

Aprendizagem de Máquina

Representação

Luiz Eduardo S. Oliveira

Universidade Federal do Paraná
Departamento de Informática



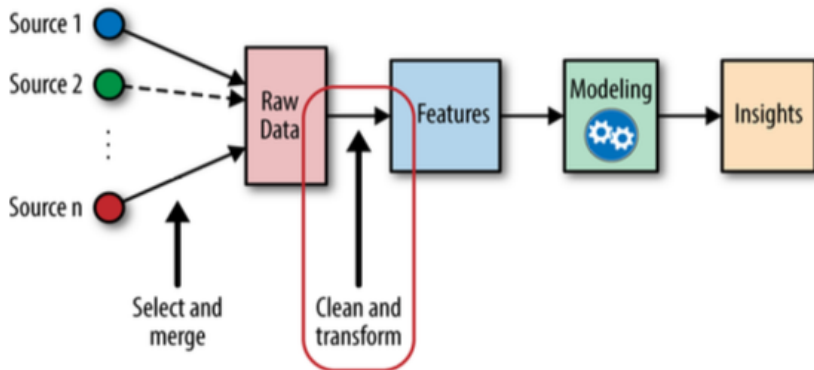
Conteúdo

- 1 Introdução
- 2 Extraíndo Representações de Imagens
- 3 Textura

Objetivos

- Apresentar diferentes formas de representação.
- Importância da extração de características para os algoritmos de aprendizagem de máquina.

ML Pipeline



Representação

Objetivo da Representação

Caracterizar um objeto através de medidas, as quais são bastante similares para objetos da mesma classe, e bastante diferentes para objetos de outras classes.

Importante: Devem ser **discriminantes** e **invariantes**

Representação

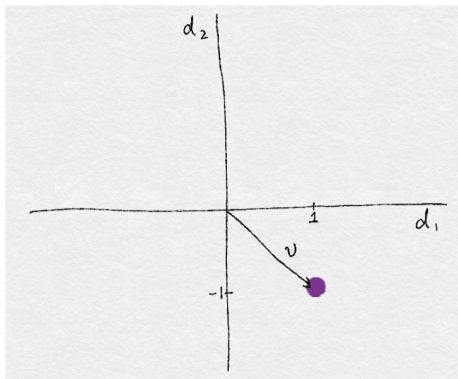
- Dados numéricos
 - ▶ Normalização
 - ▶ Pre-processamento
- Texto
- Imagens

Dados Numéricos

- Em algumas aplicações não precisamos converter uma imagem, texto, etc, em vetores numéricos
- Temos acesso direto ao vetor de características, por exemplo
 - ▶ Medidas de um sensor
 - ▶ Preço de compra
 - ▶ Etc..
- Mesmo assim, algumas transformações são necessárias para converter o dado bruto em características.
 - ▶ Escala das características (impacto no cálculo de distâncias)
 - ▶ Selecionar atributos relevantes (feature selection)

Escalares, Vetores e Espaços

- Um simples valor numérico é também conhecido como escalar.
- Uma lista ordenada de escalares é conhecido como vetor.
- Um vetor se encontra em um espaço vetorial.



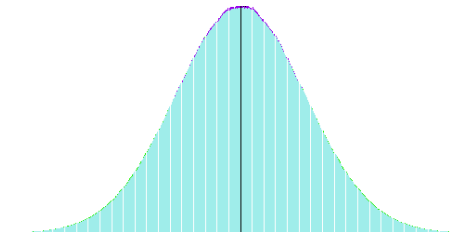
Algumas transformações básicas

Binarização

- Em alguns casos estamos interessados apenas se um evento aconteceu.
- A quantidade de vezes pode não importar.
- Nesses casos, um vetor binário no qual “1” indica se o evento aconteceu e “0”, caso contrario, pode representar o problema em questão.

