

CML-Simplex: uma abordagem de programação linear para classificadores incrementais de margem larga

Juan Camilo Fonseca-Galindo¹, Luiz Carlos Bambirra Torres^{2*},
Gustavo Rodrigues Lacerda Silva³, Paulo Cirino Ribeiro Neto⁴, and
Antônio Pádua Braga⁵

Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brasil

¹ juankmilofg@ufmg.br, ² luizlitc@gmail.com,
³ gustavolacerdas@ufmg.br, ⁴ paulocirino.neto@gmail.com, ⁵ apbraga@ufmg.br

Resumo Este trabalho apresenta a proposta de um classificador incremental supervisionado, baseado na formulação geométrica da SVM, utilizando o método Simplex para solucionar problemas de programação linear modelados em cada uma das classes. São procurados pontos ótimos em cada um dos conjuntos de dados, estes pontos são utilizados para calcular o hiperplano de separação, o qual é utilizado como preditor da classe de uma nova amostra. Os resultados demonstraram que os métodos propostos obtiveram bons resultados, ainda tendo problemas de desbalanceamentos e superposição dos dados.

1 Introdução

As Máquinas de Vetores de Suporte ou *Support Vector Machines* (SVM)[3], são utilizadas para solucionar problemas de classificação de padrões e regressão. Este método tem sido amplamente utilizado na literatura [5,6,8,10]. De modo geral, dado uma base de dados de treinamento, a SVM calcula o hiperplano de separação que divide duas classes dentro de um espaço de n dimensões, este hiperplano, o qual é utilizado como preditor da classe de uma nova amostra.

Avanços recentes na tecnologia computacional, desenvolvimento de *software*, *links* de dados, sensores sofisticados e a miniaturização de *hardware* propiciaram uma grande desafio na implementação de algoritmos de aprendizagem *on-line* [1,9,11]. Métodos *on-line* devem ser capazes de tomar decisões em tempo real, além de poder-se adaptar quando novos dados são inseridos. Este artigo propõe um método de aprendizagem supervisionado, denominado CML-Simplex (Classificador de Margem Larga-Simplex), baseado no SVM geométrico [2], com o foco em problemas de classificação. A principal contribuição deste trabalho consiste na implementação de um classificador de margem larga formulado como

* Bolsista do CNPq-Brasil (N°150254/2016-4)

um problema de programação linear (PPL), ao contrário da formulação original da SVM que necessita de um método de Programação Quadrática. Para resolver o PPL é utilizado o método Simplex [7], na teoria este método tem um custo computacional exponencial, no entanto, na prática seu custo é polinomial, sendo uma grande vantagem em algoritmos *on-line*. Além disto, é proposto um algoritmo CML-Simplex incremental, este algoritmo aproveita os resultados obtidos pela solução inicial do Simplex, o que acarreta em um menor número de iterações para encontrar o ponto ótimo, diminuindo notavelmente seu custo computacional. O método proposto foi testado com bases de dados as quais são separáveis linearmente, balanceadas e desbalanceadas, e por fim bases de dados com sobreposição. Além disto, é detalhado como a abordagem proposta pode ser expandida para problemas de classificação não linearmente separáveis e de n -dimensões.

O restante do trabalho está organizado da seguinte forma, a seção 2 apresenta a metodologia proposta para o CML-Simplex e sua versão incremental. Na seção 3 os experimentos são apresentados. Finalmente, a seção 4 apresenta as conclusões e propostas de trabalhos futuros.

2 Metodologia

A metodologia proposta neste trabalho consiste em duas etapas: a primeira é baseada em um algoritmo de classificação utilizando programação linear CML-Simplex, já a segunda, é uma versão incremental do SVM Simplex.

2.1 CML-Simplex

O CML-Simplex propõe um PPL para calcular um hiperplano de margem larga para separar duas classes. Desta forma o método é composto pelas seguintes etapas:

1. Calcular conjuntos convexos: para cada classe presente na bases de dados é calculado um poliedro, este poliedro é a envoltória convexa das observações das classes. Um exemplo dos conjuntos convexos é apresentado na Figura 1a.
2. Restrições PPL: um poliedro convexo pode ser representado como uma interseção de retas (Figura 1b), desta forma, pode-se obter dois conjuntos de restrições, um para cada classe. Dados os pontos consecutivos da envoltória convexa $p = (x_{11}, x_{21})$ e $q = (x_{12}, x_{22})$, pode-se através da Equação 1 obter a Equação da reta representada pela linha verde da Figura 1b, onde m é a inclinação da reta (Equação 2) e b é o deslocamento da reta a partir da origem, como descrito na Equação 3.

