Inteligência Computacional - Denoising Dirty Documents

Bernardo D. C. G. de Amorim & Pedro V. Nacif

18 de Novembro de 2015

1 Introdução

OBS: As sub-seções do documento são preliminares e baseadas no modelo de relatório e devem mudar ao longo do projeto.

1.1 O Problema

O problema do denoising (remoção de ruído) de imagens consiste de desenvolver algoritmos que "limpem" imagens deterioradas por uma série de fatores, desde o processo de escaneamento até manchas de fluidos e dobras no papel.

Nossa tarefa é limpar imagens: remover as manchas; remover as dobras de papel; melhorar o contraste; e deixar somente o texto, o mais legível possível. Temos uma série de imagens de treinamento, constituidas de uma imagem "limpa" e uma onde um ruido foi adicionado artificialmente. No nosso caso, queremos usar aprendizado de máquina para gerar um algorítimo que consiga limpar essas imagens, treinando-o de alguma forma com as imagens dadas.

1.2 Bibliografia

Blog do Colin Priest[1], segundo colocado no contest do Kaggle[2]

1.3 Dados

O dataset[3] consiste de cerca de 200 imagens de treinamento sujas e a mesma quantidade de treinamento limpas entre 40 e 50 Mb, em formato png.

Um dos maiores desafios desse problema, assim como em grande parte dos problemas de aprendizado de máquina são:

- Definição do tipo de problema: Define qual vai ser a saída. Será um problema de regressão que dirá o valor para cada pixel da imagem de saída? Será um problema de classificação, que usará cores fixas para aumentar o contraste? Podemos tomar diversos caminhos; um dos mais simples é a regressão, onde a saída é o brilho de cada pixel.

- Extração de Características: A imagem só nos dá a informação base, isto é, o valor de brilho para cada pixel. Podemos usar isto diretamente como uma caraterística para treinar e utilizar nosso algorítimo, isto é: determinar o valor do brilho esperado a partir do brilho dado. Podemos pensar um pouco a mais, como tentar extrair informações dos arredores de cada pixel, como o brilho de cada pixel num raio qualquer, ou a variancia dos pixels num outro raio, ou qualquer outra informação que possa ser extraída e que tenha algum valor para "entender" a imagem.

Entretanto, deve-se tomar cuidado para não adicionar muitas features sem pensar sobre o que é realmente útil para o modelo funcionar, e por isso vamos ter que explorar cautelosamente como analisar os dados.

[1] http://colinpriest.com/
[2] https://www.kaggle.com/
[3] Obtido em: https://www.kaggle.com/c/denoising-dirty-documents

2 Tecnologia

A ferramente selecionada para realizar este projeto é a linguagem R[4], que é uma implementação da linguagem de programação de estatística S.

A decisão por esta ferramenta está baseada no fato de ser um projeto de código aberto e de software livre; por conter, já incluso na linguagem, diversas ferramentas estatísticas; e por ter um repositório de pacotes (CRAN[5]) com diversos pacotes de aprendizado de máquina e de visualização de dados.

2.1 Avaliação Preliminar dos Dados

Para começar com o modelo mais simples, vamos utilizar o modelo onde realizamos uma regressão tentando relacionar o brilho da imagem desejada com o brilho da imagem com ruído, ou seja, um problema de $y = f(x)$, onde x é o brilho da imagem com ruído e y é o brilho da imagem limpa. Ao olhar a relação entre todos os x e todos os y de todas as imagens, temos a seguinte relação:

