E}_i \leq 0, -\mathcal{E}_i \leq 0 \quad \forall v_i \in \mathbb{R}^{n+1} \text{ e } \mathcal{E}_i \in \mathbb{R}^{m_i}\}$$

Ainda, é importante notar que as funções \mathcal{V}_c e \mathcal{U}_c são as parcelas pertinentes do lagrangiano aumentado referente a cada minimização.

Algoritmo

Para todo nó i , faz-se

$$\begin{aligned} (v_i, \mathcal{E}_i)^{l+1} &\in \arg \min_{(v_i, \mathcal{E}_i) \in \mathbb{H}_i} \mathcal{V}_c \left((v_i, \mathcal{E}_i), \{u_{ik}^l, u_{ki}^l, \lambda_{ik_1}^l, \lambda_{ki_2}^l\}_{k \in \mathbb{K}_i} \right) \\ u_{ik}^{l+1} &\in \arg \min_{u_{ik} \in \mathbb{R}^{n+1}} \mathcal{U}_c \left(v_i^{l+1}, v_k^{l+1}, u_{ik}, \lambda_{ik_1}^l, \lambda_{ki_2}^l \right) & \forall k \in \mathbb{K}_i \\ \lambda_{ik_1}^{l+1} &= \lambda_{ik_1}^l + c(v_i^{l+1} - u_{ik}^{l+1}) & \forall k \in \mathbb{K}_i \\ \lambda_{ki_2}^{l+1} &= \lambda_{ki_2}^l + c(v_i^{l+1} - u_{ki}^{l+1}) & \forall k \in \mathbb{K}_i \end{aligned}$$

! Simplificação

- Praticidade;
- Overhead na comunicação (v_k^l , u_{ki}^l e $\lambda_{ki_2}^l$);

Notando que $\min_{u_{ik} \in \mathbb{R}^{n+1}} \mathcal{U}_c(v_i^{l+1}, v_k^{l+1}, u_{ik}, \lambda_{ik_1}^l, \lambda_{ik_2}^l)$ é quadrática e irredutível, tem-se que

$$\begin{array}{l} \nabla_{u_{ik}} \mathcal{U}_c = 0 \\ \lambda_{ik_1}^0 = 0 \\ \lambda_{ik_2}^0 = 0 \end{array} \rightarrow \begin{array}{l} u_{ik}^{l+1} = \frac{1}{2}(v_i^{l+1} + v_k^{l+1}) \\ \lambda_{ik}^{l+1} = \lambda_{ik}^l + \frac{c}{2}(v_i^{l+1} - v_k^{l+1}) \\ \text{em que } \lambda_{ik}^l = \lambda_{ik_1}^l = -\lambda_{ik_2}^l \end{array} \rightarrow \begin{array}{l} u_{ik}^l = u_{ki}^l \quad \forall l \\ \lambda_{ik}^l = -\lambda_{ki}^l \quad \forall l. \end{array}$$

Dessa forma, evita-se a dependência em u_{ik}^l e u_{ki}^l quanto a minimização da função \mathcal{V}_c , a saber

$$\mathcal{V}_c(v_i, \mathcal{E}_i, v_i^l, \{v_k^l\}_{k \in \mathbb{K}_i}, \lambda_i^l) = f(v_i, \mathcal{E}_i) + 2v_i^T \lambda_i^l + c \sum_{k \in \mathbb{K}_i} \left\| v_i - \frac{1}{2}(v_i^l + v_k^l) \right\|^2$$

$$\text{em que } \lambda_i^l = \sum_{k \in \mathbb{K}_i} \lambda_{ik}^l.$$

Algoritmo

Para todo nó i , faz-se

$$\begin{aligned}(v_i, \mathcal{E}_i)^{l+1} &\in \arg \min_{(v_i, \mathcal{E}_i) \in \mathbb{H}_i} \mathcal{V}_c \left((v_i, \mathcal{E}_i), v_i^l, \{v_k^l\}_{k \in \mathbb{K}_i}, \lambda_i^l \right) \\ \lambda_i^{l+1} &= \lambda_i^l + \frac{c}{2} \sum_{k \in \mathbb{K}_i} (v_i^{l+1} - v_k^{l+1})\end{aligned}$$

