Optimização e Algoritmos

Reconhecimento Facial

André Ribeiro João Lourenço José Iglesias
andre.g.ribeiro@tecnico.ulisboa.pt joao.n.lourenco@tecnico.ulisboa.pt
jose.p.iglesias@tecnico.ulisboa.pt
Instituto Superior Técnico

***Resumo*—Neste artigo é proposto resolver-se um problema de optimização de reconhecimento facial, no qual há imagens de tamanho 64x64 correspondentes a um alvo e um impostor específico. A resolução do projeto é baseado em Support Vector Machine (SVM) e o código desenvolvido para resolver o problema é suportado pelo MATLAB.**

O projeto está divido em duas fases. Primeiro utiliza-se um conjunto de treino de características desconhecidas para treinar uma SVM de Soft-Margin de classificação binária. A partir deste modelo chegou-se a níveis de classificações correctas para um conjunto de validação de 100% e foi possível concluir que o problema é linearmente separável. Na segunda fase propõe se diminuir a dimensão do problema substituindo a função de custo da SVM pela norma L1 da normal do hiperplano de fronteira de modo a se obter uma solução esparsa. Isto permite quer se analisem os píxeis mais relevantes para a classificação. Com isto reduziu-se o número de píxeis a analisar de 4096 para apenas 7 com uma percentagem de classificações corretas de 94.5% na validação.

I. Introdução

O processo de reconhecimento facial é bastante útil nos dias que correm uma vez tem uma infinidade de aplicações como é o caso dos sistemas de segurança em portáteis e smartphones que hoje em dia recorrem às suas câmaras para reconhecerem a cara do utilizador ou o uso de câmaras de vigilância para a procura de pessoas ou de matrículas de carros.

Existem vários algoritmos de reconhecimento facial, sendo os principais baseados em *Principal Component Analysis*, *Linear Discriminat Analysis*, *Hidden Markov Model*, entre outros, como se pode ver em [1]. Neste artigo expõem-se os métodos utilizados para desenvolver o sistema de reconhecimento facial baseado em Support Vector Machines (SVM).

Este artigo está dividido em 4 secções, sendo que na secção II faz-se uma breve introdução ao conceito de SVM e às variantes de Hard Margin e Soft Margin. Na secção III expõe-se o trabalho desenvolvido para a resolução da primeira fase, na qual se pretende arranjar um modelo que permita classificar um pessoa, a partir de uma imagem, como sendo a pessoa alvo ou a pessoa impostora. A classificação é feita através de uma SVM de Soft-Margin e é usado um conjunto de treina para a treinar e um diferente conjunto de validação para confirmar a validade do modelo encontrado. Na secção IV é abordada a segunda fase deste projeto, onde partindo dos principais resultados da primeira fase se pretende reduzir a dimensão do problema de classificação usando métodos que permitam tornar a solução do problema de optimização o mais esparsa possível. Estas duas secções estão divididas nas subsecções: A. Formulação do Problema, B. Abordagem, C. Resultados Numéricos e D. Análise dos Resultados. Para finalizar na secção V, expõem-se as conclusões em relação a todo o projeto.

II. Support Vector Machines

Tal como referido anteriormente, para se fazer o reconhecimento facial foi usada uma Support Vector Machine (SVM) linear. SVM é um modelo de aprendizagem supervisionada que permite, depois de treinada com um conjunto de treino, classificar novos dados. A breve explicação aqui dada é baseada em [2].

No caso das SVM lineares podem ocorrer que o conjunto de treino seja linearmente separável (Hard-Margin SVM) ou que não o seja (Soft-Margin SVM), utilizando-se diferentes abordagens para estas situações.

Supõe-se que se tem L pontos de treino, onde cada entrada $\mathbf{x}_i$ tem uma dimensão D e pertence a uma de duas classes y_i que tem o valor 1 ou -1, ou seja, uma classificação binária.

Relacionando com o nosso trabalho, diferentes classes são a pessoa alvo e a pessoa impostora enquanto que o conjunto de treino são as imagens correspondentes a cada uma das classes.

A. Hard-Margin SVM

Para o caso em que o conjunto de treino é linearmente separável é possível encontrar um hiperplano definido por $\mathbf{w} \cdot \mathbf{x} + b = 0$, onde $\mathbf{w}$ é a normal do hiperplano e $\frac{b}{\|\mathbf{w}\|}$ corresponde à distância perpendicular do hiperplano à origem, que separem espacialmente as duas classes. Para melhor compreensão a Figura 1 mostra uma representação de um SVM de Hard-Margin.