$$m(x_{1i}) + (-1)(x_{2i}) + b = 0 \quad (1)$$

$$m = \frac{x_{11} - x_{12}}{x_{21} - x_{22}} \quad (2)$$

$$b = x_{2i} - mx_{1i} \quad (3)$$

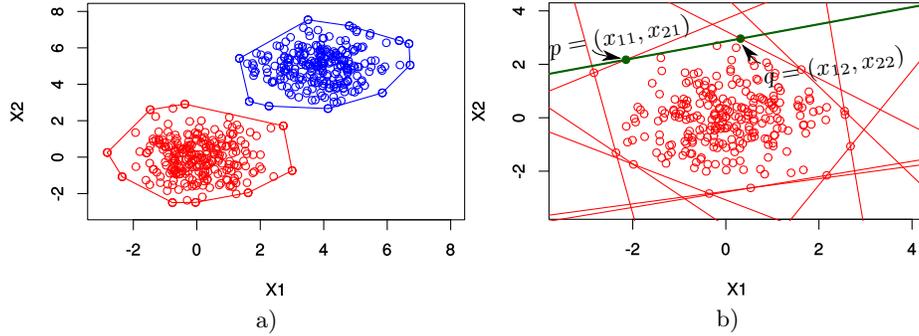


Figura 1: (a) Poliedros que representam a envoltória convexa de cada uma das classes. (b) Interseção de retas que geram um politopo.

Define-se o centroide de C_j pela Equação 4 para a classe j . Avalia-se a posição de C em cada reta (Equação 3) da classe j , com a finalidade de definir a matriz \mathbf{A}_j e a matriz \mathbf{b}_j das desigualdades das restrições da classe j . Deste modo as restrições são definidas pela Equação 5, onde \mathbf{x}_j é o vetor coluna das observações da classe j e n_j é a quantidade de observações da classe j .

$$C_j = \left(\frac{\sum_{i=1}^n x_{1i}}{n_j}, \frac{\sum_{i=1}^n x_{2i}}{n_j} \right) \quad (4)$$

$$\mathbf{A}_j \mathbf{x}_j \leq \mathbf{b}_j \quad (5)$$

3. Função objetivo: define-se como função objetivo, a função da reta ($F(\mathbf{x}) = \mathbf{c}\mathbf{x}_j$) na qual a inclinação é definida pela reta que une os centroides de cada um dos grupos. Dependendo da avaliação dos centroides na função objetivo, tem-se uma função de maximização e uma de minimização para cada um dos centroides (Equação 7).
4. Definição dos PPL: utilizando as matrizes das restrições e as funções objetivos obtidas nas etapas anteriores são definidos os PPL de cada uma das classes, isto é:

$$\text{PPL}_1 \quad \text{PPL}_2 \quad (6)$$

$$\text{Min } F(\mathbf{x}) = \mathbf{c}\mathbf{x}_1 \quad \text{Max } F(\mathbf{x}) = \mathbf{c}\mathbf{x}_2 \quad (7)$$

$$\mathbf{A}_1 \mathbf{x}_1 \leq \mathbf{b}_1 \quad \mathbf{A}_2 \mathbf{x}_2 \leq \mathbf{b}_2. \quad (8)$$

5. Implementação do Simplex: nesta etapa é implementado o método Simplex primal para cada PPL, para encontrar o ponto da classe 1 mais próximo da classe 2. Os pontos pretos da Figura 2a são os pontos ótimos o_1 e o_2 obtidos na implementação do primal nos PPL.

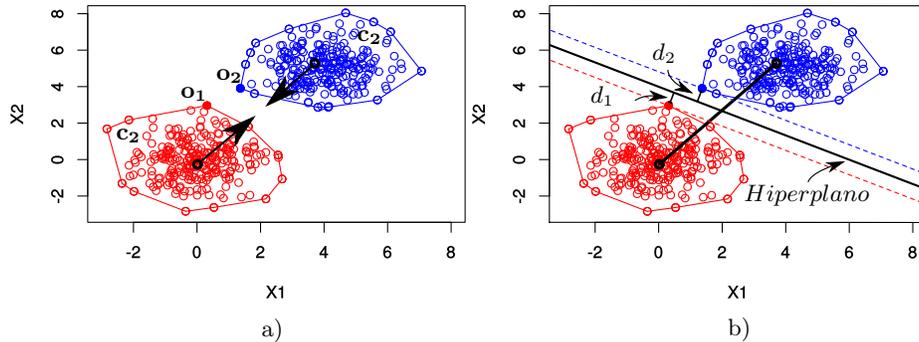


Figura 2: (a) Politopo, centroide e ponto ótimo para cada classe. (b) Hiperplano que divide as duas classes, este hiperplano minimiza a distância entre d_1 e d_2 .