Quantização

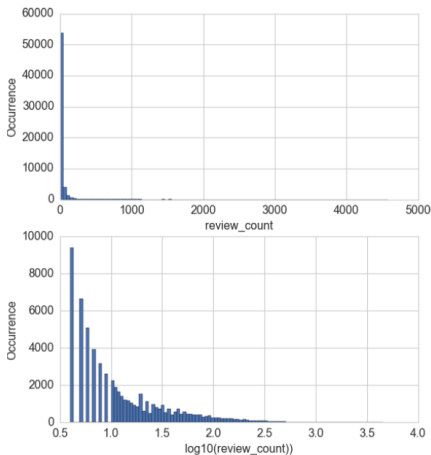
- Transformar valores contínuos em discretos (categorias).
- Cada bin representa um grau de intensidade específico.
- Reduz o tamanho do vetor de atributos.



Algumas transformações básicas

Log

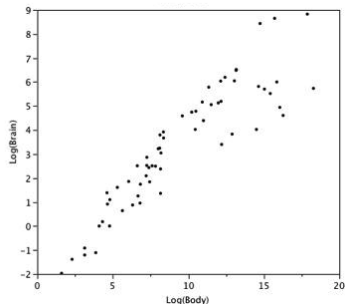
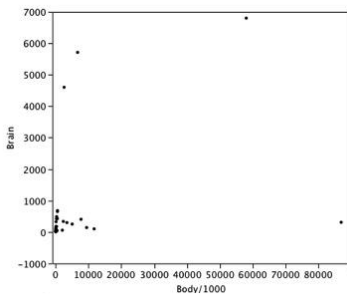
- O uso de funções log são ferramentas interessantes para lidar com distribuições fortemente desbalanceadas (heavy-tailed distribution).



Algumas transformações básicas

Log

- Realçar a relação entre os dados.



Normalização

Escala

- Alguns modelos de aprendizagem de máquina são sensíveis a escala dos dados.
- Nesses casos é importante normalizar os dados de forma que todas as características tenham a mesma escala
- Esse processo também é conhecido como “Feature Scaling”
- Uma função comumente utilizada para isso é a min-max

$$\hat{x} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Normalização

Variância (“Standartization”)

- Subtrai a média e divide pela variância
- Como resultado temos características com média 0 e variância 1

$$\hat{x} = \frac{x - \mu}{\sigma}$$

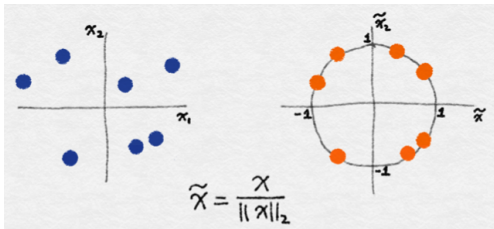
- Esse tipo de normalização é frequentemente utilizado em redes neurais. Melhora o tempo de convergência no treinamento.

Normalização

Normalização L2

- Divide o vetor pela sua norma, resultado em vetores com a mesma magnitude.

$$\hat{x} = \frac{x}{\|x\|_2}, \text{ em que } \|x\|_2 = \sqrt{\sum x_i^2}$$



Seleção de Características

- Eliminar características correlacionadas ou pouco discriminantes
- Reduz a complexidade do modelo, consequentemente o tempo de aprendizagem.
- Não é um processo trivial, pois para encontrar o melhor conjunto de características todas as combinações devem ser testadas
- Nesse caso, o total de combinações é 2^N , em que N é o número de características.
- Invariavelmente trabalhamos com mais de 100 atributos.
- Os métodos de selecção de características¹ podem ser divididos em
 - ▶ Filter
 - ▶ Wrapper

¹M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.* 1, 3 (1997) 131–156

- Diversas aplicações, como por exemplo, classificação de texto, recuperação de informação, agrupamento e organização de documentos, identificação de autoria, etc..
- Algoritmos de aprendizagem de máquina não trabalham com texto diretamente, ou seja, o texto deve ser convertido em números.
- Alguns métodos populares que convertem textos em números são:
 - ▶ Bag-of-Words
 - ▶ Word Embedding

Bag-of-Words - BoW

- Estratégia simples e flexível para representar textos.
- Envolve dois passos:
 - ▶ Definir um vocabulário conhecido.
 - ▶ Quantificar a presença de palavras conhecidas.
- O termo “bag” é usado por a ordem e estrutura de aparição das palavras não são levados em consideração.
- O modelo se preocupa somente se as palavras do vocabulário aparecem no texto.