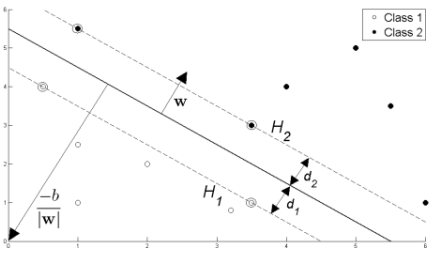


Figura 1: Hard-Margin SVM.

Os denominados vectores de suporte (support vectors) são os pontos mais próximos do hiperplano de cada uma das

classes sendo que o objectivo do modelo SVM é encontrar os parâmetros $\mathbf{w}$ e b que maximizem a distância do hiperplano a estes pontos. É possível descrever o nosso conjunto de treino através da equação:

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \ \forall i. \tag{1}$$

Para os vectores de suporte determinam-se dois novos hiperplanos definidos pelas equações (2) e (3).

$$\mathbf{x}_i \cdot \mathbf{w} + b = +1, \text{ para } y_i = +1, \tag{2}$$

$$\mathbf{x}_i \cdot \mathbf{w} + b = -1, \text{ para } y_i = -1. \tag{3}$$

A distância destes dois hiperplanos ao hiperplano fronteira, denominada margem, é igual e dada por $\frac{1}{\|\mathbf{w}\|}$. Assim, concluiu-se que maximizar a margem é equivalente a minimizar a norma euclideana de $\mathbf{w}$.

B. Soft-Margin SVM

No caso em que o conjunto de treino não é linearmente separável não é possível encontrar um hiperplano que separe espacialmente as duas classes. O que se faz nestes casos é introduzir no modelo uma tolerância a erros, isto é, a classificações erradas. Essa tolerância é implementada introduzindo em (1) slack variables não negativas ξ_i, $i = 1, ..., L$, resultando:

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0, \text{ onde } \xi_i \geq 0 \ \forall i. \tag{4}$$

As variáveis ξ_i, tal como $\mathbf{w}$ e b, são desconhecidas e é a partir delas que vai permitir obter a equação do hiperplano que melhor separa as duas classes ainda que com alguns erros de classificação.

A Figura 2 mostra um representação deste tipo de SVM.

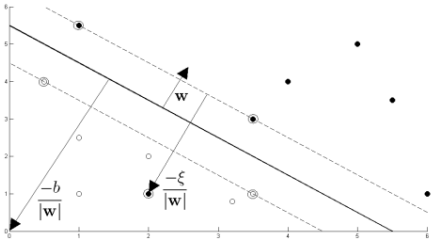


Figura 2: Soft-Margin SVM.

C. Classificação

Em ambos os casos a classificação é feita submetendo os padrões não classificados à equação do hiperplano cujos parâmetros maximizam as margens, isto é, a distância da hiperplano, ou fronteira, aos vetores de suporte. A classificação é calculada da forma

$$y_i = sign(\mathbf{x}_i \cdot \mathbf{w} + b). \tag{5}$$

É a partir deste método de classificação que se vai determinar se os parâmentros encontrados para o nosso hiperplano são adequados, testando um conjunto de validação, cuja classificação é conhecida, e contabilizando os erros de classificação. Isto aplica-se principalmente ao caso da SVM com Soft-Margin para determinar qual o valor da variável de ajuste C que causa menos erros no conjunto de validação.

III. Primeira Fase

Nesta fase desenvolve-se um detector facial tendo em consideração conjuntos de imagens da pessoa alvo e da pessoa impostora.

A. Formulação do Problema

Sem nenhuma informação sobre a linearidade do problema o mais acertado é uma abordagem baseada no modelo SVM de Soft-Margin em que se tem formular um problema de optimização com base na teoria por trás deste modelo.

Como já referido anteriormente, pretende-se minimizar a norma do hiperplano fronteira dando uma tolerância a erros de classificação que possam ocorrer. Isto significa que a função de custo a utilizar tem de ter em conta estes dois fatores sendo que que se introduz uma variável de ajuste C, independente do problema, que serve para definir a importância que se dá a cada um deles. Um valor de C elevado torna os erros de classificação muito dispendiosos na função de custo pelo que o modelo vai tender a evitá-los ao máximos aproximando-se do comportamento da SVM com Hard-Margin. Por outro lado um valor de C reduzido torna os erros baratos permitindo que estes aconteçam em grande número. Nenhumas destas situações é desejável pelo que é necessário encontrar um meio termo para o valor de C.