! Simplificação

- Desassociar os vetores v_i e \mathcal{E}_i ;

Seja o problema $\min_{(v_i, \mathcal{E}_i) \in \mathbb{H}_i} \mathcal{V}_c \left((v_i, \mathcal{E}_i), v_i^l, \{v_k^l\}_{k \in \mathbb{K}_i}, \lambda_i^l \right)$ e considere os vetores μ_i e η_i em \mathbb{R}^{m_i} como sendo as variáveis duais deste problema. Aplicando KKT e notando que o problema em questão é quadrático, é possível derivar que

$$\eta_i^{l+1} = C1_i - \mu_i^{l+1}$$

$$v_i^{l+1} = D_i^{-1} \left(X_i^T Y_i \mu_i^{l+1} - r_i^l \right)$$

$$\mu_i^{l+1} \in \arg \max_{0 \leq \mu_i \leq C1_i} \left\{ -\frac{1}{2} \mu_i^T Y_i X_i D_i^{-1} X_i^T Y_i \mu_i + (1_i + Y_i X_i D_i^{-1} r_i^l)^T \mu_i \right\}$$

em que

$$r_i^l = 2\lambda_i^l - c \sum_{k \in \mathbb{K}_i} (v_i^l + v_k^l)$$

$$D_i = (I - e_{n+1} e_{n+1}^T) + 2c \# \mathbb{K}_i I$$

Algoritmo DLSVM

Inicializa-se v_i^0 $i = 1, \dots, N$, e toma-se como nulos os multiplicadores λ_i^0 $i = 1, \dots, N$. Então, faz-se para cada nó

$$\mu_i^{l+1} \in \arg \max_{0 \leq \mu_i \leq C1_i} \left\{ -\frac{1}{2} \mu_i^T Y_i X_i D_i^{-1} X_i^T Y_i \mu_i + (1_i + Y_i X_i D_i^{-1} r_i^l)^T \mu_i \right\}$$

$$v_i^{l+1} = D_i^{-1} \left(X_i^T Y_i \mu_i^{l+1} - r_i^l \right)$$

$$\lambda_i^{l+1} = \lambda_i^l + \frac{c}{2} \sum_{k \in \mathbb{K}_i} (v_i^{l+1} - v_k^{l+1})$$

Considere o núcleo $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ associado ao espaço $RKHS \mathcal{F}^M$ de dimensão $M \in [1, \infty]$, tal que define o mapa ϕ como $\phi(x) = K(\cdot, x)$ e o vetor $\omega_i \in \mathcal{F}^M$ responsável por descrever o hiperplano em \mathcal{F}^M .

Problema

Formulação anterior é impraticável, pois as restrições consensuais exigem a manipulação explícita de ω_i .

É imposto que os agentes concordem quanto a transformação linear dada por $\Phi_{\chi}\omega_i = \tilde{\omega}_i$, em que

$$\Phi_{\chi}^T = [\phi(\chi_1) \cdots \phi(\chi_p)] \in \mathcal{F}^{M \times p}$$

com $\{\chi_i\}_{i=1, \dots, p}$ vetores em \mathbb{R}^n comuns a toda rede.

! A dimensão p deve ser tal que $p \ll M$.

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{i=1}^N \|\omega_i\|_{\mathcal{F}^M}^2 + C \sum_{i=1}^N \mathcal{E}_i^T 1_i \\ \text{s.a.} \quad & 1 - Y_i(\Phi_{\chi_i} \omega_i + b_i 1_i) - \mathcal{E}_i \leq 0 \quad i = 1, \dots, N \\ & -\mathcal{E}_i \leq 0 \quad i = 1, \dots, N \\ & \tilde{\omega}_i - \tilde{\omega}_k = 0 \quad i = 1, \dots, N \quad k \in \mathbb{K}_i \\ & b_i - b_k = 0 \quad i = 1, \dots, N \quad k \in \mathbb{K}_i \\ & (\omega_i, b_i) \in \mathcal{F}^M \times \mathbb{R} \quad i = 1, \dots, N, \end{aligned}$$

