

Trabalho de Formatura Supervisionado

Aplicação de análise morfológica para segmentação de páginas em imagens de documentos

Ricardo de Cillo

Supervisora: Nina S. T. Hirata

Departamento de Ciência da Computação
Instituto de Matemática e Estatística, IME-USP

Resumo: Neste texto apresentamos nosso estudo sobre a aplicação de operadores morfológicos à segmentação de páginas de documentos, etapa importante na análise de documentos que busca extrair informações sobre a sua estrutura: regiões com títulos, legendas, figuras e blocos de texto. A qualidade da solução obtida será medida e comparada, segundo os mesmo critérios aplicados à resultados considerados estado da arte por pesquisadores da área.

São Paulo, 3 de dezembro de 2012

1 Introdução

Processamento e análise de documentos é uma importante subárea da área de reconhecimento de padrões cujo principal objetivo é a interpretação de um documento, ou seja, o entendimento da sua estrutura bem como o reconhecimento de cada um dos componentes estruturais.

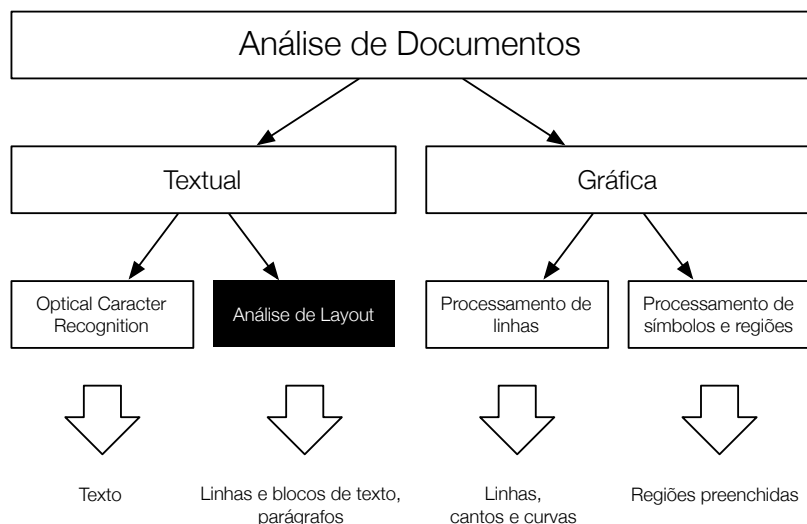


Figura 1: Contextualização do tema do trabalho entre as áreas da análise de documentos. Adaptado de [1].

Segmentação de página refere-se à tarefa de separar e rotular os diferentes componentes que fazem parte da estrutura das páginas de um documento, tais como: blocos de texto, gráficos, figuras, títulos, legendas, separadores, tabelas, fórmulas matemáticas e regiões com ruído.

Em geral, a segmentação de página é um dos primeiros passos no processo de entendimento de um documento. Uma vez identificados os blocos estruturais, processamentos específicos para cada tipo de bloco podem ser aplicados. Por exemplo, no caso de blocos de textos é conveniente fazer o reconhecimento de texto para que o mesmo possa ser armazenado em formato texto (e não imagem). Por outro lado, no caso de imagens, pode ser interessante armazená-las em alta resolução para manter a qualidade. Documentos digitalizados podem ser processados eficientemente em processos que envolvem armazenamento, edição, transmissão, ou busca, por exemplo.

Devido à grande quantidade de documentos, é interessante que o seu processamento seja realizado de forma automatizada ou pelo menos semi-automatizada. Para tal, diversas soluções computacionais vêm sendo propostas para o problema ao longo dos anos desde o surgimento desse campo de pesquisa. Automatizar esta tarefa reduz custos, aumenta a velocidade e capacidade de processamento de documentos além de possivelmente reduzir a taxa de erro humano na classificação de uma região.

Neste trabalho exploraremos a aplicabilidade de operadores morfológicos automaticamente gerados ao problema de segmentação de páginas.

completar a descrição da estrutura do texto

Este texto está organizado da seguinte forma. Na seção 2, apresentamos as definições e conceitos básicos que serão importantes para a leitura deste texto.