Deste modo, o problema de optimização que se pretende resolver assume a forma

$$\begin{aligned} \underset{\mathbf{w},b,\xi}{\text{minimize}} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{L}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0 \ \forall i \\ & \xi_i \geq 0 \ \forall i \end{aligned} \tag{6}$$

onde $\mathbf{w}$, b e ξ são as variáveis de optimização a descobrir. Note-se que minimizar a norma de $\mathbf{w}$ equivale a minimizar a mesma norma ao quadrado, um vez que a posição do minimo não se altera. Com isto ganha-se o facto de a função de custo (6) se tornar mais simples. Por último referir que a função de custo é convexa uma vez que é a soma de duas funções convexas (o quadrado de uma norma e uma função afim).

B. Abordagem

Este projeto é realizado em MATLAB devido à facilidade inerente aos cálculos com matrizes e uma vez que é necessário utilizar a extensão do MATLAB, CVX, que serve para obter parâmetros de condições convexas.

Utiliza-se um dataset do Instituto Superior Técnico (IST), fornecido pelo corpo docente com características específicas. A dataset é composta por uma conjunto de treino que contém 83 imagens relativas ao impostor e com 101 imagens do alvo e um conjunto de validação com 84 imagens do impostor e 97 imagens do alvo. Os dados analisados são fotografias, todas do mesmo tamanho (64×64 pixeis), as quais estão centradas de forma a que os olhos, o nariz e a boca fiquem aproximadamente nos mesmos sítios, como se verifica na Figura 3.

Para resolver este problema é necessário agrupar todas as imagens em vetores de imagens $\mathbf{x}_i \in \mathcal{R}^{4096}$. Uma vez que

(a) Impostor (b) Alvo

Figura 3: Exemplo do tipo de imagens analisadas.

as imagens correspondem a matrizes de 64 por 64 pixeis é necessário transformar as matriz de cada imagem em vetor e após isso concatenar todos estes vetores numa matriz de dimensão $N \times M$ sendo que N corresponde ao número total de pixeis (4096) e M ao número total de imagens. É necessário criar um vetor de classificação y que tem as classificações correspondentes a cada uma das imagens, sendo $y_i = +1$ para as imagens da pessoa alvo e $y_i = -1$ para a pessoa impostora.

Após este processo de carregamento e tratamento de imagens procede-se ao cálculo dos parâmetros $\mathbf{w}$, b e ξ utilizando o CVX de forma a minimizar a condição (6). Sendo que é necessário definir o tamanho de cada variável previamente sendo que $\mathbf{w}$ é o vetor de dimensão 4096 uma vez que é o número de píxeis de cada imagem. A variável b é um escalar. A variável ξ vai ser do tamanho do número de imagens.

Com os parâmetros que o CVX retorna, nomeadamente $\mathbf{w}$ e b, procede-se à validação. Para este processo utiliza-se o conjunto de validação de forma a escolher o valor de C ideal. A classificação das imagens de validação é efetuada como referido em (5). O critério para determinar a validade do modelo é a percentagem de classificações corretas no conjunto de validação, considerando-se um erro de classificação quando a determinada pela SVM difere da classificação real.

C. Resultados Numéricos

Como foi dito anteriormente, através do conjunto de treino obteve-se os parâmetros $\mathbf{w}$, b e ξ que são necessários para avaliar os dados do conjunto de validação. Para analisar os resultados obtidos, variou-se o valor de C numa gama de valores de 10^{-5} até 10^{8} em potências de 10 de forma a se variar o peso dos erros de forma significativa em cada teste.

Os resultados númericos obtidos estão demonstrados na Figura 4.

D. Análise dos Resultados

Com base nos resultados anteriormente apresentados verifica-se , como era expectável, uma grande importância no valor do parâmetro C pois o peso desse parâmetro vai ter influência direta no número de classificações certas e erradas das imagens analisadas.