Um exemplo

- Passo 1: Aquisição dos dados. Considere o texto abaixo escrito por Charles Dickens

“ *It was the best of times,
it was the worst of times,
it was the age of wisdom,
it was the age of foolishness,*

- Passo 2: Definir o vocabulário.
 - ▶ Considerar palavras únicas
 - ▶ Nesse caso, teríamos o seguinte vocabulário com 10 palavras: [it, was, the, best, of, times, worst, age, wisdom, foolishness]

Um exemplo

- Passo 3: Criar vetores
 - ▶ O vetor tem o tamanho do vocabulário.
 - ▶ Contar a frequência de aparição de cada palavra no texto.
 - ▶ Para o documento “it was the age of wisdom”, temos o seguinte vetor [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
- Novos documentos que contem palavras fora do vocabulário podem ser codificados. Nesse caso, as palavras desconhecidas são descartadas.

Gerenciando o Vocabulário

- Evitar vetores muito grandes e esparsos.
- Ignorar maiúsculas/minúsculas, pontuação, palavras muito frequentes (of, the, a, etc)
- Verbos no infinitivo (“stemming”)
- N-gram: Sequência de N palavras
 - ▶ Para k palavras temos k^2 2-grams (bigramas).
 - ▶ Nem todos fazem sentido.
- Eliminar palavras muito frequentes, pois elas discriminam pouco.
- Cuidado com palavras muito raras, pois elas podem representar ruído e não informação discriminante.

- Um problema com a simples frequência de palavras é que algumas delas podem dominar o documento, mas não necessariamente conter informação discriminante.
- Uma solução é normalizar a frequência de palavras pela quantidade que elas aparecem no documento.
- Essa abordagem é denominada TF-IDF
 - ▶ Term Frequency: quantifica a frequência da palavra no documento
 - ▶ Inverse Document Frequency: Quantifica o quão rara é a palavra considerando todos os documentos.

TF-IDF

Exemplo

The course is about building 'web-intelligence' applications exploiting big data sources arising social media, mobile devices and sensors, using new big-data platforms based on the 'map-reduce' parallel programming paradigm. The course is being offered ..

- Considere uma coleção de 50000 documentos
- A tabela abaixo mostra o cálculo TF-IDF para algumas palavras. Note que aquelas mais relevantes possuem um score maior

A	B	C	D	E
		IDF (50k/B)	TF	TF-IDF $\log_2(\text{IDF}/\text{TF})$
the	25000	2	2	0.00
course	2000	25	2	3.64
media	7000	7	1	2.84
map-reduce	200	250	1	7.97
web-intelligence	300	167	1	7.38

Limitações BoW

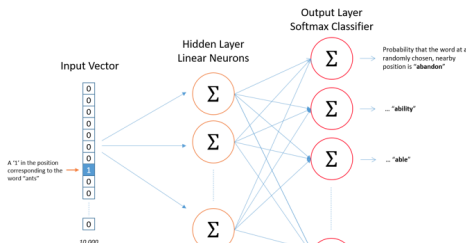
Vocabulário

- Definição cuidadosa do vocabulário (tamanho)
- Falta de contexto (old bike vs used bike)

Word Embedding

Word2Vec

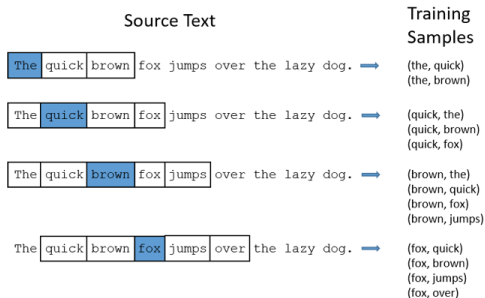
- Modelo baseado em uma rede neural com uma camada escondida em que,
 - ▶ A entrada é um “hot-vector” do tamanho do nosso vocabulário (transformar o vocabulário numa entrada numérica).
 - ▶ A saída tem o mesmo tamanho da entrada.
 - ▶ A camada escondida é um parâmetro (tamanho do vetor pelo qual cada palavra será representada)
- A camada escondida não tem função de ativação
- A camada de saída utiliza uma soft-max (garante que a soma de todas as saídas seja = 1)



Word2Vec

Treinando a rede

- O treinamento é baseado em pares de palavra extraídos do texto
- A quantidade de pares é definido pelo tamanho da vizinhança (W - window size).
- Na figura abaixo, $W = 2$
- A rede aprenderá as estatísticas a partir do número de vezes que cada par aparece na base de treinamento.