2 Fundamentos

2.1 Imagens digitais

Uma imagem digital monocromática pode ser definida como uma função $f : \mathbb{E} \subset \mathbb{Z}^2 \rightarrow \mathbb{K} = \{0, 1, \dots, k-1\}$, na qual k representa o número de tons de cinza. Tipicamente adota-se $k = 256$, ou seja, 8-bits de cor. Quando $k = 1$ as imagens são denominadas **binárias**; quando $k > 1$ as imagens são denominadas **tons de cinza**. Na prática, o domínio \mathbb{E} é um retângulo finito de dimensões $m \times n$ (uma matriz de m linhas e n colunas).

Uma imagem RGB (colorida) é uma função $f : \mathbb{E} \rightarrow \mathbb{K}^3$, onde cada componente \mathbb{K} representa a intensidade das cores vermelho, verde e azul, respectivamente.

2.2 Operadores de imagens

Um operador de imagens é uma função que mapeia imagens em imagens. Denotando $\mathbb{E} = \mathbb{Z}^2$, $K = \{0, 1, \dots, k-1\}$ e todas as imagens definidas em \mathbb{E} por $K^{\mathbb{E}}$, podemos representar um operador de imagens como $\Psi : K^{\mathbb{E}} \rightarrow K^{\mathbb{E}}$.

2.2.1 Binarização de imagens

A classe de operadores estudada restringe-se ao domínio das imagens binárias. Porém as imagens obtidas através de digitalização usualmente são coloridas (RGB de 24-bits). O processo de binarização é realizado por um operador que mapeia uma imagem colorida ou monocromática em uma imagem binária.

Neste trabalho, primeiramente transformamos as imagens coloridas para níveis de cinza e posteriormente aplicamos a binarização.

Existem muitos algoritmos que realizam esta tarefa. Uma revisão extensa dos mais conhecidos métodos de binarização é apresentada em [2]. Todos eles se aplicam a imagens em níveis de cinza, portanto inicialmente transformaremos a imagem colorida f em níveis de cinza g :

$$f(x) \in \mathbb{K}^3 \rightsquigarrow g(x) \in \mathbb{K} \rightsquigarrow b(x) \in \{0, 1\} \quad (1)$$

2.3 Classificação de objetos

Na área de reconhecimento de padrões e aprendizado computacional estudam-se métodos e técnicas para classificação de dados em geral. Os dados (padrões) a serem classificados correspondem, em geral, à representação digital de algum objeto concreto ou abstrato. O objetivo da classificação é atribuir um rótulo de classe a cada padrão observado.

Dependendo do problema, os rótulos de classe podem ser conhecidos ou não. Por exemplo, se desejamos fazer o reconhecimento de caracteres, os padrões são a imagem dos caracteres e os rótulos de classe são as identificações dos possíveis caracteres. Por outro lado, em problemas como na classificação de perfil de consumidores, pode não haver um conjunto de perfis pré-estabelecidos e o objetivo seria então identificar a possível existência de perfis. O primeiro é conhecido como problema de classificação supervisionada e o segundo como classificação não-supervisionada.

No caso da classificação supervisionada, supõe-se que os padrões são elementos de um espaço X e que o conjunto de rótulo de classe é dado por $Y = \{y_1, y_2, \dots, y_c\}$. Assim, um classificador pode ser expresso por uma função $f : X \rightarrow Y$.

Frequentemente X é um subespaço de \mathbb{R}^d . Assim, um padrão é representado por uma d -upla $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$.

Tabela 1: Exemplo de segmentação de imagem: separando frente e fundo.



2.4 Segmentação de imagens

A segmentação de imagens é um processamento comum a praticamente todos os processos que envolvem análise de imagens. Segmentar uma imagem corresponde a particionar o seu domínio, de forma que cada região resultante corresponda (do ponto de vista semântico) a uma componente de interesse na análise em questão. Este problema pode ser modelado como uma classificação de objetos, onde o conjunto de pixels de uma imagem são os objetos em X e as componentes em Y são regiões de interesse, como ilustrado na tabela 1.