Para valores de C muito pequenos o modelo vai permitir valores de ξ elevados o que significa que vai permiter que ocorram erros muito elevados e, provavelmente, em grande quantidade. É exatamente isso que se pode observar para $C < 10^{-2}$. Numa gama de valores de C mais altos, os erros já começa a ter muito peso na função de custo pelo que vai fazer com que os valores de ξ tendam a ser muito pequenos, ou seja, o modelo vai rejeitar erros de classificação. Quando C

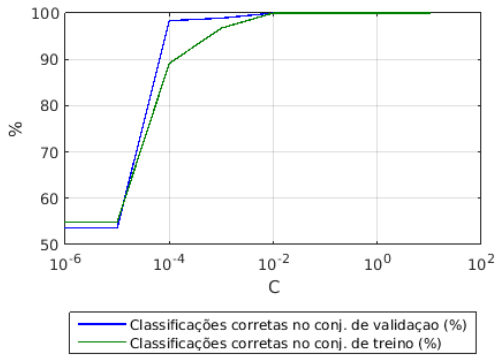


Figura 4: Percentagem de classificações corretas nos conjuntos de validação e de treino para diferentes valores de C.

tende para infinito todos os erros são rejeitados e a SVM passa a comportar-se como uma Hard-Margin SVM. Assim sendo, para $C > 10^{-2}$ já não há erros de classificação, nem para os dados de treino nem para os dados de validação, ou seja, há uma correspôndecia total das imagens e pode-se concluir que a dataset utilizada é linearmente separável.

IV. Segunda Fase

Os resultados da primeira fase do projecto levaram-nos a concluir que o nosso conjunto de treino é linearmente separável, o que implica que se pode aplicar um modelo Hard Margin, fazendo com que C e ξ possam ser retirados do problema de optimização.

A segunda fase do trabalho consiste em tornar a solução $\mathbf{w}$ do problema de optimização (6) o mais esparsa possível, ou seja, com o maior número de valor (aproximadamente) aproximadamente que as restrições permitem. Considerando o nosso problema, isto implica que se pretende que muitos dos píxeis de cada imagem sejam multiplicados por um peso (aproximadamente) nulo, tendo apenas relevância para a classificação da imagem os píxeis com informação real importância que permitam distinguir as duas pessoas.

A. Formulação do Problema

Tal como já referido no parágrafo anterior, tendo em conta o facto de se estar a trabalhar com um conjunto de treino linearmente separável é possível adoptar o modelo Hard Margin da SVM. Isto é, pretende-se minimizar a norma de $\mathbf{w}$, considerando a restrição (1). De forma a se obter um vector $\mathbf{w}$ com o máximo de componentes (aproximadamente) nulas vai-se minimizar a norma L1 de $\mathbf{w}$ mantendo a restrição (1). Deste modo a optimização vai tender a minimizar cada componente de $\mathbf{w}$ individualmente, aproximando-as o mais possível do valor nulo sempre que permitido, como explicado em [3].

Note-se também que se deixou de minimizar a norma euclideana de $\mathbf{w}$ para se passar a minimizar a norma L1. Obviamente a solução deixa de ser a mesma e deixa-se de se maximizar a margem em relação à fronteira objectivamente, mas tendo em conta a separabilidade do nosso conjunto de treino a nova solução para o problema acabará por separar

convenientemente os dados, ainda que não como a margem óptima. Esta função de custo é também convexa visto tratar-se de uma norma.

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & \|\mathbf{w}\|_1 \\ \text{subject to} \quad & y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \ \forall i \end{aligned} \tag{7}$$

B. Abordagem

A abordagem para a segunda parte do projeto foi análoga ao desccrito para a primeira fase no que toca ao tratamento do conjunto de dados, sendo que a única diferença é a já referida reformulação do problema de optimização (7).

Após obtidos os parâmetros do hiperplano que minimizam o problema a normal $\mathbf{w}$ terá uma grande quantidade de componentes com valor residual, isto é, muito próximos de zero, sendo que todas as outras componentes serão as que têm real preponderância na classificação. Para se considerar apenas estas últimas aplica-se $\mathbf{w}$ uma filtragem de forma a anular as componentes com valor aproximadamente nulo. Essa filtragem é implementada através de um threshold positivo que anulas todas as componentes de $\mathbf{w}$ com módulo inferior ao seu valor sendo que o valor ideal para este threshold é encontrado realizando vários testes em que se aumenta progressivamente o valor do threshold.