Gera
classificadores
distintos entre
os agentes

Apesar de \mathcal{F}^M possuir dimensionalidade arbitrariamente grande é possível, para qualquer i , expressar a solução ω_i^* do problema formulado anteriormente como uma combinação linear finita do núcleo K , a saber

$$\omega_i^*(\cdot) = \sum_{j=1}^{m_i} \alpha_{ij}^* K(\cdot, x_{ij}) + \sum_{j=1}^p \beta_{ij}^* K(\cdot, x_j),$$

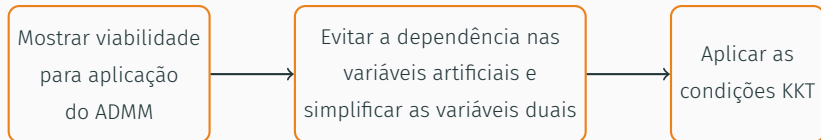
Isso se deve ao problema não linear distribuído se enquadrar na classe de problemas do teorema do **Representante Semi-Paramétrico**.

❗ Busca-se a função para classificação de novos dados, a saber

$$g_i^*(x) = \phi(x)^T \omega_i^* + b^*.$$

Para tanto, bastar determinar α_i^* e β_j^* de forma distribuída.

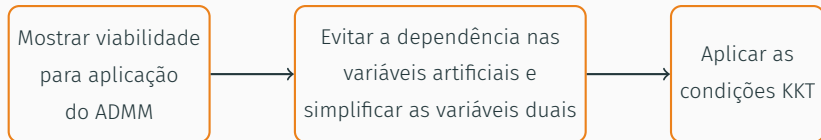
Aplica-se o mesmo processo utilizado para o caso linear



Considerando $A \in \mathbb{R}^{p \times n}$ e $B \in \mathbb{R}^{q \times n}$ com linhas dadas por a_i e b_i , define-se $\tilde{K}(Z, Z') = 2c \# \mathbb{K}_i K(Z, \chi) Q_i^{-1} K(\chi, Z')$,

$$\begin{aligned}
 K(A, B) &= \begin{bmatrix} K(a_1, b_1) & \cdots & K(a_1, b_q) \\ \vdots & \vdots & \vdots \\ K(a_p, b_1) & \cdots & K(a_p, b_q) \end{bmatrix} \\
 r_i^l &= 2\lambda_i^l - c \sum_{i \in \mathbb{K}_i} (\tilde{\omega}_i^l + \tilde{\omega}_k^l) \\
 s_i^l &= 2\zeta_i^l - c \sum_{k \in \mathbb{K}_i} (b_i^l + b_k^l) \\
 Q_i &= I + 2c \# \mathbb{K}_i K(\chi, \chi)
 \end{aligned}$$

Aplica-se o mesmo processo utilizado para o caso linear



$$\mu_i^{l+1} \in \arg \max_{0 \leq \mu_i \leq C1_i} \left\{ -\frac{1}{2} \mu_i^T Y_i \left(K(X_i, X_i) - \tilde{K}(X_i, X_i) + \frac{1_i 1_i^T}{2c \# \mathbb{K}_i} \right) Y_i \mu_i \right. \\ \left. + \left[1_i + Y_i \left(K(X_i, \chi) - \tilde{K}(X_i, \chi) \right) r_i^l + \frac{s_i^l}{2c \# \mathbb{K}_i} Y_i 1_i \right]^T \mu_i \right\}$$

$$\tilde{\omega}_i^{l+1} = \left(K(\chi, X_i) - \tilde{K}(\chi, X_i) \right) Y_i \mu_i^{l+1} - \left(K(\chi, \chi) - \tilde{K}(\chi, \chi) \right) r_i^l$$

$$b_i^{l+1} = \frac{1}{2c \# \mathbb{K}_i} (1_i^T Y_i \mu_i^{l+1} - s_i^l)$$

$$\alpha_i^{l+1} = Y_i \mu_i^{l+1}$$

$$\beta_i^{l+1} = 2c \# \mathbb{K}_i Q_i^{-1} \left(K(\chi, \chi) r_i^l - K(\chi, X_i) Y_i \mu_i^{l+1} \right) - r_i^l,$$