2.5 Classificação dos componentes

O problema de segmentação de imagens de documentos pode ser modelado como um problema de classificação de objetos, onde cada pixel da imagem é rotulado através de uma função classificadora

$$\Psi: \mathbb{E} \rightarrow \mathbb{Y} \quad (2)$$

sendo \mathbb{Y} um conjunto composto pelas regiões de interesse:

- blocos de texto: região com parágrafos
- gráficos
- figuras
- títulos
- legendas
- separadores
- tabelas
- fórmulas matemáticas
- regiões com ruído

2.6 Avaliação da segmentação

Dizer aqui como se avalia uma segmentação. Poderia ser a nível de pixel, ou a nível de regiões ou componentes. Pixel é fácil, mas tem os potenciais problemas. No caso de região é preciso definir como comparar regiões — caso dos polígonos isotéticos, etc etc

Os dois principais métodos utilizados para avaliar a qualidade das soluções para este problema são a comparação da classificação dos pixels individualmente ou a classificação de regiões. Optamos pela comparação entre pixels pois nosso método produz como resultado a classificação de pixels e não a delimitação de regiões. Seria necessário agrupar os pixels em regiões para que pudessemos realizar outros tipo de avaliação, o que foge ao escopo deste trabalho.

Uma vantagem na comparação entre pixels e não regiões é o fato de que evitamos restrições arbitrárias na forma das regiões.

3 Aprendizado de operadores morfológicos

Construir operadores morfológicos que resolvam problemas complexos como o de segmentar uma página de documento, pode ser uma tarefa que demande muito tempo, experiência e conhecimento específico do assunto. Como estas imagens possuem características distintas dependendo da publicação, é possível que apenas um operador não consiga ser aplicado a todas as imagens. Ou seja, construir operadores com facilidade é um fator sensível para a escolha desta abordagem.

Nesta seção apresentamos um método para projetar operadores morfológicos de forma automática utilizando técnicas de aprendizado computacional. Primeiramente definiremos o conceito de operadores morfológicos.

3.1 Operadores morfológicos binários

Informalmente, operadores morfológicos são transformações entre imagens binárias que podem ser descritas através de propriedades geométricas e topológicas. Um exemplo amplamente conhecido é o da dilatação, cuja aplicação está exemplificada na figura [ref](#).

[figura com dilatação e erosão](#)

Formalmente define-se um operador morfológico binário como uma transformação localmente definida e invariante por translação entre reticulados completos.

Um dos principais resultados da morfologia matemática revela que qualquer operador binário pode ser decomposto em uma combinação de operadores mais simples, dilatação (3) e erosão (4), chamados de elementares.

$$\delta_B(X) = \{x \in E : B_x \cap X \neq \emptyset\} \quad (3)$$

$$\varepsilon_B(X) = \{x \in E : B_x \subseteq X\} \quad (4)$$

O conjunto B é denominado elemento estruturante.

3.2 Treinamento

A fim de projetar operadores morfológicos complexos de forma automática, utilizaremos o método proposto em [ref](#). Nele, um conjunto de pares de imagens de treinamento é transformado em um operador. Estes pares são compostos da imagem **original**, denotada por X e sua **derivada** Y , que exemplifica a transformação desejada. ([figura ref](#)).

O processo consiste de 3 etapas (diagrama [ref](#)): coleta, decisão e minimização. Ao fim, teremos uma função booleana que aproxima é uma aproximação do operador desejado. A seguir detalhamos cada uma das etapas.

3.2.1 Coleta

Seja $C(X_W) = \{X \cap W + z : z \in E\}$ o conjunto de todas as configurações observadas ao se transladar a janela W sobre a imagem X . Seja Y_z o valor 0 ou 1 encontrado na imagem Y na posição z . Construímos uma tabela com três colunas: $C(X_W)$, frequência observada $Y_z = 0$ e frequência $Y_z = 1$. (tabela exemplo [ref](#))

3.2.2 Decisão

As configurações observadas na etapa anterior possuirão valores relacionados em Y iguais a 0, 1 ou ambos. No caso de tanto 1 como 0 terem sido observados, escolheremos como valor para a função final a com maior número de ocorrências. No caso de alguma configuração não ter sido observada ou de o número de ocorrências empatar, escolheremos o valor que simplifica a próxima etapa: minimização.

3.2.3 Minimização

[Descrever algoritmo ISI](#)

4 Metodologia

O método empregado para resolver o problema é baseado em operadores morfológicos automaticamente gerados. Os dados de entrada são um conjunto de imagens de treinamento e um XML com a descrição do tipo e forma das regiões. A seguir realizamos a seguinte sequência de etapas:

1. Aquisição das imagens de treinamento
2. Binarização
3. Geração dos pares de imagens de treinamento
4. Construção do operador segmentador específico para o tipo de região especificada
5. Aplicação do operador em um conjunto de imagens

Cada um dos passos é detalhado nas seções seguintes.