C. Resultados Numéricos

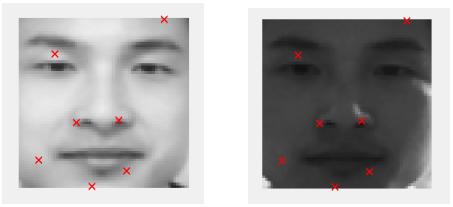
Foram realizados 18 testes para diferentes valores de threshold, variando de 10^{-14} até 1 em potências de 10 e a partir dai com incrementos unitários até 4. A razão pela qual se começou a efectuar incrementos unitários a partir de 1 foi para melhor mostrar a variação da percentagem de classificações correctas nos conjuntos de treino e de validação. O valor inicial de 10^{-14} está relacionado de ser este, aproximadamente, o valor do erro numérico do MATLAB. A Tabela I é ilustrativa dos resultados obtidos para os testes realizados com que forneceram dados mais relevantes.

Tabela I: Resultados dos testes para diferentes valores de threshold.

Threshold	10^{-14}	10^{-11}	10^{-10}	10^{-2}	1	3
Componentes de $\mathbf{w}$ não anuladas	4096	3364	413	14	7	1
Acertos no conj. de treino(%)	100%	100%	100%	100%	95.7%	45.1%
Acertos no conj. de validação (%)	98.9%	98,9%	98,9%	98,9%	94.5%	46.4%

D. Análise dos Resultados

A partir dos resultados obtidos para os vários teste conclui-se que é possível reduzir bastante as componentes não nulas de $\mathbf{w}$ e ainda assim conseguir percentagens elevadas de acertos. Para thresholds de 0.01 e 1 foram obtidos acertos de 98.9% e 94.5% no conjunto de validação para apenas 14 (0.34%) e 7 (0.17%) componentes não nulas de de $\mathbf{w}$. Isto apenas é possível devido às caracteristicas muito específicas do conjunto de imagens utilizadas, como já referido anteriormente. Na Figura 5 mostra-se a localização dos 7 píxeis usados na classificação quando se aplica um threshold de 1. Destaca-se a distribuição desses píxeis em zonas específicas das caras como por exemplo nariz, sobrancelha e boca.



(a) Impostor (b) Alvo

Figura 5: Localização dos 7 píxeis relevantes para a classificação quando se aplica um threshold de 1.

V. Conclusões

Neste trabalho foi possível explorar com algum detalhe a forma como um simples problema de optimização consegue ter uma aplicação tão interessante como reconhecimento facial. Para a primeira fase do projeto realizou-se em MATLAB um programa que permite obter um modelo SVM de Soft-Margin que permite classificar imagens de tamanho 64x64 como sendo referentes a uma pessoa alvo ou a uma pessoa impostora, cujas imagens pertencem a uma dataset fornecida pelo corpo docente. Com esta abordagem foi possivel obter 100% de classificações corretas e ainda concluir que o nosso dataset era na verdade linearmente separável. Na segunda fase deseja-se aproveitar ao máximo as caracteristicas do nosso dataset e reduzir a dimensão do nosso problema analisando apenas os píxeis mais relavantes para classificação, ou seja, tornar a solução do problema de optimização esparsa. Isto foi conseguido substituindo a função de custo de uma SVM de Hard-Margin pela norma L1 da normal do hiperplano de separação das duas classes tendo-se conseguido realizar a classificação com 98.9% de acertos usando apenas 7 píxeis das imagens.

Podiam-se ter optado por outras abordagens como por exemplo, na segunda fase, recalcular os parâmetros do hiperplano usando apenas os 7 píxeis mais relevantes ou mesmo alterar a função de custo de forma a que esta fosse composta pela soma da norma L1 e da norma L2 da normal do hiperplano de separação de forma a se ter um compromisso entre esparsidade e margem maximizada, mas os resultados obtidos para a abordagem escolhida foram tão satisfatórios que isso não se justificou. Para problemas de maior dimensão a norma L1 pode também tornar-se bastante lenta de minimizar comparativamente à norma euclideana devido à não diferenciabilidade na origem, algo que para o nosso trabalho não foi problemático.

Referências

[1] Wikipedia, *Facial recognition system*.
[2] Tristan Fletcher, *Support Vector Machines Explained*, 2009.
[3] Mark Schmidt, *Least Squares Optimization with L1-Norm Regularization*, 2005.