Algoritmo DNSVM

Inicializa-se v_i^0 e toma-se como nulos os multiplicadores λ_i^0 e ζ_i^0 para todo $i = 1, \dots, N$. Então, faz-se para cada nó

$$\mu_i^{l+1} \in \arg \max_{0 \leq \mu_i \leq C1_i} \left\{ -\frac{1}{2} \mu_i^T Y_i \left(K(X_i, X_i) - \tilde{K}(X_i, X_i) + \frac{1_i 1_i^T}{2c \# \mathbb{K}_i} \right) Y_i \mu_i \right. \\ \left. + \left[1_i + Y_i \left(K(X_i, \chi) - \tilde{K}(X_i, \chi) \right) r_i^l + \frac{s_i^l}{2c \# \mathbb{K}_i} Y_i 1_i \right]^T \mu_i \right\}$$

$$\tilde{\omega}_i^{l+1} = \left(K(\chi, X_i) - \tilde{K}(\chi, X_i) \right) Y_i \mu_i^{l+1} - \left(K(\chi, \chi) - \tilde{K}(\chi, \chi) \right) r_i^l$$

$$b_i^{l+1} = \frac{1}{2c \# \mathbb{K}_i} (1_i^T Y_i \mu_i^{l+1} - s_i^l)$$

$$\lambda_i^{l+1} = \lambda_i^l + \frac{c}{2} \sum_{i \in \mathbb{K}_j} \tilde{\omega}_i^{l+1} - \tilde{\omega}_k^{l+1})$$

$$\zeta_i^{l+1} = \zeta_i^l + \frac{c}{2} \sum_{i \in \mathbb{K}_j} (b_i^{l+1} - b_k^{l+1})$$

Introdução

Otimização

SVM

SVM Distribuído

Simulações Numéricas

Simulações Numéricas

- Implementação
- Caso Linear
- Caso Não Linear

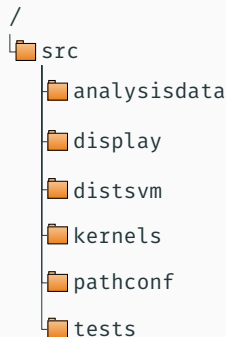
Linguagem Python 3, principais arcabouços são `mpi4py`, `cvxopt`, `numpy`, `scikit-learn` e `networkx`.

Parâmetros Busca em grade.

Acurácia Validação cruzada para 3 amostras estratificadas.

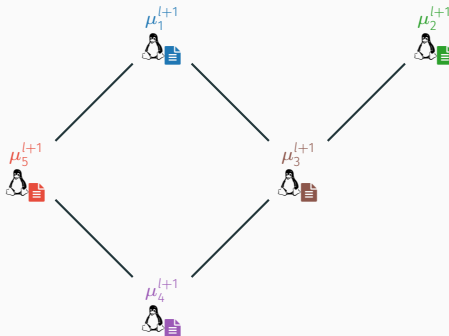
Normalização Dados de treino tomados com $\bar{\mu} = 0$ e $\sigma = 1$ a cada atributo.

Núcleo Em todos os testes não lineares é utilizado o núcleo *RBF*.