4.1 Aquisição das imagens de treinamento

As imagens utilizadas nos experimentos foram obtidas de um banco de dados construído pelos pesquisadores do PRImA ao longo de anos [ref](#). Ele inclui um conjunto de documentos que busca simular um cenário realístico de trabalho, com layouts complexos e diferentes tipos de fontes e formatos de regiões. Isto é importante para avaliar a aplicabilidade do método em situações práticas, onde um controle sobre o formato do conteúdo seria indesejável ou inviável.

No artigo [A Realistic Dataset for Performance Evaluation of Document Layout Analysis] de 2009, os autores apresentam um conjunto de dados com páginas de revistas, artigos científicos diversos, documentos modernos e não apenas históricos.

O conjunto de dados contém não só imagens mas também arquivos XML [The PAGE (Page Analysis and Ground-truth Elements) Format Framework] com metadados como informações bibliográficas (título, autor, publicação), informações das imagens (resolução, bit depth, modelo do scanner), características do layout (número de colunas, variedade de tamanhos de fontes) e informações administrativas (direitos autorais).

Os documentos são digitalizados com um cartão escuro por trás para minimizar a exposição da contra página. Posteriormente um algoritmo analisa possíveis falhas, como rotação do documento, marcando-os para redigitalização. Uma correção automática não é utilizada pois isto pode comprometer a qualidade da imagem.

Uma vez que a imagem foi aceita no banco de dados, inicia-se um processo manual de marcação do ground-truth. Este trabalho deve ser realizado da forma mais precisa possível pois é a base para determinar a corretude dos algoritmos segmentadores. Por se tratar de uma etapa muito custosa, uma ferramenta semi-automática chamada Aletheia é utilizada para agilizar o processo. Esta ferramenta permite a uma pessoa desenhar uma região poligonal em torno de uma região de interesse. Em seguida esta região é automaticamente ajustada pelo software, como se a pessoa estivesse colocando um elástico que aperta a região.

As imagens utilizadas para exemplificação nesta monografia não são as mesmas do banco de dados referido por limitações de licença.

4.2 Algoritmo de Otsu para binarização

Implementar e avaliar cada algoritmo binarizador vai além do escopo deste trabalho, portanto escolhemos um bom algoritmo segundo os resultados obtidos em [2]. Como o próprio artigo aponta, não foi encontrado um método que apresentasse desempenho superior em todas os cenários de teste realizados.

Os requisitos para a escolha foram o da independência de supervisionamento e parametrização. Caso o algoritmo demandasse ajustes específicos de acordo com a imagem, o seu uso comprometeria a promessa de automatização. Dentre a lista de soluções com esta característica, escolhemos o algoritmo de Otsu [3], por ser de fácil implementação e apresentar resultados satisfatórios nos experimentos realizados.

O algoritmo de Otsu encontra um nível de cinza t tal que a soma ponderada da variância dentro das classes $\mathbb{F} = \{x: f(x) \geq t\}$ (foreground) e $\mathbb{B} = \{x: f(x) < t\}$ (background) seja minimizada, ou seja,

$$t = \operatorname{argmin}\{w_{\mathbb{B}}\sigma_{\mathbb{B}}^2 + w_{\mathbb{F}}\sigma_{\mathbb{F}}^2\} \quad (5)$$

onde $w_{\mathbb{B}} = \frac{|\mathbb{B}|}{|E|}$ e $w_{\mathbb{F}} = \frac{|\mathbb{F}|}{|E|}$ são os pesos respectivamente do background e foreground e $\sigma_{\mathbb{B}}^2$ e $\sigma_{\mathbb{F}}^2$ são as variâncias das classes.

A tabela 2 apresenta um passo a passo do algoritmo aplicado à figura 2.

Como podemos notar, para $t = 50\%$ atingimos o menor valor de $\sigma_w^2 \approx 512693389$. Neste caso o limiar coincide com o único vale no histograma, porém isto nem sempre será válido. Algumas imagens não possuem vales bem definidos. Este algoritmo não se baseia no formato do histograma mas sim na coesão intra classe e na separabilidade das classes.

Uma desvantagem de utilizar este algoritmo é a influência da média de todo os pixels da imagem. Isto pode fazer com que o limiar ótimo para a página toda não seja o mesmo que o de dentro de uma janela.