Word2Vec

Usando a rede

- Como palavras similares geralmente aparecem no mesmo contexto, a rede tende a aprender a mesma representação para elas.
- Por exemplo, “inteligente” e “esperto” geralmente aparecem em um contexto diferente de “pneu” e “motor”.
- Desta forma a rede deve codificar pesos parecidos na sua camada intermediária para palavras que aparecem no mesmo contexto.
- Ou seja, a camada de saída é desprezada e uma dada palavra é representada pela multiplicação do vetor de entrada pelos pesos entre a camada escondida e a camada de saída.

$$[0 \ 0 \ 0 \ 1 \ 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \ 12 \ 19]$$

```
In [14]: w1 = "dirty"
print model.wv.most_similar (positive=w1)

[(u'filthy', 0.8605918288230896), (u'unclean', 0.7872298955917358), (u'stained', 0.7809049487113953), (u'dusty', 0.7737701535224915), (u'grubby', 0.7640288472175598), (u'smelly', 0.7638705968856812), (u'dingy', 0.7330941557884216), (u'gross', 0.7314556837081909), (u'disgusting', 0.711143970489502), (u'grimy', 0.71081527277107239)]
```

```
In [22]: print model['dirty']

[ -2.19303086e-01 -2.88474411e-02 -8.11420250e+00  2.83480191e+00
 -1.35823917e+00  1.56496143e+00 -2.11412811e+00 -1.62608135e+00
 1.57304680e+00  1.30592442e+00 -3.51201391e+00  4.09215689e-01
 -2.68659759e+00  1.29351676e+00 -1.57445872e+00  5.18705882e-02
 1.78067422e+00 -6.56598377e+00 -9.09853399e-01 -1.96509421e+00
 2.41910005e+00  9.83066320e-01  7.38306880e-01 -3.03441143e+00
 -3.46890062e-01  2.12788367e+00  3.04096246e+00  6.02690220e-01
 4.34922743e+00  2.05738497e+00  6.58910370e+00  1.44861686e+00
 2.99755168e+00 -1.20672274e+00 -1.70791793e+00 -5.29811907e+00
 -3.61028242e+00  2.15190387e+00 -8.76770794e-01  2.16893077e-01
 -1.92470384e+00 -1.39617062e+00  1.45718467e+00 -3.35419118e-01
 -1.89717567e+00  1.49699616e+00 -1.50018942e+00 -5.36821038e-02
 -4.08865422e-01  2.79571116e-01  3.93044472e+00  2.92954922e-01
 -1.75992060e+00  4.83922958e+00  7.18770549e-03 -2.03018188e+00
 3.15000486e+00  2.04503608e+00  3.13388586e+00  3.07883203e-01
 -2.27254295e+00 -7.57412195e-01 -2.04093289e+00 -3.46738839e+00
 2.60640979e+00  6.26769483e-01 -1.53049171e+00 -3.69118118e+00
 -4.08453703e-01 -3.10141873e+00 -4.34588134e-01 -4.76577431e-01
 -1.75007510e+00 -2.43567204e+00 -1.76848733e+00  4.33461070e-01
 2.10797501e+00 -1.50509930e+00 -9.59908366e-01  4.31083679e+00
 7.25064468e+00 -1.97127008e+00 -1.16364408e+00 -3.66479850e+00
 6.77183032e-01 -5.17210531e+00 -4.88205814e+00  4.87751454e-01
 -3.10203743e+00 -4.99226183e-01 -2.41884559e-01  1.21832609e+00
 -7.90648580e-01  4.80678082e+00 -1.22414458e+00 -5.98880386e+00
 3.12212016e-02  2.12671232e+00 -3.14978933e+00 -1.26722567e-02]
```

```
In [23]: w1 = "france"
print model.wv.most_similar (positive=w1,topn=6)

[(u'germany', 0.707061119270325), (u'canada', 0.6965396404266357), (u'spain', 0.6817945241928101), (u'england', 0.6776233315467834), (u'mexico', 0.6762300133705139), (u'hawaii', 0.6679205298423767)]
```