6. Hiperplano de separação: define-se como hiperplano de separação em duas dimensões como a reta que divide os dois grupos. Desta maneira, é escolhida a reta perpendicular à reta que une os dois centroides a qual minimiza a distância perpendicular entre a reta e os dois pontos ótimos. Em outras palavras, é a reta que minimiza a distância d_1 e d_2 da Figura 2b. Sendo este hiperplano o preditor da classe para uma nova amostra.

O algoritmo explicado anteriormente foi implementado em uma base de dados de duas dimensões, com o objetivo de demonstrar graficamente o resultados da sua implementação. Porém, pode ser implementado em base de dados para n -dimensões, para isso, apenas se tem que declarar as restrições do polígono convexo e a função objetivo como hiperplanos de n dimensões, solucionar os PPL e gerar o hiperplano separador no espaço das n dimensões.

2.2 CML-Simplex incremental

Aproveitando as vantagens do método Simplex, foi proposto o método CML-Simplex incremental. Esta abordagem é apresentada no diagrama da Figura 3. Inicialmente são inseridos um conjunto de dados, é aplicado o método CML-Simplex ao conjunto de dados calculando assim o hiperplano de separação, são salvados C_j , c_j , o_j , A_j e b_j para cada PPL. Uma nova observação inserida é analisada em conjunto aos demais pontos da classe que corresponde, dessa forma, existem duas opções: a primeira é que o novo ponto inserido está contido na envoltória convexa, apresentado na Figura 4a, portanto as restrições não mudam, os centroides e as funções objetivos são atualizadas. Já se o novo ponto inserido não está contido na envoltória convexa, como é apresentado na Figura 4b, são retiradas as restrições que não fazem parte da nova envoltória convexa, neste caso r_1 e r_2 , e são acrescentadas as novas restrições criadas a partir do novo ponto inserido, r_3 e r_4 , além disto, são atualizados os centroides e as funções objetivos. Método Simplex é executado introduzindo como ponto inicial os pontos ótimos (o_j) obtidos anteriormente, conjunto com os valores de C_j , c_j , A_j e b_j , isto

reduz o custo computacional, já que só é analisada a viabilidade do ponto ótimo no novo PPL, e no máximo é executada uma nova iteração para encontrar o novo ponto ótimo. Finalmente é implementada a etapa 6 do CML-Simplex para calcular o hiperplano que separa as duas classes.

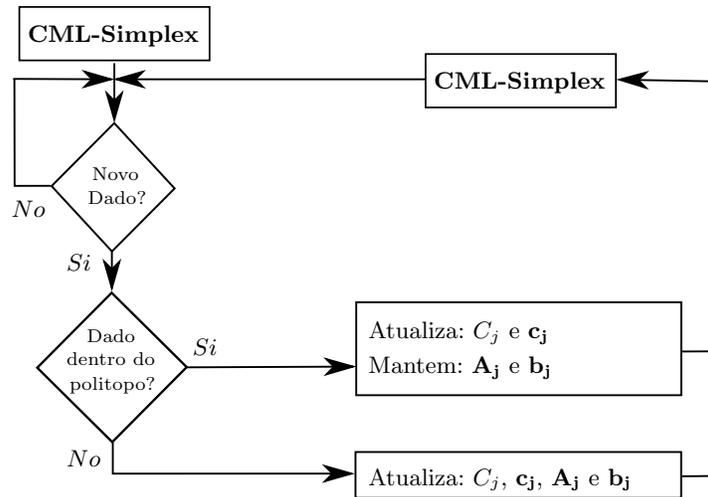


Figura 3: Diagrama do CML-Simplex incremental.

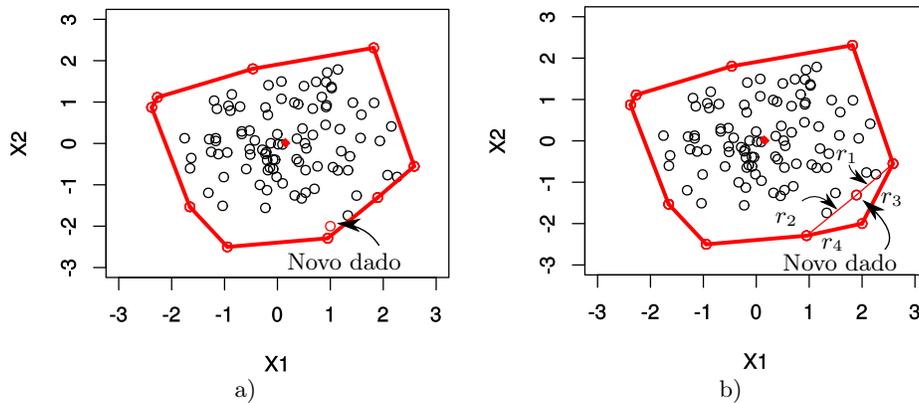


Figura 4: (a) O novo ponto inserido está contido na envoltória convexa. (b) O novo ponto inserido não está contido na envoltória convexa.