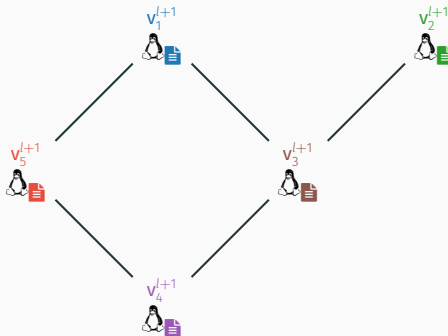
Para cada nó, faz-se

$$\mu_i^{l+1} \in \arg \max_{0 \leq \mu_i \leq C1_i} \left\{ -\frac{1}{2} \mu_i^T Y_i X_i D_i^{-1} X_i^T Y_i \mu_i + (1_i + Y_i X_i D_i^{-1} r_i^l)^T \mu_i \right\}$$

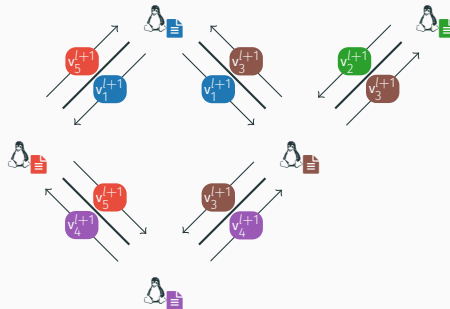


Para cada nó, faz-se

$$v_i^{l+1} = D_i^{-1} \left(X_i^T Y_i \mu_i^{l+1} - r_i^l \right)$$

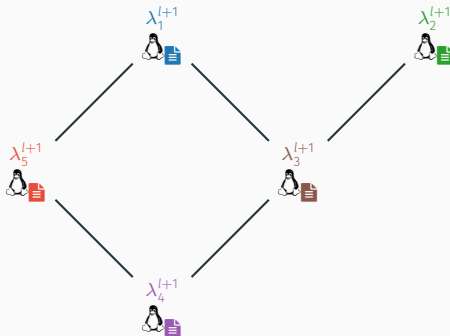


Após Sincronização,
cada nó i compartilha v_i^{t+1} com seus vizinhos



Para cada nó, faz-se

$$\lambda_i^{l+1} = \lambda_i^l + \frac{c}{2} \sum_{k \in \mathbb{K}_i} (v_i^{l+1} - v_k^{l+1})$$



Foram gerados 2500 dados bidimensionais de forma aleatória em uma distribuição gaussiana. Os testes foram realizados para uma rede de 40 nós.

| Risco | |
|---------------|--------|
| <i>DL SVM</i> | 0.0299 |
| SVM Central | 0.0301 |
| SVM Local | 0.0365 |

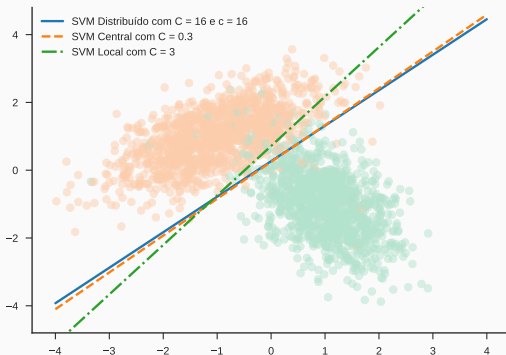


Figura 1: *DL SVM* comparado com o SVM centralizado após 400 iterações.

Procura-se averiguar se as soluções entre os agentes de fato concordam entre si. Para tanto, define-se a dispersão das soluções v em uma dada iteração l

$$\Delta^l(\bar{v}^l) = \frac{1}{N} \sum_{i=1}^N \|v_i^l - \bar{v}^l\|$$

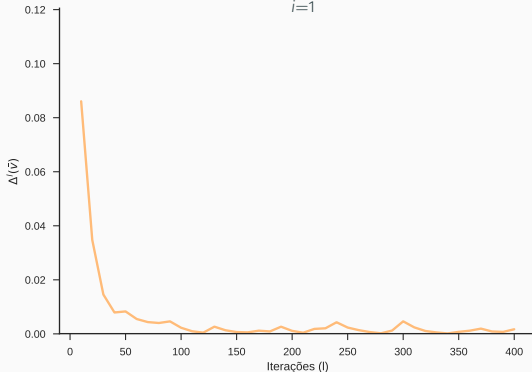


Figura 2: Dispersão medida entre as 400 primeiras iterações de *DLSVM*.

Descrição Dados oriundos da aplicação na análise para concessão de crédito de instituições financeiras de Taiwan, os quais são fornecidos pelo *Machine Learning Repository*.

