Universidade Federal de Minas Gerais
Departamento de Ciência da Computação

DCC057 Mineração de Dados

Trabalho Prático 3 — Relatório de Carregamento dos Dados

Alexander Thomas Mol Holmquist

15 de março de 2021

1 Objetivo

Este relatório tem como objetivo delinear conclusões a respeito dos dados, obtidas durante o processo básico de carregamento.

2 Localização e Divisão dos Dados

O diretório principal para dados é ”data”. Os dados são compostos de 3297 imagens, e estão divididos fisicamente em dois diretórios, ”train”e ”test”. Esta divisão se refere a uma separação aleatória de 5 dobras realizada previamente sobre os dados. Assim, o conjunto de treinamento agrupa 2637 imagens, enquanto o conjunto de testes agrupa 660 imagens.

Um nível além, há mais uma divisão, desta vez entre ”benign”e ”malignant”. Como os nomes indicam, as imagens rotuladas como sendo de câncer benigno foram colocadas em ”benign”, e as rotuladas como maligno foram colocadas em ”malignant”.

3 Codificação dos Arquivos

Os arquivos das imagens estão todos no formato JPEG. Cada imagem possui dimensões 224 x 224 pixels, sendo que cada pixel é um valor RGB. O espaço total ocupado pelos arquivos é de aproximadamente 171.7MB.

Este formato pode ser carregado em python3 através da biblioteca skimage, disponível publicamente. Especificamente, a função skimage.io.imread lê o arquivo referido pelo caminho passado como argumento, e retorna um vetor Numpy de dimensões (224, 224, 3).

4 Manipulações

Um pouco de prática com os dados de treinamento leva a um entendimento de que são muito extensos para se utilizar em qualquer algoritmo, e a sua dimensionalidade é muito grande, quando transformados em uma matriz de dados (dimensões 2697, 150528 para dados de treino).

Os algoritmos padrão de redução de dimensionalidade: PCA e KernelPCA foram aplicados, mas não terminam. Ficou evidente, então, a necessidade de trabalhar com uma representação inferior.

Esta foi então gerada por um algoritmo rudimentar, que toma amostras de blocos de pixels de tamanho 4x4 da imagem, e cria um novo pixel cujos valores é a média dos valores dos píxels do bloco original. Assim, foi criada uma representação de dimensões (56, 56) para cada imagem, e a matriz bidimensional gerada ao aplanar as imagens ficou com 16 vezes menos atributos (9408) que a original.

Nesta manipulação, 98% da variância é preservada. Porém, não é garantido que aspectos que podem ser fundamentais para o aprendizado de máquina foram preservados. Mesmo assim, por conta da restrição de tempo, ferramentas e qualificação, este caminho foi escolhido.

5 Métodos

Como foi dito, a aplicação do algoritmo de redução de dimensionalidade PCA falhou, sem conseguir terminar a execução, mesmo na matriz final de dimensões (2697, 9408). Isto é um forte indicativo de que haverão sérias restrições sobre a viabilidade dos métodos a serem utilizados.

Todavia, explorações mais profundas não foram feitas, pois foge do escopo da fase um do projeto.

6 Código fonte produzido

Como resultado do trabalho desta fase, foram escritas uma série de funções em python3 para carregamento de dados. Todas elas estão no diretório ”pysrc/loading”. Notavelmente, a função **loadImgDataset** está encarregada de resumir todo o trabalho, produzindo uma matriz em que cada linha se refere a uma imagem já com a qualidade reduzida pelo método mencionado na Seção 4.