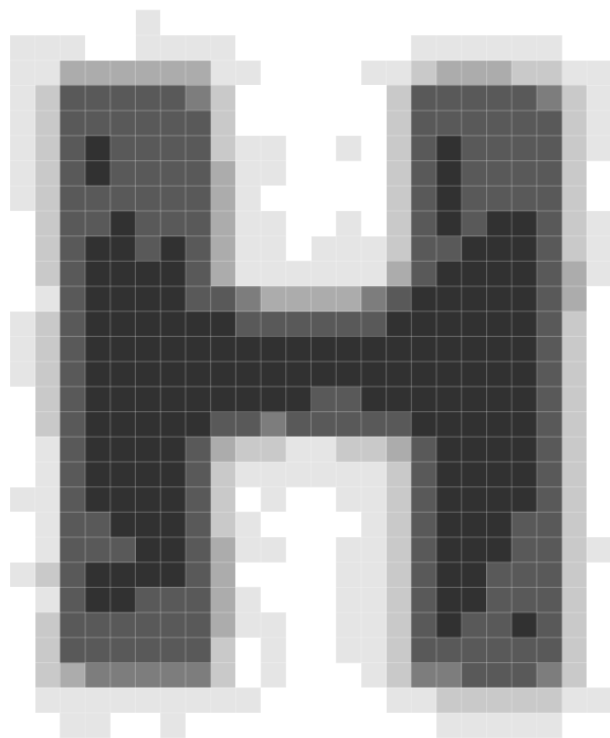


Figura 2: Imagem digitalizada de uma letra H obtida de uma revista.

4.3 Construção do operador

O algoritmo gerador de operadores morfológicos recebe como entrada duas imagens, uma é chamada de original e consiste da imagem branco e preto resultante da binarização e a segunda, chamada de alvo, é uma transformação da imagem original. Esta transformação é feita manualmente e procura demonstrar o conceito de segmentação para o algoritmo de treinamento.

De posse apenas da imagem original, devemos gerar a imagem de destino aplicando uma transformação adequada. Neste ponto exploraremos algumas estratégias diferentes revelando pontos positivos e negativos de se usar esta abordagem.








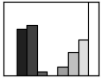
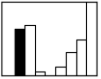
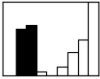
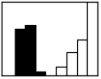
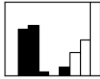
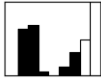

4.3.1 Apagar área de interesse

Nossa primeira estratégia foi apagar as regiões indicadas, pintando-as de branco. Partindo da premissa de que o operador gerado seria o ótimo para este tipo de transformação, poderíamos obter os pixels em regiões do tipo especificado ao fazer a diferença entre a imagem de entrada e a de saída.

4.3.2 Pintar área de interesse

Preenchemos todos os pixels das regiões indicadas, pintando-a completamente. Apagamos todos os demais. Desta forma poderíamos produzir componentes conexos com os pixels de uma certa região.

Tabela 2: Estágios da execução do algoritmo de Otsu.

original	25%	37,5%	50%	62,5%	75%	87,5%
						
						
w_B	0.1931	0.4015	0.4179	0.4532	0.5479	0.6969
μ_B	34.00	51.64	53.61	61.37	83.08	112.56
σ_B^2	7.27	288.58	372.94	1054.08	3128.03	5656.37
w_F	0.80	0.59	0.58	0.54	0.45	0.30
μ_F	184.87	225.55	229.03	233.95	243.79	255.0
σ_F^2	5799.89	1409.01	1006.11	673.10	255.43	0.0
σ_w^2	21495165144	967521582	512693389	595067475	4269882800	17660993341

4.3.3 Manter apenas área de interesse

Apagamos toda a imagem exceto a região de interesse. Desta forma procuramos construir um operador preserve apenas os pixels da área de interesse, porém sem aglutiná-los, como na estratégia anterior.

4.3.4 Aprendizado iterativo

No caso de a abordagem anterior gerar um operador cujo MAE seja insatisfatório, podemos gerar outro operador a partir da imagem processada e a imagem ideal, novamente. Esta técnica funciona como um reforço.

4.3.5 Diminuir resolução

Reduzimos a resolução da imagem reduzindo a complexidade das formas e o tempo de processamento para construção do operador. Esta técnica é especialmente útil para o tratamento de títulos.

4.4 Pós processamento

O operador gerado na etapa anterior é uma aproximação de um operador considerado ideal. A aplicação deste operador produz resultados considerados sub ótimos. Após realizar experimentos com as diferentes estratégias citadas anteriormente e também com diferentes tamanhos de janela, observamos que as imagens produzidas poderiam ser melhoradas com técnicas simples.