Característica O conjunto de dados possui 30000 instâncias cada uma com 24 atributos classificadas por 1 (crédito aprovado) ou -1 (crédito não aprovado).

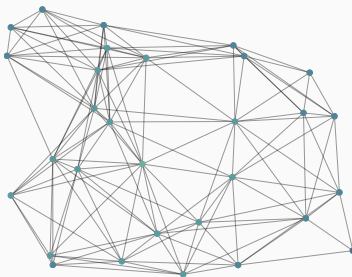


Figura 3: Rede de 30 nós utilizada para os testes com os dados reais.

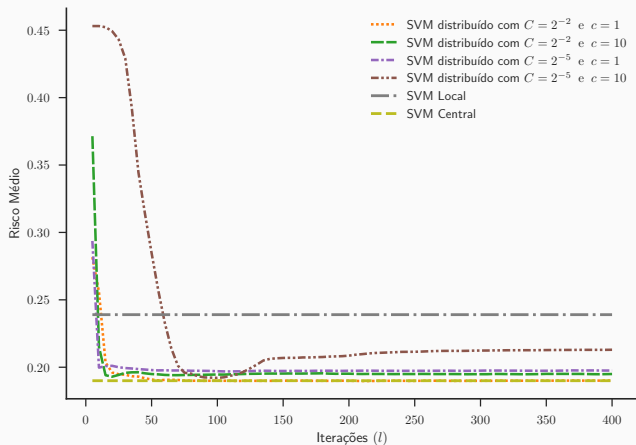
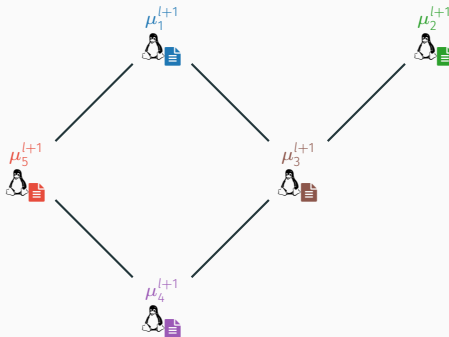


Figura 4: Risco a cada 5 iterações do DLSVM para cada conjunto de parâmetros, comparados com o SVM centralizado.

Para cada nó, faz-se

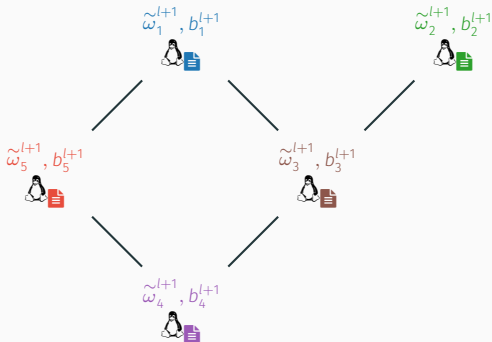
$$\mu_i^{l+1} \in \arg \max_{0 \leq \mu_i \leq C1_i} \left\{ -\frac{1}{2} \mu_i^T Y_i \left(K(X_i, X_i) - \tilde{K}(X_i, X_i) + \frac{1_i 1_i^T}{2c \# \mathbb{K}_i} \right) Y_i \mu_i \right. \\ \left. + \left[1_i + Y_i \left(K(X_i, \chi) - \tilde{K}(X_i, \chi) \right) r_i^l + \frac{s_i^l}{2c \# \mathbb{K}_i} Y_i 1_i \right]^T \mu_i \right\}$$



