

Predição de imagens da família “Os Simpsons”

Leonardo Pontes Baiser¹, Marco Cezar Moreira de Mattos¹,
Rômulo Manciola Meloca¹

¹DACOM – Universidade Tecnológica Federal do Paraná (UTFPR)
Caixa Postal 271 – 87301-899 – Campo Mourão – PR – Brazil

{lpbaiser, marco.cmm,rmeloca}@gmail.com

Abstract. This report shows the process of developing a program whose goal is to apply on the basis of images the concepts of artificial intelligence seen in the classroom along with the concepts of literature, therefore discriminate five classes referring to characters in the TV series Simpsons, these classes extract features and starting these characteristics apply sorting algorithms.

Resumo. Este relatório mostra o processo de desenvolvimento de um programa cujo objetivo é aplicar sobre uma base de imagens os conceitos de inteligência artificial vistos em sala de aula juntamente com os conceitos da literatura, para tanto discriminamos cinco classes referentes aos personagens do seriado de TV Simpsons, nestas classes extraímos características e a partir destas características aplicamos algoritmos de classificação.

1. Introdução

Muito se discute, sobre o tema “Inteligência artificial” e a utilização desta área para solucionar os inúmeros problemas existentes no mundo atual, embora a humanidade ainda engatinhe no desenvolvimento de soluções, alguns sistemas inteligentes já praticam o que chamamos de inteligência artificial, como é o caso de reconhecedores de padrões utilizando-se de imagens.

Podemos “ensinar” um computador através de três maneiras, à moda dos seres humanos. São elas: O aprendizado supervisionado, o aprendizado não-supervisionado e o aprendizado por reforço. A saber, o aprendizado não supervisionado é aquele em que o computador precisa extrair informações que julgue relevantes sem qualquer apoio ou regra informada; Obviamente este é o modelo de aprendizagem que mais se busca. O aprendizado por reforço também exige do computador extrair conhecimento de um amontoado de informações, encontrando conexões a respeito dos dados analisados, contando ainda com uma bonificação a cada resposta certa e uma reclamação a cada resposta incorreta; À moda dos que recebem gorjeta. Há ainda, o aprendizado não-supervisionado, que fornece ao computador um conjunto de treinamento para que este aprenda a classificar itens já sabendo da real resposta de cada um destes e só em seguida seja lançado ao mundo para prever itens não conhecidos.

Para formar um conjunto sobre o qual os classificadores poderão operar, deve-se criar uma matriz relacionando as instâncias com as características, ou seja, cada

instância (elemento que deseja-se classificar) possui um rol de características, sejam elas numéricas ou não-numéricas, que a define enquanto classe.

De longe, esta é, portanto, a principal parte do problema: Escolher bem as características para que elas definam bem a instância, de modo que um classificador possa encontrar semelhança esta e outra classe a fim de predizê-lo.

Para este trabalho, tratamos desta última categoria de classificadores. Neste, foi dado um problema que demandava escolher características de cada instância e classificá-las. A seção que segue, encarrega-se de elucidar o problema.

2. O Problema

Foi dado dois conjuntos, um para treino e outro para teste. Cada conjunto continha dezenas de imagens, cada uma com um (em alguns casos mais que um) personagem do desenho animado “Os Simpson”; São eles: Bart, Lisa, Homer, MARGE, e MAGGIE. Deste modo, cada imagem pertencente a um dos conjuntos configurou-se como uma instância de classificação.

Cabia ao escopo deste trabalho escolher características para serem extraídas destas instâncias, bem como escolher os classificadores de aprendizagem não-supervisionada para operarem sobre os conjuntos.

Como exaustivamente discutido, mais importa a boa escolha de características que o algoritmo de classificação empregado, pois uma vez que as características não delimitam bem as classes, não há comparação que permita distingui-las. Sob esse aspecto, consistia nisto a maior dificuldade do trabalho: extrair características das imagens dos personagens que fossem relevantes para distingui-los.

Aos olhos humanos distinguir os personagens da famílias “Simpsons” é muito fácil, todavia, obter características que os descrevam computacionalmente, uma vez que os personagens são muito semelhantes, em termos de cor, além da inconstância da localização dos pixels (isto é, localização dos personagens na imagem), encontrou-se grande dificuldade em fazê-lo.

Para extrair as características da imagem, utilizou-se: A predominância de cores, (adotando uma tolerância para mais ou para menos) orientando a busca pelas cores que caracterizam os personagens; O perímetro da borda; O Histograma horizontal normalizado de cada imagem em escala de cinza; E a Transformação Rápida de Fourier obtendo como entrada o histograma. Gerando conjuntos de características que a posteriori serão classificados utilizando classificadores do tipo K-NN (k-nearest neighbors) utilizando distância euclidiana, SVM (support vector machine) com kernel linear e sigmoidal e DT (decision tree) discreta binária.