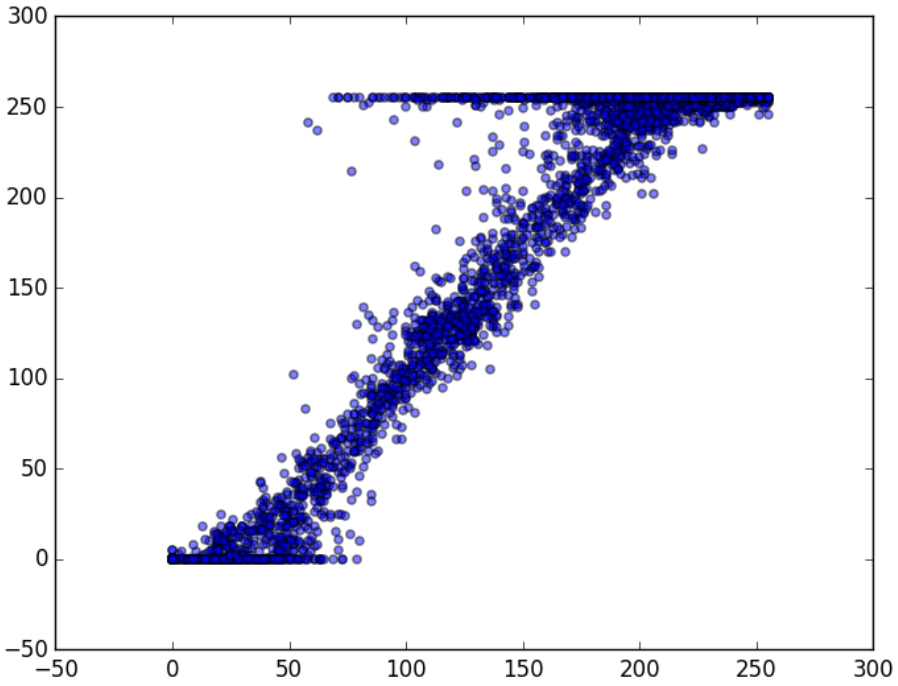


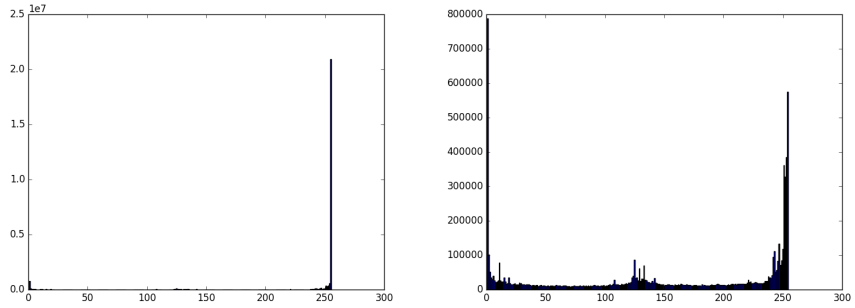
Figura 2.1: Relação amostral (10.000 amostras aleatorias) entre pixel com ruído(x) e pixel limpo (y)

Ou seja, para grande parte dos casos, podemos representar $y = \alpha x + \beta$, obtendo o α e o β a partir de um algorítimo de regressão linear. Entretanto, isto não é verdade para todas as imagens e todos os pixels, portanto devemos pensar em modelos diferentes, ou numa combinação linear de potencias de x,

[4] https://www.r-project.org/
[5] https://cran.r-project.org/

ou até mesmo adicionando outras características como variáveis a nossa função de regressão: $y = f(a, b, c, d, e, ...)$.

Além disso, podemos ver também que grande parte dos dados de saída são brancos (255), como na Figura 2.2. Isso indica que um problema de classificação pode ser útil também. Ou apenas uma regressão com um thresholding.



(a) Todos

(b) Não brancos

Figura 2.2: Distribuição de pixels limpos

3 Metodologia

3.1 Descrição da Metodologia

O objetivo é achar um modelo que consiga limpar a imagem, isto é, determinar o brilho de um pixel dado propriedades referentes a ele. O modelo consiste em entender **quais dados** serão usados e qual é a **função** e o **método de treinamento** que modela o resultado a partir dos dados utilizados.