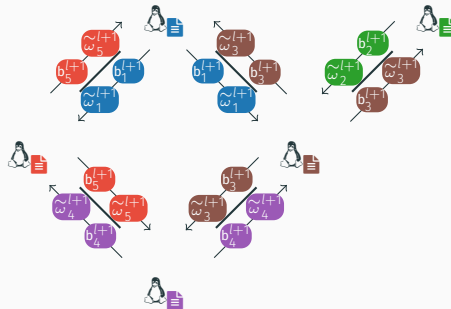
Para cada nó, faz-se

$$\tilde{\omega}_i^{l+1} = \left(K(\chi, X_i) - \tilde{K}(\chi, X_i) \right) Y_i \mu_i^{l+1} - \left(K(\chi, \chi) - \tilde{K}(\chi, \chi) \right) r_i^l$$

$$b_i^{l+1} = \frac{1}{2c\#\mathbb{K}_i} (1_i^T Y_i \mu_i^{l+1} - s_i^l)$$



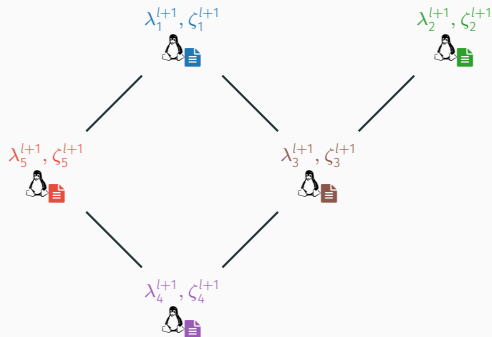
Após Sincronização,
cada nó i compartilha $(\tilde{\omega}_i, b_i^{l+1})$ com seus vizinhos.



Para cada nó, faz-se

$$\lambda_i^{l+1} = \lambda_i^l + \frac{c}{2} \sum_{i \in \mathbb{K}_i} (\tilde{\omega}_i^{l+1} - \tilde{\omega}_k^{l+1})$$

$$\zeta_i^{l+1} = \zeta_i^l + \frac{c}{2} \sum_{i \in \mathbb{K}_i} (b_i^{l+1} - b_k^{l+1})$$



Dados Comuns Cada coordenada t dos vetores $\{\chi_i\}_{i=1,\dots,p}$ é gerada pela distribuição uniforme em (x_t^{\min}, x_t^{\max}) com

$$x_t^{\min} = \min_{\substack{i=1,\dots,N \\ j=1,\dots,m_i}} \{[x_{ij}]_t\} \quad x_t^{\max} = \max_{\substack{i=1,\dots,N \\ j=1,\dots,m_i}} \{[x_{ij}]_t\}$$

Dados Artificiais Foram gerados 2560 dados bidimensionais com distribuição espacial simulando uma malha de xadrez.

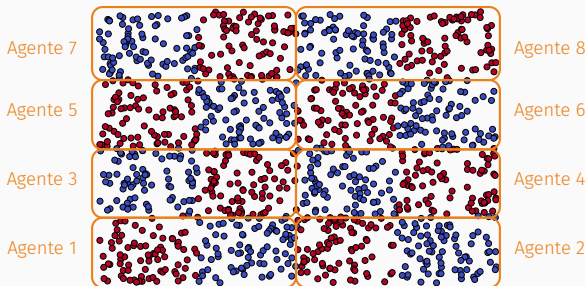
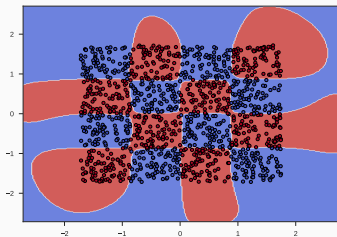
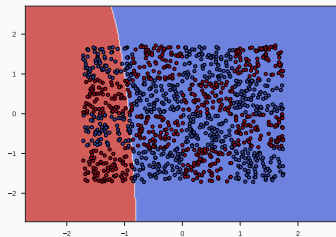


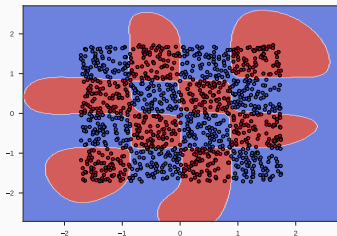
Figura 5: Distribuição Local dos dados para 8 agentes.



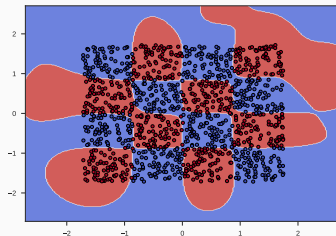
(a) SVM central.