3. Organização da Solução

Para implementação da solução do problema a priori observou-se os conjuntos de treino e teste do problema proposto afim de encontrar características que maior definem e distinguem uma imagem de outra. Tomando-se um problema onde o conjunto de imagens expressam cores que melhor definem os personagens, nós optamos por extratores de características que trabalham melhor com coloração de imagem, então dada a escolha dos extratores, definimos o primeiro o extrator

de cores predominantes, cada imagem é iterada sobre sua matriz de cores e dela quantificamos as cores que aparecem e são condizentes com as cores presentes na vestimenta de cada personagem, ao final da iteração temos um hashmap onde cada chave é a cor da roupa de um personagem e seu valor a quantidade de pixels presente na matriz, com este hashmap pegamos a cor predominante.

O segundo extrator de característica é um descritor de contorno (shape descriptor), com este método extraímos o contorno de uma imagem, obtemos uma matriz binária onde 1 define o contorno, para podermos utilizar este resultado no classificador extraímos o perímetro do contorno.

O terceiro extrator de característica também é classificado com descritor de cor, com histogramas de imagens temos um vetor de variação de tons de cores, para utilização do histograma convertemos a imagem para tons de cinza, em seguida é quantificado a variação de tons cinzas em toda a imagem, este vetor gerado é definido como um histograma, em alguns testes utilizamos o vetor normalizado para comparação de resultados.

No quarto extrator aplicou-se uma transformada de fourier, neste extrator o vetor de valores gerado no histograma é enviado para a classe FFT do java que transforma estes valores em números complexos, utilizamos este conjunto de números complexos para... XXXXXXXX.

Para classificação das características extraídas nós utilizamos 4 classificadores conhecidos na literatura, K-NN (k-nearest neighbors), SVM (support vector machine) e DT (decision tree). Utilizamos o classificador k-nn implementado para esta disciplina, para determinar a classe de um elemento que não está presente no conjunto de treinamento, o classificador procura K elementos do conjunto de treinamento que estejam mais próximos deste elemento desconhecido, ou seja que tenham a menor distância. O knn implementado utiliza a distância euclidiana para os cálculos da distância. O segundo classificador SVM é um classificador linear binário não probabilístico que dada uma entrada é verificado quais possíveis classes a entrada faz parte, o algoritmo de treinamento constrói um modelo que atribui novos exemplos a uma categoria ou outra, após esta etapa o SVM utiliza este modelo para prever os resultados com base no conjunto de teste.

O terceiro classificador decision tree toma como entrada uma situação descrita por um conjunto de atributos e retorna uma decisão, ou seja este valor retornado é o valor predito para o valor de entrada, na árvore de decisão os valores de entrada podem ser discretos ou contínuos. Para nossa implementação da solução foram utilizados os valores de entrada contínuos em uma árvore de decisão binária e não multiclasse. O funcionamento da decision tree é equiparado a uma árvore de busca onde em cada nó é tomada uma decisão até chegar ao nó folha. Na implementação da biblioteca utilizada continha apenas a árvore de decisão binária e entrada dos valores discretos.

Tabela 1. My caption

20	1	7	3	4
1	1	7	0	1
2	0	21	0	2
0	0	5	7	0
0	0	2	3	8

Tabela 2. My caption

K	Accuracy
1	57%
3	60%
5	54%
7	57%
21	45%

3.1. Diagramação

4. Implementação

5. Considerações Finais

Exemplo de referência CC01a [1].

<https://github.com/cjlin1/libsvm> <https://github.com/saebyn/java-decision-tree> <https://xanadu1010.wordpress.com/scripts-e-programas/transformada-rapida-de-fourier-com-java/>

Referências

- [1] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

KNN com $k = 3$, conjuntos normalizados e 4 características

4 características normalizadas, conjuntos normalizados

KNN com distância euclidiana e $k = 3$ e conjuntos normalizados por minimax

0.5789473684210527

0,6

Tabela 3. My caption

P. da Borda	29%
Cor Predominante	51%
Histograma	31%
T. de Fourier	30%
P. da Borda + Cor Predominante	60%
P. da Borda + Histograma	31%
P. da Borda + T. de Fourier	30%
Cor Predominante + Histograma	53%
Cor Predominante + T. de Fourier	52%
Histograma + T. de Fourier	50%
P. da Borda + Cor Predominante + Histograma	53%
P. da Borda + Cor Predominante + T. de Fourier	52%
P. da Borda + Histograma + T. de Fourier	30%
P. da Borda + Cor Predominante + T. de Fourier	52%
P. da Borda + Histograma + Cor Predominante + T. de Fourier	52%

Tabela 4. My caption

29	1	2	1	2
7	1	1	1	0
11	0	10	3	1
3	0	2	7	0
3	0	0	2	8

Tabela 5. My caption

30	1	2	0	2
7	1	1	1	0
14	0	9	1	1
3	0	2	7	0
3	0	0	0	10