Imagens



Imagens



Imagens



Características Estatísticas

Momentos de Hu

- Característica Global e Invariante
- Medidas puramente estatísticas da distribuição dos pontos.
- Considere a imagem binária de um objeto $M \times N$ onde $I(x, y)$ representa o estado do pixel (x, y) - preto ou branco

Um momento regular de ordem $(p+q)$ é definido por:

$$m_{pq} = \sum_{x=1}^N \sum_{y=1}^M x^p y^q I(x, y), p, q = 0, 1, 2.$$

Momentos de Hu

- O momento de ordem 0 (m_{00}) representa a superfície, enquanto os momentos de ordem 1 (m_{01} e m_{10}) definem o centro de gravidade (x_g e y_g) da imagem

$$x_g = \frac{m_{10}}{m_{00}}, y_g = \frac{m_{01}}{m_{00}}$$

- Com o intuito de serem invariantes a rotação e translação, Hu definiu os momentos centrais n_{pq}

$$n_{pq} = \sum_{x=1}^N \sum_{y=1}^M (x - x_g)^p (y - y_g)^q I(x, y), p, q = 0, 1, 2, \dots$$

Momentos de Hu

- Os momentos centrais de ordem 2 permitem achar os eixos principais de inércia, os prolongamentos e as orientações da forma.
- Para que os momentos sejam invariantes a escala, os mesmos devem ser normalizados pelo tamanho da imagem.

$$\mu_{pq} = \frac{n_{pq}}{m_{00}^{\gamma}} \text{ com } \gamma = \frac{p+q}{2} + 1$$

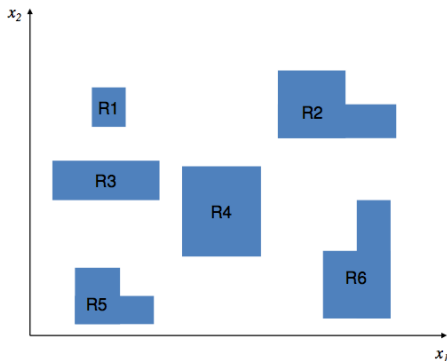
Momentos de Hu

Finalmente, os momentos mais utilizados são os 7 momentos invariantes de HU, (de ordem 2 e 3):

- $\Phi(1) = (\mu_{20} + \mu_{02})$
- $\Phi(2) = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$
- $\Phi(3) = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$
- $\Phi(4) = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$
- $\Phi(5) = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]$
- $\Phi(6) = (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$
- $\Phi(7) = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + 3\mu_{12})^2 - (\mu_{21} + \mu_{03})^2]$

Monentos de Hu

- Um exemplo



Monentos de Hu

Momento	R1	R2	R3	R4	R5	R6
1	1.67E-01	1.94E-01	2.08E-01	1.67E-01	1.94E-01	1.94E-01
2	0.00E+00	6.53E-03	1.56E-02	0.00E+00	6.53E-03	6.53E-03
3	0.00E+00	1.02E-03	0.00E+00	0.00E+00	1.02E-03	1.02E-03
4	0.00E+00	4.56E+05	0.00E+00	0.00E+00	4.56E+05	4.56E+05
5	0.00E+00	4.25E-09	0.00E+00	0.00E+00	4.25E-09	4.25E-09
6	0.00E+00	1.70E+06	0.00E+00	0.00E+00	1.70E+06	1.70E+06
7	0.00E+00	-8.85E+09	0.00E+00	0.00E+00	-8.85E+09	-8.85E+09

R1 e R4, R2 e R5 são diferentes escalas do mesmo objeto R6 é a versão rotacionada de R2 e R5

Características Estruturais

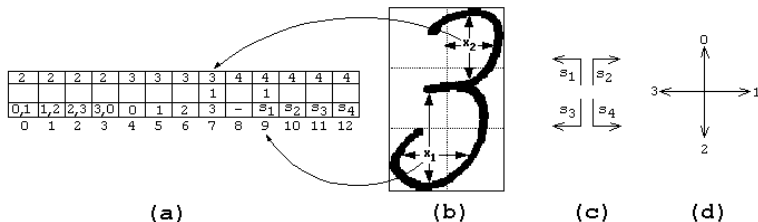
Extraem informações da estrutura do padrão.