Para obter os dados podemos pensar em diferentes formas de **extrair essas informações** da imagem. Na hora de escolher a **função** podemos testar várias, evoluindo em complexidade, partindo de um modelo mais simples (linear) até modelos não-lineares mais complexos (uma Rede Neural, por exemplo). Além disso devemos nos atentar ao **método de treinamento**, isto é, como vamos utilizar os dados para chegar nos parametros do modelo. Isto envolve selecionar as features mais relevantes, para agilizar o tempo de treinamento e envolve em testar diferentes algorítimos de treinamento, visando evitar mínimos locais na redução das medidas de erro.

3.2 Descrição da Solução do Problema Proposto

Como primeira solução, será utilizado uma relação linear entre um pixel "sujo" e o pixel "limpo" correspondente. Precisamos achar então uma função $f(x) = ax + b$ que modele este problema. Para isso vamos utilizar um regressor linear por **RMS**.

3.3 Resultados Preliminares

Comparação entre quantidade de pixels próximos utilizados

Percebemos que adicionar pixels ao redor melhora um pouco o resultado de um regressor linear simples:

Pixels	RMS
1x1	572.78 ± 101.96
3x3	542.44 ± 86.59
5x5	476.94 ± 70.04

Imagens filtradas

Em algumas das imagens conseguimos obter um bom resultado (onde o ruido era basicamente uma cor só), e em outras nem tanto.

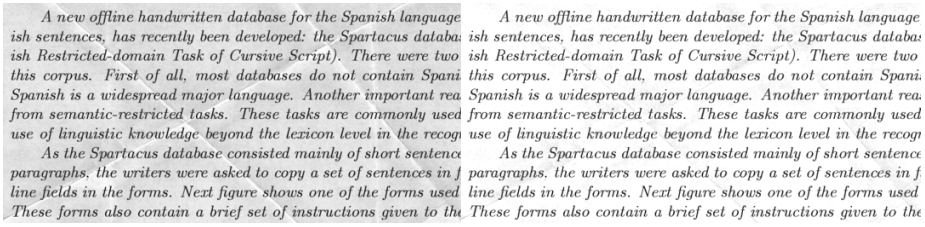


Figura 3.1: Bom resultado, utilizando apenas uma janela de 1x1

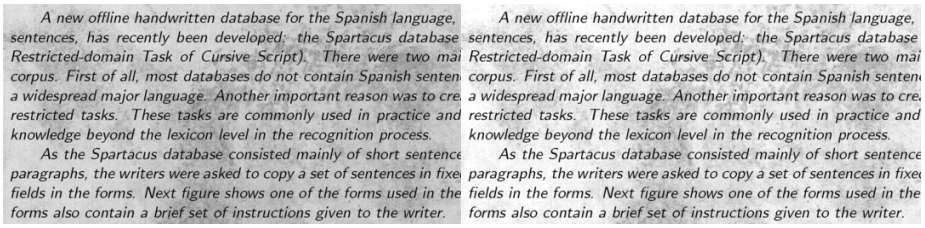


Figura 3.2: Resultado medíocre, com janela 1x1 também

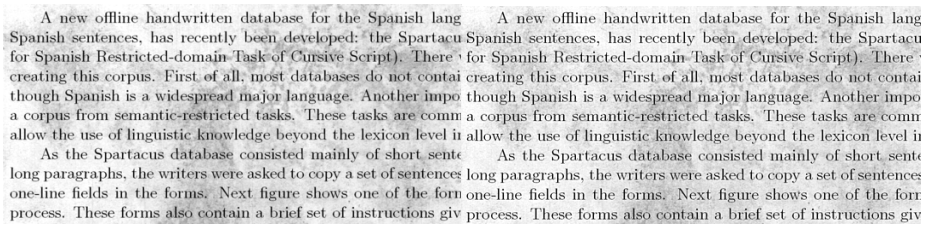


Figura 3.3: Pouca melhoria perceptiva entre janela 1x1 e 5x5

4 Resultados