3 Experimentos e resultados

Os experimentos realizados foram divididos em três etapas, na primeira etapa foi avaliado o método CML-Simplex, na segunda o método CML-Simplex incremental e na terceira etapa foi apresentada a extensão dos métodos para problemas não linearmente separáveis.

3.1 CML-Simplex

Para avaliar o método CML-Simplex foram realizados 4 testes, com bases de dados com duas classes linearmente separáveis. As características das bases de dados e os resultados obtidos são descritos na Tabela 1. No primeiro teste, foi utilizada uma base de dados desbalanceada, onde a classe 1 tem o dobro da classe 2. O segundo teste apresenta uma base de dados balanceada, neste teste as variâncias das classes são diferentes mas a diferença não é significativa. No terceiro teste implementado, a base de dados tem classes balanceadas, mas a variância do \mathbf{x}_2 é três vezes maior que a variância no \mathbf{x}_1 para as duas classes. Finalmente, no último teste, a base de dados tem um alto grau de desbalanceamento, sendo 400 da classe 1 e 50 da classe 2, além disto, apresentam variâncias diferentes em cada uma das características \mathbf{x}_1 e \mathbf{x}_2 . Todas as bases de dados implementadas possuem uma pequena sobreposição de dados. Os resultados obtidos são apresentados nas Figuras 5a, 5b, 5c e 5d, respectivamente. Estes resultados demonstraram uma boa escolha do hiperplano de separação, mesmo quando a base de dados é desbalanceada e tem sobreposição.

Tabela 1: Testes CML-Simplex

Teste	1		2		3		4	
	1	2	1	2	1	2	1	2
Classe	1	2	1	2	1	2	1	2
Centroide	(0;0)	(5;0)	(0;0)	(4;4)	(1;1)	(6;1)	(0;4)	(4;0)
Variância	1;1	1;1	1;1	1,3;1,3	1;3	1;3	1;1	2;3
Nro. Pontos	200	100	100	100	100	100	400	50

3.2 CML-Simplex incremental

Para poder testar o CML-Simplex incremental, foram gerados dados a partir de uma distribuição normal e seguidamente inseridos pontos com o objetivo de observar o comportamento das restrições e dos pontos ótimos. Na primeira iteração foram geradas 6 observações para cada uma das classes. Com estes dados foram obtidos os resultados apresentados na Tabela 2 e na Figura 6a. Na Tabela 2, \mathbf{c} é o vetor da função objetivo \mathbf{cx} , a matriz \mathbf{A} e o vetor \mathbf{b} são as restrições e \mathbf{o} é o ponto ótimo de cada PPL. Na Figura 6a o PPL₁ é representado pela cor lilás e para o PPL₂ pela cor verde. Nesta iteração foram obtidas 5 restrições para cada um dos grupos.