- Contornos
- Concavidades
- Esqueleto
- Perfil
- Área, Distribuição

Muitas vezes informações estatísticas são computadas a partir das informações estruturais.

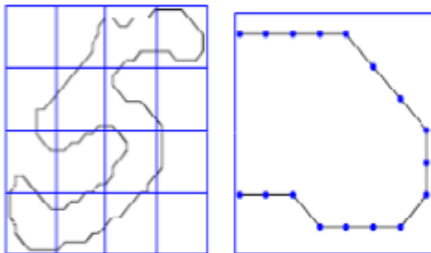
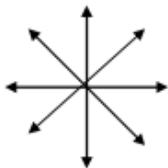
Concavidades

Obter informações estruturais com base nas concavidades do objeto.
Baseia-se na quantidade de vizinhos pretos em n-direções



Contorno

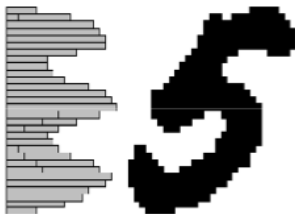
Para cada pixel do contorno, contabiliza-se a direção do próximo pixel.



Vetor de características teria 8 posições onde cada posição teria a soma das direções.

Histogramas de Projeção

Nesse caso podemos usar um histograma para representar a distribuição dos pixels da imagem.



Informações do histograma podem ser extraídas em diferentes níveis (quantização).

Vetores de diferentes tamanhos

Zoneamento

- Zoneamento é uma estratégia bastante usada para enfatizar determinadas regiões de um padrão.
- Características locais
- Zonas simétricas e assimétricas
 - ▶ Depende do problema que está sendo abordado
- Normalização implícita

Zoneamento



4 zonas simétricas

Com base na informação das duas zonas inferiores somente, temos informações similares ao dígito 3



Qual seria a melhor estratégia de zoneamento?

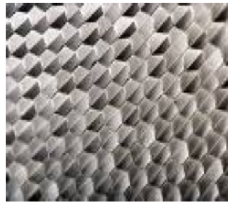
Padrão visual que possui algumas propriedades de homogeneidade que não resultam simplesmente de uma cor ou intensidade.

- Constituída de elementos mutuamente relacionados: a primitiva de textura dependente de escala
- Composta de um grande número de elementos similares mais ou menos ordenados
- Relacionada com coeficientes de uniformidade, densidade, aspereza, regularidade, intensidade, dentre outros, oriundos da probabilidade de ocorrência de variações tonais.
- Descritas por medidas que quantificam suas propriedades de suavidade, rugosidade e regularidade.

Descritores de Textura

Encontrar padrões de homogeneidade que não estão presentes em uma simples cor ou intensidade.

- Bastante utilizadas em problemas de classificação e também na recuperação de imagens
 - ▶ Imagens médicas
 - ▶ Imagens de satélite



Principais Descritores

- Momentos do Histograma
- Matriz de co-ocorrência
- Fractais
- Filtros de Gabor
- Local Binary Patterns (LBP)

Matriz de Co-ocorrência

- Medidas de textura calculadas a partir do histograma sofrem a limitação de não carregarem informações sobre a posição relativa dos pixels em relação uns aos outros.
- Uma maneira de trazer essa informação ao processo de análise de texturas é considerar não apenas a distribuição de intensidades, mas também as posições dos pixels com valores de intensidade iguais ou similares.
- Medidas de segunda ordem.

Matriz de Co-ocorrência

- Seja P um operador de posição e A uma matriz $k \times k$, cujo elemento a_{ij} seja o número de vezes que os pontos com o nível de cinza z_i ocorrem (na posição especificada por P), relativamente a pontos com o nível de cinza z_j , com $i \leq k, j \leq k$.

Matriz de Co-ocorrência

- Considere por exemplo uma imagem com 3 níveis de cinza, $z_0 = 0$, $z_1 = 1$ e $z_2 = 2$.

0	0	0	1	2
1	1	0	1	1
2	2	1	0	0
1	1	0	2	0
0	0	1	0	1

P = um píxel a direita e pixel abaixo

- Sendo assim, A seria uma matriz 3×3 .
- a_{00} é o número de vezes que um ponto com nível de cinza 0 aparece **abaixo e a direita** de outro píxel com o nível 0.
- Nesse caso a_{00} teria o valor 4.