(b) SVM local.



(c) Agente 1



(d) Agente 8

Figura 6: DNSVM com $p = 150$ após 800 iterações.

Descrição Dados oriundos da aplicação no diagnóstico de diabetes em população feminina indígena norte americana, os quais são fornecidos pelo *Machine Learning Repository*.

Característica O conjunto de dados possui 768 instâncias cada uma com 8 atributos classificadas por 1 (com diabetes) ou -1 (sem diabetes).

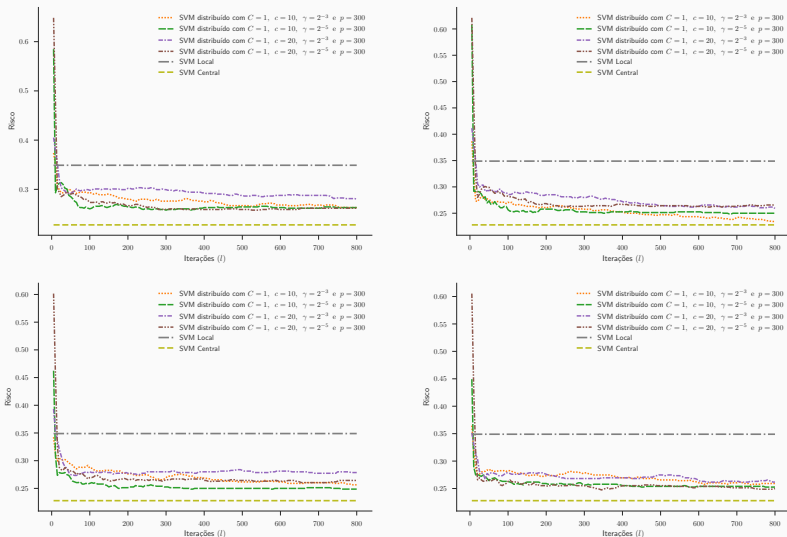
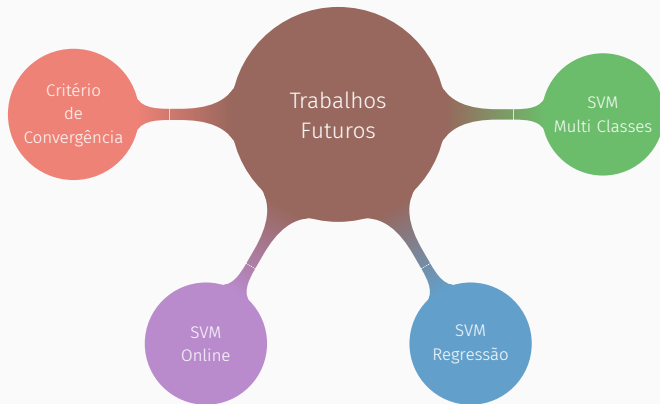







Figura 7: Risco a cada 5 iterações para cada conjunto de parâmetro do DNSVM para uma rede de 6 nós.



-  Dimitri P. Bertsekas. *Convex Optimization Algorithms*. 1ª ed. Massachusetts: Athena Scientific, 2015.
-  Dimitri P. Bertsekas. *Convex Optimization Theory*. Massachusetts: Athena Scientific, 2009.
-  Dimitri P. Bertsekas e John N. Tsitsikhs. *Parallel and Distributed Computation: Numerical Methods*. 2ª ed. Massachusetts: Athena Scientific, 1997.
-  Pedro A. Forero, Alfonso Cano e Georgios B. Giannakis. “Consensus-Based Distributed Support Vector Machines”. Em: *Journal of Machine Learning Research* 11 (2010), pp. 1663–1707.
-  Trevor Hastie, Robert Tibshirani e Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2ª ed. Springer, 2013.



David G. Luenberger e Yinyu ye. *Linear and Nonlinear Programming*. 4ª ed. Stanford University: Springer, 2016.



Bernhard Schölkopf e Alex Smola. *Learning with Kernels*. MIT Press, 2002.