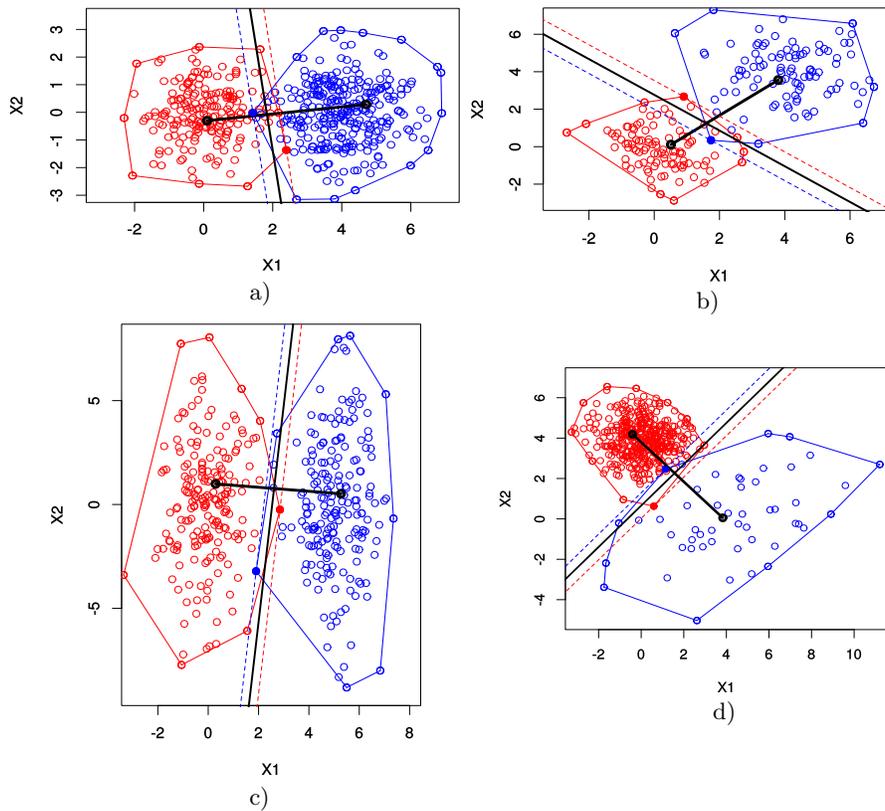


Figura 5: (a) Resultados obtidos para o teste 1. (b) Resultados obtidos para o teste 2. (c) Resultados obtidos para o teste 3. (d) Resultados obtidos para o teste 4.

Na segunda iteração foi inserida uma nova observação para cada uma das classes. O ponto inserido na classe de cor lilás mudou a envoltória convexa da solução anterior, sendo eliminadas a segunda e terceira restrição da primeira execução e inseridas mais duas novas restrições na mesma posição (Tabela 2). Coincidentemente, aconteceu o mesmo com a classe de cor verde, isto pode ser observado com as mudanças dos poliedro da Figura 6a para a Figura 6b.

Na terceira iteração foram inseridos 7 novos dados para cada um dos grupos. Os valores das restrições e funções objetivos são apresentadas na Tabela 2. Na Figura 6c, pode-se observar como os polítopos vão acrescentando-se quando novos valores são inseridos, as duas classes mantiveram 5 restrições. Neste passo toda as restrições da classe de cor lilás mudaram, mas na segunda classe pode ser visto que além de ter mudado as últimas três restrições o ponto ótimo foi o mesmo, isto demonstra que ao inserir o ponto ótimo da última execução como solução inicial se conseguiu diminuir o custo computacional.

Tabela 2: CML-Simplex Incremental

Iteração	1		2	
c	[0.6275175 1.0000000]		[0.6745162 1.0000000]	
PPL	1	2	1	2
Função Obj.	Max	Min	Max	Min
A	$\begin{bmatrix} 1.0913 & -1 \\ -0.1271 & 1 \\ 0.9396 & 1 \\ -2.5589 & -1 \\ -1.1106 & -1 \end{bmatrix}$	$\begin{bmatrix} -280.3525 & 1 \\ -1.6439 & -1 \\ -0.2955 & -1 \\ 1.6865 & 1 \\ 0.8080 & 1 \end{bmatrix}$	$\begin{bmatrix} 1.0913 & -1 \\ -0.6486 & 1 \\ 1.1982 & 1 \\ -2.5589 & -1 \\ -1.1106 & -1 \end{bmatrix}$	$\begin{bmatrix} -280.3525 & 1 \\ -2.0193 & -1 \\ 0.6276 & -1 \\ 1.6865 & 1 \\ 0.8080 & 1 \end{bmatrix}$
b	$\begin{bmatrix} 2.4904 \\ 0.3379 \\ 0.4100 \\ 1.3546 \\ 1.2334 \end{bmatrix}$	$\begin{bmatrix} -1008.7599 \\ -13.0261 \\ -1.3208 \\ 13.9892 \\ 9.8248 \end{bmatrix}$	$\begin{bmatrix} 2.4904 \\ 0.6665 \\ 0.7793 \\ 1.3546 \\ 1.2334 \end{bmatrix}$	$\begin{bmatrix} -1008.7599 \\ -13.0260 \\ -1.3208 \\ 13.9892 \\ 9.8248 \end{bmatrix}$
o	(0.0676 0.3465)	(3.9254 5.2143)	(0.0611 0.7061)	(4.4221 4.0961)