Matriz de Co-ocorrência

- O valor de a_{02} é o número de vezes que um ponto com nível de cinza 0 aparece abaixo e a direita de um pixel 2.

0	0	0	1	2
1	1	0	1	1
2	2	1	0	0
1	1	0	2	0
0	0	1	0	1

Desta maneira, a matriz de co-ocorrência será

$$A = \begin{vmatrix} 4 & 2 & 1 \\ 2 & 3 & 2 \\ 0 & 2 & 0 \end{vmatrix}$$

A qual deve ser normalizada.

Matriz de Co-ocorrência - Descritores de Haralick

Variância (contraste da imagem)

$$\sum_i \sum_j (i - j)^2 P(i, j)$$

Entropia (suavidade)

$$- \sum_i \sum_j P(i, j) \log P(i, j)$$

Energia (uniformidade)

$$\sum_i \sum_j P^2(i, j)$$

Homogeneidade (distribuição)

$$\sum_i \sum_j P(i, j) / (1 + |i - j|)$$

Momento de 3a. ordem (distorção)

$$\sum_i \sum_j (i - j)^3 P(i, j)$$

Variância Inversa (inverso do contraste)

$$\sum_i \sum_j P(i, j) / (i - j)^2$$

Matriz de Co-ocorrência

Ex: Calcule a Entropia para a seguinte matriz de co-ocorrência

$$\begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 4 & 3 & 0 \\ 0 & 3 & 6 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}$$

i	j	$p(i, j, 1, 0^\circ)$	$\log p(i, j, 1, 0^\circ)$	$p(i, j, 1, 0^\circ) \log p(i, j, 1, 0^\circ)$
0	0	2	0,301	0,602
0	1	1	0	0
0	2	0	-	-
0	3	1	0	0
1	0	1	0	0
1	1	4	0,602	2,408
1	2	3	0,477	1,431
1	3	0	-	0
2	0	0	-	0
2	1	3	0,477	1,431
2	2	6	0,778	4,668
2	3	0	-	0
3	0	1	0	0
3	1	0	-	0
3	2	0	-	0
3	3	2	0,301	0,602

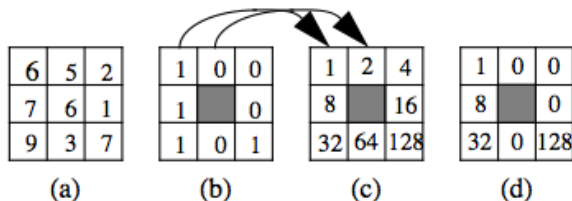
Entropia = -11.142

Local Binary Patterns

- Baseia-se no fato de que certos padrões binários locais à região de vizinhança de um pixel são propriedades fundamentais da textura de uma imagem.
- Deste modo, o histograma de ocorrência destas características é um bom descritor de textura.
- O operador $LBP_{P,R}$ produz 2^P padrões binários diferentes que podem ser formados por P pixels na sua vizinhança.
- (P, R) significa uma vizinhança de P pontos uniformemente distribuídos com um raio R

Local Binary Patterns

Calculo do LBP

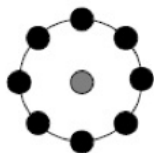


$$\mathbf{LBP} = 1+8+32+128=\mathbf{169}$$

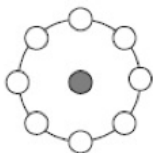
- (a) O valor do pixel central é escolhido como limiar
- (b) Imagem é binarizada com base nesse limiar
- Com base na mascara (c), o valor do LBP é calculado.
- O LBP calculado é o valor que deve ser atualizado no histograma

Local Binary Patterns

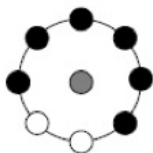
- Note que para uma vizinhança de 8 pixels (2^8) temos um histograma de 256 posições.
- Na prática, porem, 59 configurações são utilizadas.
- Aquelas que mostram padrões uniformes, como por exemplo:
00000000, 00111000, 11100001



