PROJETO COMPUTACIONAL 2

PCA & Sistemas de Recomendações

- Nicolas Toledo de Camargo RA: 242524
- Vinícius Oliveira da Silva RA: 195068
- Luana Aparecida Praxedes da Silva RA: 182255

O código foi desenvolvido na linguagem python e foi enviado junto com este relatório para teste.
Utilizamos as bibliotecas : numpy, scipy, matplotlib, e pandas.

Este projeto é dividido em duas partes:
Parte I: reconhecimento facial usando PCA;
Parte II: sistema de recomendação de filmes por filtragem colaborativa.

Parte I

Quando se tem muitas features para se tratar em algum problema de machine learning, podemos tentar redimensionar o problema para um com menos atributos para reduzirmos o custo computacional e ruídos, onde esses novos atributos tenham sido feitos relacionando os atributos iniciais.

Um método que pode ser utilizado para esta compressão é a análise das componentes principais (PCA), e é este que vamos utilizar para redimensionar imagens que representam faces, visando seu reconhecimento.

Partimos de um conjunto de dados com 5 mil faces, em que cada linha representa uma imagem 32 x 32 vetorizada. Se dimensionalizar cada uma dessas linhas para uma matriz 32 x 32, podemos mostrar cada face.

Imagem 1: representação de 100 faces aleatórias dos dados.

Para esse caso de imagens de faces usar o método de análise de componentes principais é interessante pois ele acaba funcionando muito bem, foi até dado um nome específico para essas componentes principais: eigenfaces.

Implementamos o algoritmo como na teoria vista em aula para realizar o PCA e rodamos para nossos dados. Podemos então ver as imagens dos componentes principais obtidos:

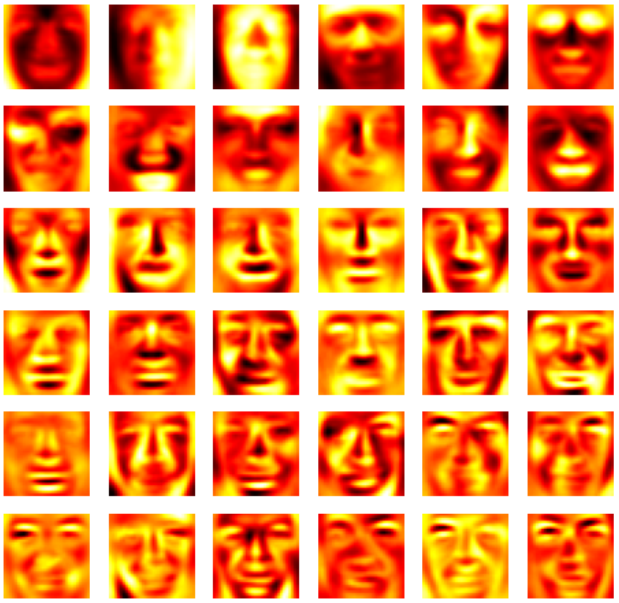

Imagem 2: eigenfaces dos 36 primeiros componentes principais.

Aqui já podemos notar por que este método é interessante para reconhecimento facial, suas componentes principais nos traz as informações gerais mais importantes para

cada rosto. A informação da silhueta/características chave dos rostos permanece, como a localização, tamanho e formato dos olhos, nariz e boca e o formato do próprio rosto, reduzindo apenas detalhes que podem ser considerados como “ruídos” para seu reconhecimento.

Fizemos então a projeção dos dados iniciais nas 100 primeiras componentes principais obtidas e realizamos o processo para reconstrução dos dados. Com esses dados reconstruídos, plotamos as imagens referentes a eles e as imagens dos dados originais.

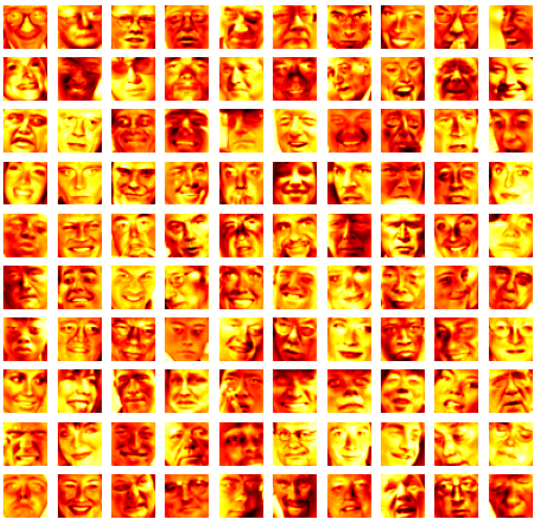

Imagem 3: Imagens originais.