Iteração	1		2	
c	[0.7631306 1.0000000]		[0.8368211 1.0000000]	
PPL	1	2	1	2
Função Obj.	Max	Min	Max	Min
A	$\begin{bmatrix} -0.4833 & 1 \\ -0.6689 & -1 \\ 0.1652 & -1 \\ 6.0907 & 1 \\ 4.2117 & 1 \end{bmatrix}$	$\begin{bmatrix} -280.3525 & 1 \\ -2.0193 & -1 \\ 0.2741 & 1 \\ 2.2659 & -1 \\ 0.1078 & -1 \end{bmatrix}$	$\begin{bmatrix} 1.9817 & 1 \\ -0.8124 & -1 \\ 5.4441 & -1 \\ -1.0737 & 1 \\ -0.3210 & 1 \\ -0.2881 & -1 \\ -0.5462 & -1 \\ 0.0932 & -1 \\ 0.2999 & -1 \end{bmatrix}$	$\begin{bmatrix} -0.0825 & 1 \\ 0.1775 & -1 \\ 1.4656 & 1 \\ -61.0506 & 1 \\ -1.5691 & 1 \\ 3.3821 & -1 \\ 2.2137 & -1 \\ 0.8525 & 1 \\ 0.8854 & 1 \end{bmatrix}$
b	$\begin{bmatrix} 1.4750 \\ 1.4855 \\ 1.9617 \\ 8.3561 \\ 5.7670 \end{bmatrix}$	$\begin{bmatrix} -1008.7599 \\ -13.0261 \\ 7.8909 \\ 8.4570 \\ -3.6193 \end{bmatrix}$	$\begin{bmatrix} 4.7999 \\ 1.8542 \\ 10.6681 \\ 2.9921 \\ 2.0255 \\ 1.7029 \\ 1.5782 \\ 1.9205 \\ 2.1838 \end{bmatrix}$	$\begin{bmatrix} 6.5987 \\ -2.6059 \\ 15.5588 \\ -172.5442 \\ 1.2131 \\ 17.5222 \\ 9.8996 \\ 11.6343 \\ 11.8248 \end{bmatrix}$
o	(0.9142 1.9168)	(4.4221 4.0961)	(1.2049 2.4123)	(2.8772 3.1166)

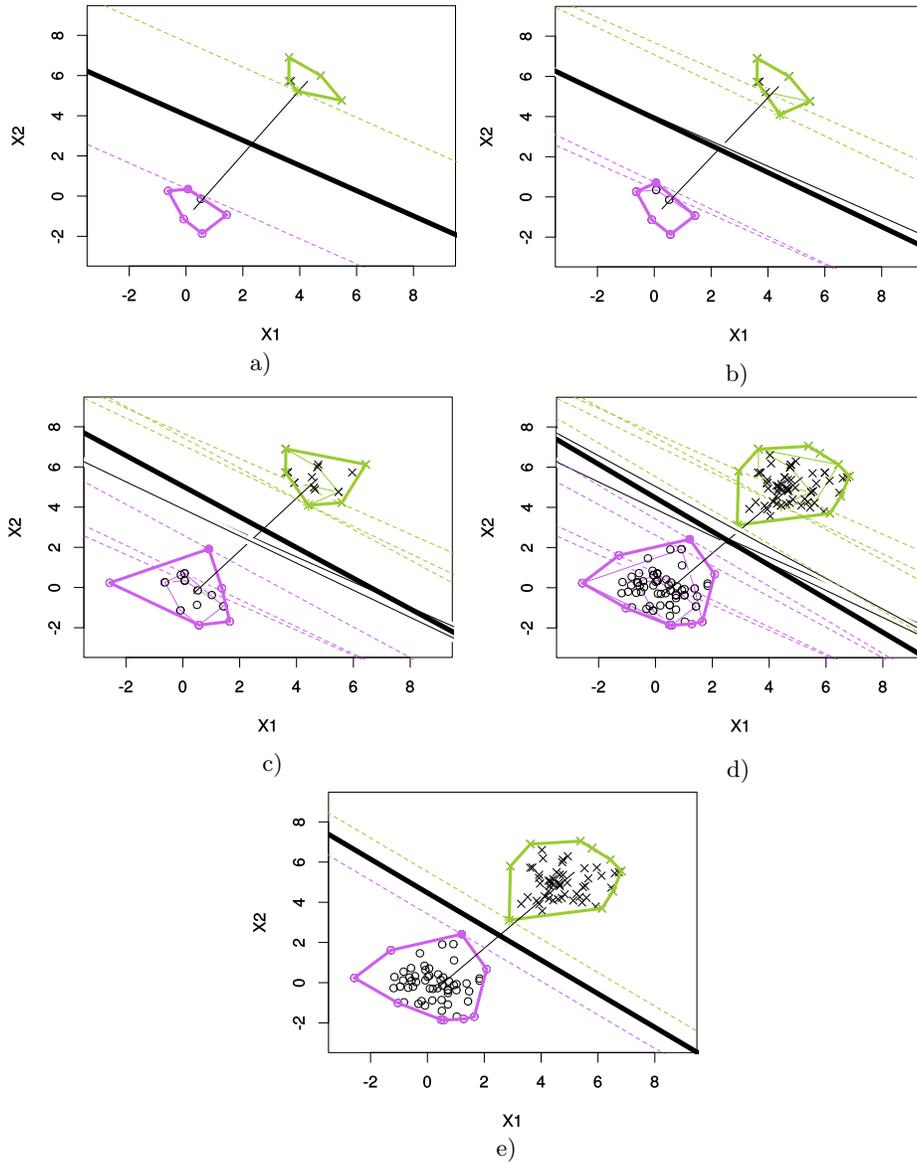


Figura 6: Primeira (a), segunda (b), terceira (c) e quarta (d) iteração do CML-Simplex incremental. (e) Quarta iteração do CML-Simplex incremental limpa.