Buscando complementar a transformação, adicionamos mais alguns passos, descritos a seguir.

4.4.1 Reconstrução de pixels na área de interesse parcialmente apagada

Caso o operador apague alguns pixels de uma componente conexa, mas não apague a componente toda, recuperamos a componente a partir dos pixels restantes e da imagem original.

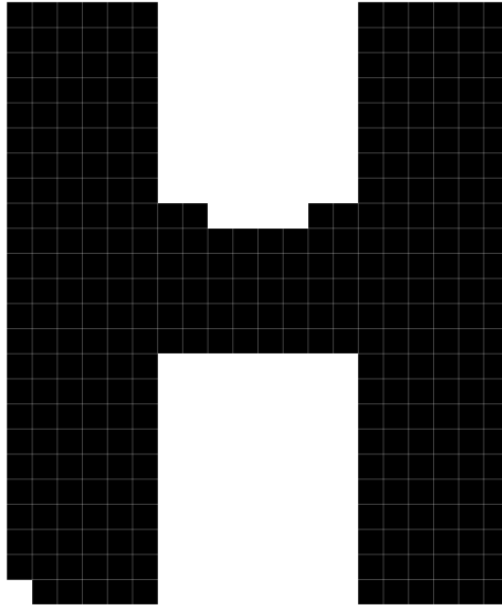


Figura 3: Imagem binarizada com $t = 50\%$.

4.4.2 Unir pixels de uma região em componentes conexas

Para facilitar a definição do polígono que envolve uma região, aplicamos um operador que aglutina as componentes remanescentes.

4.5 Definição dos polígonos delimitadores de regiões

Definir pontos do polígono que envolve cada componente conexa.

4.6 Análise do desempenho

O PRIMa disponibiliza um software que implementa o algoritmo de comparação descrito em [Scenario Driven In-Depth Performance Evaluation of Document Layout Analysis Methods].

A comparação entre a solução obtida e a solução ideal, é dividida nas seguintes situações:

- a região segmentada não possui intersecção alguma com as regiões growndtruth.
- a região growndtruth é totalmente coberta pela região segmentada.
- a região growndtruth é coberta por duas regiões segmentadas, dividindo-a.
- duas regiões growndtruth são unidas por uma única região segmentada.
- uma região growndtruth não é coberta por nenhuma região segmentada, ou seja, esquecida.

A importância de cada erro ou acerto pode ser contextualizado de acordo com o problema em questão. Por exemplo, a união de duas regiões growntruth em uma única região segmentada pode não ser um erro indesejável caso a ordem de leitura seja respeitada.

5 Resultados experimentais

Nesta seção serão descritos os resultados experimentais obtidos.

5.1 Base de dados

Descrever a base de imagens utilizadas.

Qual pré-processamento foi aplicado e com quais parâmetros.

5.2 Experimento A

5.3 Experimento B

5.4 Discussão

- Estatísticas com base no MAE das estratégias propostas.
- Estatísticas do ICDAR.

6 Conclusão

7 Apêndice

7.1 TRIOS

O TRIOS é apenas uma implementação do processo de treinamento de operadores morfológicos. Assim, se for para colocar isso na monografia, acho que deveria estar no Apêndice.

- Imagem enquanto conjunto ou função
- Transformação entre conjuntos
- Exemplos: dilatação, erosão, abertura, fechamento, gradiente, hit-or-miss, sup-gerador.
- Operadores invariantes por translação e localmente definidos.
- W-Operadores.
- Teorema da decomposição canônica (não sei quanto disto eu consigo explicar).
- Conjuntos aleatórios S e I. Caracterização por um processo estacionário local (X, y).
- Otimalidade de Psy com base num operador localmente definido (MAE).
- Algoritmo: Estimativa de $P(y | X)$, decisão, generalização (ISI?).
- Bias-Variance Tradeoff
- Explorando estrutura de Psy? (talvez isso caiba melhor na lista de estratégias a seguir)
- Escolha da janela ótima.
- Operador multi-nível.

Referências

- [1] Rangachar Kasturi, Lawrence O’Gorman, and Venu Govindaraju. Document image analysis: A primer. *Sadhana*, 27(1):3–22, 2002.
- [2] Mehmet Sezgin and Bülent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168, January 2004.
- [3] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979.