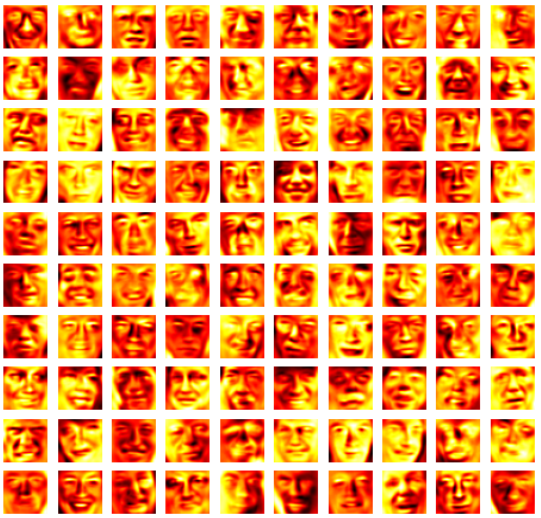

Imagem 4: Imagens reconstruídas.

Percebemos que as imagens reconstruídas ficaram muito boas no quesito de semelhança com as originais, houve perda apenas de pequenos detalhes como alguns traços faciais mais sutis, mas em geral é possível reconhecer e diferenciar cada rosto.

Este exercício nos leva a crer que ao trabalhar com reconhecimento facial, usar PCA para redução de dimensionalidade antes de usarmos os dados em algum outro algoritmo é uma ótima ideia, já que reduzir a dimensionalidade irá reduzir o custo computacional nos tratamentos intermediários do trabalho em questão (algum algoritmo supervisionado que que requer treinamento por exemplo) e ainda podendo facilitar o reconhecimento dos rostos ao reduzir os detalhes/ruídos das faces. Além disso, essa redução não prejudica a reconstrução da informação se for necessária.

Parte II

Para a segunda parte do trabalho, iremos fazer um sistema de recomendação de filmes usando filtragem colaborativa que prevê a nota que cada usuário daria para certo filme. Usamos como base o código apresentado em aula.

Temos a disposição um arquivo com notas de 1 a 5 de cada usuário para cada filme, em que cada linha representa um filme e cada coluna um usuário, o arquivo possui 1682 filmes e 943 usuários. Temos também uma matriz binária que apresenta 1's se o usuário da coluna j deu nota para o filme da linha i, e zero caso contrário. Por fim, um arquivo de texto indicando a ordem e nome dos filmes.

É definida uma função que ao receber os parâmetros, a matriz das notas, a matriz binária, o número de features, filmes e usuários, retorna as funções custo e gradiente usadas para a filtragem colaborativa, como visto nas teorias de aula.

Como na filtragem colaborativa queremos aprender tanto os features de cada filme como os pesos deles para cada usuário, inicializamos tanto os features como os pesos aleatoriamente em com relação a uma distribuição normal padrão. Uma lista com todos esses parâmetros é o que a função do custo e gradiente irá receber.

Se um usuário não tiver dado nenhuma nota, o algoritmo irá aprender para prever nota zero em todos os filmes para esse usuário, isso é um problema pois não é algo realista. Para contornar isso, usamos a técnica de tirar a média das notas de cada filme de todas as notas na matriz das notas dos usuários; então a previsão será a nota prevista menos a média do filme, portanto somamos essas médias ao fim para ter a previsão correta.

Para o treino dos parâmetros, usamos a função scipy.optimize.minimize com o método do gradiente conjugado.

Finalmente realizamos o treino fornecendo a função custo e gradiente para o método de otimização, juntamente com a matriz das notas já descontadas as médias de cada filme e os parâmetros iniciados aleatoriamente.

Com os parâmetros aprendidos, podemos então realizar as predições e somar as médias obtidas inicialmente para ter a nota correta. Dessas notas previstas, tiramos a média de cada filme, organizamos em uma lista e tabelamos para a ordem decrescente das notas, indicando sobre qual filme a nota é referente.

```
10 filmes com notas médias mais altas:

Nota predita 5.000000073076413  para o filme: 1599 Someone Else's America (1995)
Nota predita 5.000000061566244  para o filme: 1189 Prefontaine (1997)
Nota predita 5.000000042828005  para o filme: 1536 Aiqing wansui (1994)
Nota predita 5.000000030159169  para o filme: 1201 Marlene Dietrich: Shadow and Light (1996)
Nota predita 5.000000018869574  para o filme: 1467 Saint of Fort Washington, The (1993)
Nota predita 5.000000005390686  para o filme: 1293 Star Kid (1997)
Nota predita 4.999999987841885  para o filme: 1653 Entertaining Angels: The Dorothy Day Story
(1996)
Nota predita 4.999999975890671  para o filme: 1122 They Made Me a Criminal (1939)
Nota predita 4.999999960428242  para o filme: 814 Great Day in Harlem, A (1994)
Nota predita 4.999999940786569  para o filme: 1500 Santa with Muscles (1996)
```

Imagem 5: 10 filmes com maiores notas previstas. (sem arredondamento)

Imagem 6: 10 filmes com maiores notas previstas. (com arredondamento)

Referências:

[1]: Slides e Apostila de Machine Learning - Prof. João Batista Florindo. (https://github.com/jbflorindo/MS-MT571)