Na quarta iteração foram inseridos 45 novos dados de forma sequencial, para um total de 60 observações em cada um dos grupos. Os valores das restrições e funções objetivos são apresentadas na Tabela 2. Na Figura 6d, pode-se observar os politopos vão acrescentando-se quando novos valores são inseridos, na classe

lilás mantiveram o mesmo número de restrições por enquanto, e a classe verde teve nove restrições. Na Figura 6e, observa-se a quarta iteração apagando as soluções anteriores.

3.3 Extensão do método para problemas não linearmente separáveis

Foi proposta uma abordagem para garantir a sua implementação em base de dados não linearmente separáveis. Para isto, implementa-se um método de agrupamento na base de dados para poder gerar grupos convexos separáveis linearmente e dessa forma calcular os hiperplanos de separação de todos contra todos. Para uma melhor interpretação é apresentada a Figura 7. Nesta Figura foi implementada a classificação para uma base de dados *Smiley* [4] de 800 observações (Figura 7a). Foram calculados 6 conjuntos de observações utilizando o método de agrupamento hierárquico *BIRCH* [12], onde três dos grupos tem o mesmo rótulo (Figura 7b). A Figura 7c mostra cada um dos hiperplanos de separação calculados para os grupos. Finalmente a Figura 7d apresenta as regiões de classificação obtidas pelo algoritmo.

4 Conclusões

Este trabalho apresentou um novo método para classificação que utiliza o método de programação linear Simplex para encontrar os pontos ótimos e seguidamente um hiperplano de margem larga. Além disto, foi apresentado uma variação do método para pode ser implementado como algoritmo incremental, esta variação aproveita a matriz de restrições e os pontos ótimos da solução anterior para diminuir o custo computacional podendo ser implantado como classificador *on-line*. Os resultados obtidos indicaram que o método conseguiu bons resultados mesmo que a base de tenha sobreposição, porém o método é sensível à presença de *outlier* nos dados, por isto se recomenda fazer etapa de pré-processamento nos dados para filtragem de *outliers*. O trabalho foi desenvolvido no *software R Project* para poder fazer uma análise gráfica do problema, foi utilizado o pacote de Rplex, e o algoritmo incremental foi desenvolvido em C++, utilizando a biblioteca Cplex.

Foi apresentada uma alternativa para utilizar o método em problemas não linearmente separáveis. Onde foi usado o método de agrupamento hierárquico *BIRCH*, porém pode-se implementar algum outro algoritmo de agrupamento. Como trabalhos futuros, visa-se escolher o algoritmo de agrupamento mais adequado para sua implementação em bases de dados não linearmente separáveis, além de realizar testes com bases de dados reais e comparar com métodos de classificação da literatura.

Agradecimentos

O presente trabalho foi realizado com o apoio financeiro da FAPEMIG, CNPq e CAPES - Brasil.