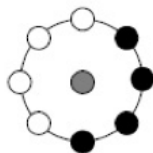
Pontos



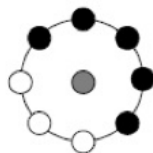
Áreas Planas



Fim de Linha

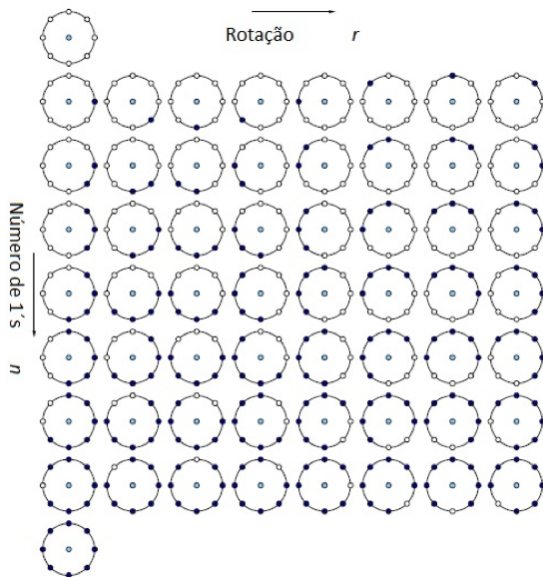


Bordas



Cantos

Local Binary Patterns



Gradientes

- Experimento conduzido por Hubel e Wiesel em 1962 mostrou “por acidente” que certos neurônios do cérebro respondiam somente na presença de bordas com certas orientações (([▶ Video Youtube](#)))



- Diversos descritores foram propostos para quantificar bordas e orientações, também conhecidos como gradientes.

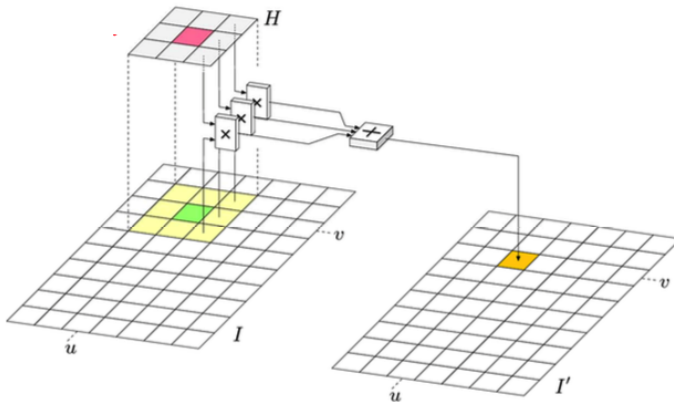
Gradientes

- Imagens de gradientes geralmente são produzidas através da convolução de filtros pré-determinados em imagens.
- Por exemplo, Sobel, Canny, Prewitt, etc

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Gradientes

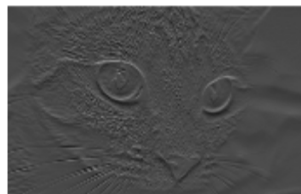
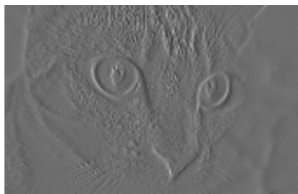
Convolução



$$I'(u, v) \leftarrow \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} I(u+i, v+j) \cdot H(i, j)$$

Gradientes

Exemplos de gradientes horizontais e verticais

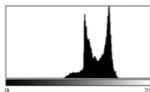


Características baseadas em Gradientes

- HoG (Histogram of Oriented Gradients)
- SIFT (Scale Invariant Feature Transform)
 - ▶ Características bastante utilizadas em problemas de visão computacional (rastreamento de objetos, etc.)

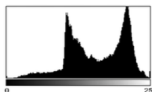
- Histogramas de Cor

- ▶ RGB (Red,Green,Blue)
- ▶ HSI (Hue, Saturation, Intensity)
- ▶ CMYK (Cyan, Magenta, Yellow, Preto)



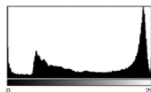
(a)

Baixo Contraste



(b)

Contraste Normal



(c)

Alto Contraste

Imagem

Histograma

CNN - Convolutional Neural Nets

Extratores de características baseados em redes pré-treinadas