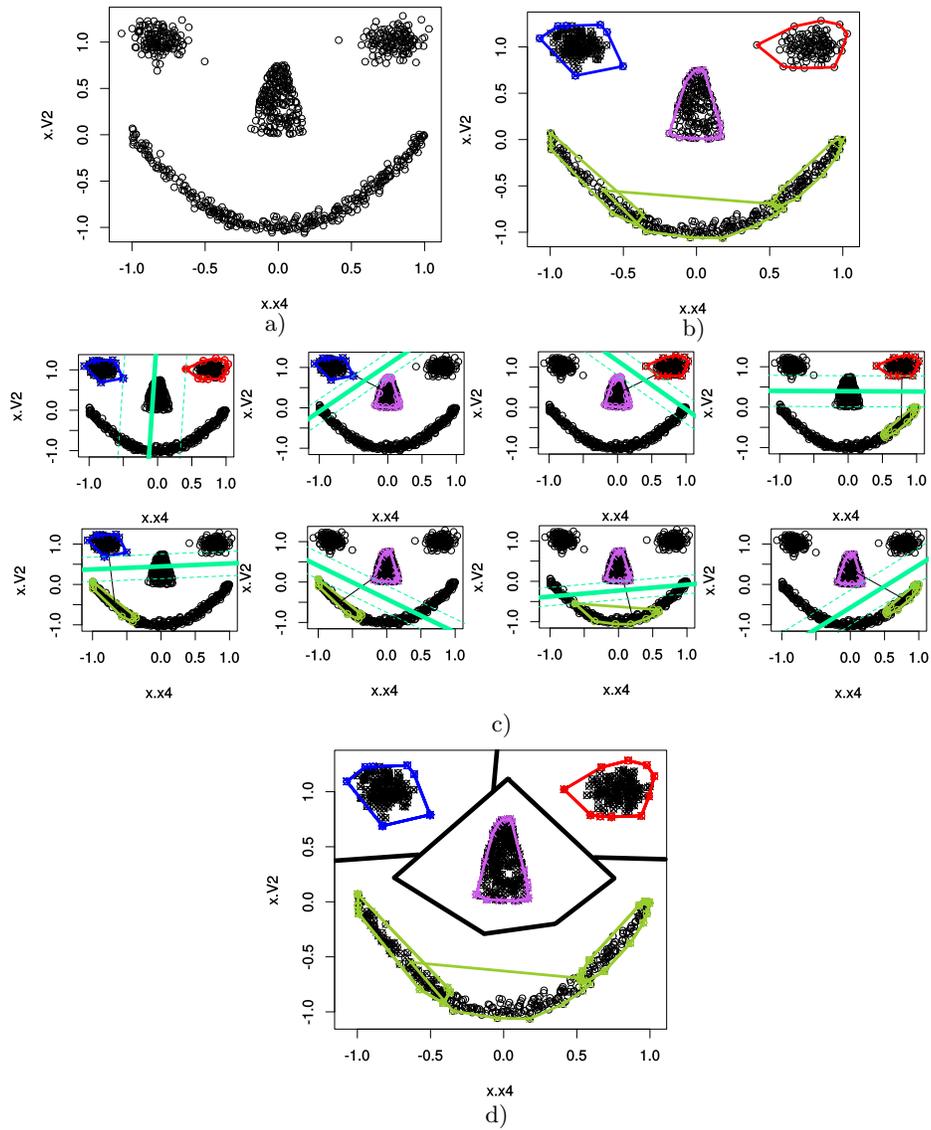


Figura 7: (a) Base de dados *Smiley*. (b) Agrupamento da base de dados com o método *BIRCH*. (c) Hiperplanos de separação entre os grupos. (d) Regiões de classificação da base de dados *Smiley*.

Referências

1. Abdelwahab, M., Busso, C.: Incremental adaptation using active learning for acoustic emotion recognition. In: International Conference on Acoustics, Speech and Signal Processing (2017)
2. Bennett, K.P., Breidensteiner, E.J.: Duality and geometry in svm classifiers. In: ICML. pp. 57–64 (2000)
3. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory. pp. 144–152. ACM (1992)
4. Leisch, F., Dimitriadou, E.: Machine learning benchmark problems (2010)
5. Lu, L., Ge, F., Zhao, Q., Yan, Y.: A svm-based audio event detection system. In: Electrical and Control Engineering (ICECE), 2010 International Conference on. pp. 292–295. IEEE (2010)
6. Madroñal, D., Lazcano, R., Fabelo, H., Ortega, S., Callicó, G.M., Juárez, E., Sanz, C.: Hyperspectral image classification using a parallel implementation of the linear svm on a massively parallel processor array (mppa) platform. In: Design and Architectures for Signal and Image Processing (DASIP), 2016 Conference on. pp. 154–160. IEEE (2016)
7. Nelder, J.A., Mead, R.: A simplex method for function minimization. The computer journal 7(4), 308–313 (1965)
8. Pravallika, P., Prasad, K.S.: Svm classification for fake biometric detection using image quality assessment: Application to iris, face and palm print. In: Inventive Computation Technologies (ICICT), International Conference on. vol. 1, pp. 1–6. IEEE (2016)
9. Strazzulla, I., Nowak, M., Controzzi, M., Cipriani, C., Castellini, C.: Online bimanual manipulation using surface electromyography and incremental learning. IEEE Transactions on Neural Systems and Rehabilitation Engineering 25(3), 227–234 (2017)
10. Xu, J., Tang, Y.Y., Zou, B., Xu, Z., Li, L., Lu, Y., Zhang, B.: The generalization ability of svm classification based on markov sampling. IEEE transactions on cybernetics 45(6), 1169–1179 (2015)
11. Zhang, J., Chen, L., Wang, C., Zhuo, L., Tian, Q., Liang, X.: Road recognition from remote sensing imagery using incremental learning. IEEE Transactions on Intelligent Transportation Systems (2017)
12. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: An efficient data clustering method for very large databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Montreal, Canada (1996)