Métodos para Seleção de Atributos

Cesar de Oliveira F. Silva

Setembro / 2026

Roteiro

A seleção de variáveis é sempre necessária?

- Alguns algoritmos já fazem seleção de atributos *implicitamente*:
  - Árvores de decisão escolhem quais atributos usar pelos nós internos.
  - Redes neurais aprendem conexões fortes (pesos altos) e fracas (pesos próximos de zero).
  - k-NN com distância ponderada pode dar peso quase zero a alguns atributos.
  - Redes Bayesianas evidenciam variáveis com pouco efeito no modelo.
- Ainda assim, **seleção explícita** costuma ser útil, sobretudo com muitos atributos.

Por que precisamos de seleção de atributos?

- Conjuntos modernos de dados podem ter **milhares** de variáveis (texto, bioinformática, química, imagens).
- Muitos atributos são redundantes ou irrelevantes:
  - Ex.: preço e imposto sobre um produto trazem quase a mesma informação.
  - Ex.: RA do aluno é irrelevante para prever seu coeficiente de rendimento.
- Selecionar variáveis é como arrumar a mochila:

Metáfora

Você não leva tudo o que tem em casa para uma viagem; escolhe o que é mais útil. Seleção de atributos faz o mesmo com o seu dataset.

Objetivos da seleção de atributos

- **Melhorar desempenho** dos algoritmos de aprendizado:
  - Modelos mais rápidos de treinar e de aplicar.
  - Menos risco de *overfitting*.
- **Simplificar modelos** e facilitar interpretação.
- **Remover ruído** e atributos redundantes ou irrelevantes.
- **Manter (quase) o mesmo poder preditivo** com um conjunto de atributos menor.

Seleção x Extração de Atributos

Seleção de atributos

Escolhe um **subconjunto** dos atributos originais. Ex.: manter apenas `pH`, `turbidez` e `temperatura` em um modelo de qualidade da água.

Extração de atributos

Cria **novos** atributos como combinação dos originais. Ex.: componentes principais (PCA), índices espectrais em imagens, etc.

- Seleção preserva o significado físico dos atributos.
- Extração pode gerar atributos mais compactos, mas menos interpretáveis.

Visão geral das abordagens

- **Filtros** (filter):
  - Avaliam atributos *antes* do algoritmo de aprendizado.
  - Usam medidas estatísticas: correlação, χ^2, informação mútua, etc.
- **Embutidos** (embedded):
  - A seleção acontece *dentro* do próprio algoritmo (árvores, LASSO, etc.).
- **Wrappers**:
  - Tratam o algoritmo de aprendizado como *caixa-preta*.
  - Buscam o subconjunto que dá melhor desempenho em validação.

Métodos Embutidos

Determinação de relevância: métodos embutidos

- Em muitos algoritmos, a seleção ocorre **naturalmente**:
  - Árvores de decisão (ID3, C4.5, CART) escolhem atributos com maior **ganho de informação** ou redução de impureza.
  - Modelos lineares com regularização L_1 (LASSO) zeram coeficientes pouco relevantes.
  - Redes neurais podem ter conexões com pesos muito próximos de zero.
- Vantagem: não é preciso rodar um processo separado de seleção.
- Limitação: os atributos escolhidos são ótimos **para aquele modelo**, não necessariamente para outros.

Ganho de informação e entropia

- Entropia mede a **impureza** de um conjunto S:

$$H(S) = -\sum_{i=1}^{C} p_i \log_2 p_i,$$

  onde p_i é a proporção da classe i em S.
- Ao particionar S por um atributo A:

$$\text{Gain}(S, A) = H(S) - \sum_{v \in \text{Valores}(A)} \frac{|S_v|}{|S|} H(S_v)$$

- Quanto maior o ganho de informação, maior o poder de separação do atributo.

Taxa de ganho de informação (Gain Ratio)

- A medida de **ganho de informação** tem viés a favor de atributos com muitos valores distintos.
- Para corrigir isso, usa-se a **taxa de ganho de informação**:

$$\text{SplitInfo}(A) = -\sum_{v \in \text{Valores}(A)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$$

$$\text{GainRatio}(S, A) = \frac{\text{Gain}(S, A)}{\text{SplitInfo}(A)}$$

- Atributos com alto *gain ratio* tendem a ser mais informativos e menos “viciados”.

Wrappers

Wrappers: ideia geral

- O algoritmo de aprendizado é usado como **avalidador** dos subconjuntos de atributos.
- Para cada subconjunto candidato:
  1. Treina-se o modelo usando apenas esses atributos.
  2. Mede-se o desempenho em validação (acurácia, F1, AUC, etc.).
- O melhor subconjunto é aquele que maximiza o desempenho escolhido.

Prós e contras

- **Pró**: normalmente fornecem ótimos conjuntos de atributos para um modelo específico.
- **Contra**: podem ser **muito caros** computacionalmente; risco de overfitting se a validação não for cuidadosa.

Wrappers: estratégias de busca

- **Forward stepwise selection**
  - Começa com conjunto vazio $A = \emptyset$.
  - Adiciona, a cada passo, o atributo que mais melhora o desempenho.
- **Backward elimination**
  - Começa com todos os atributos.
  - Remove, a cada passo, o atributo menos útil.
- **Busca bidirecional**
  - Combina inclusão e exclusão em ciclos.
- Em todos os casos, é essencial usar **validação cruzada** para evitar overfitting.

Filtros Univariados

Filtros: visão geral

- Avaliam cada atributo **individualmente** em relação ao atributo-meta.
- Produzem um **ranqueamento** de atributos.
- Ex.: escolher os Top K atributos com maior pontuação.
- Vantagens:
  - Simples, rápidos, independentes do algoritmo de aprendizado.
- Desvantagens:
  - Não capturam interações entre atributos.

Teste Qui-quadrado (χ^2)

- Útil para **atributos categóricos**.
- Mede associação entre atributo X e a classe Y:

$$\chi^2 = \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}},$$

onde O_{ij} é a frequência observada e E_{ij} a esperada na célula (i, j).

- Quanto maior χ^2, maior a evidência de associação (menor chance de independência).
- Atributos com maiores valores de χ^2 são candidatos mais fortes.

Seleção baseada em correlação

- Para atributos numéricos, é comum usar o coeficiente de correlação de Pearson:

$$r_{A,B} = \frac{\sum_{i=1}^{n}(A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i=1}^{n}(A_i - \bar{A})^2}\sqrt{\sum_{i=1}^{n}(B_i - \bar{B})^2}}$$

- Atributos fortemente correlacionados com a classe são bons candidatos.
- Atributos fortemente correlacionados entre si são **redundantes**.
- Lembrar: correlação não implica causalidade!

Seleção baseada em correlação (CFS)

CFS: Correlation-based Feature Selection

- Em vez de avaliar atributos isolados, o CFS avalia **subconjuntos**.
- Um bom subconjunto deve:
  - Conter atributos altamente correlacionados com a classe;
  - Ter baixa correlação entre os próprios atributos.
- Mérito de um subconjunto S com k atributos:

$$\text{Merit}(S) = \frac{k\,\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}}$$

onde $\bar{r}_{cf}$ é a correlação média atributo–classe e $\bar{r}_{ff}$ é a correlação média atributo–atributo.

CFS: interpretação

- O numerador ($k\bar{r}_{cf}$) mede o **poder preditivo** conjunto.
- O denominador mede a **redundância** entre atributos.
- Quanto maior o mérito, melhor o subconjunto.
- A busca por subconjuntos é feita com heurísticas (como *best-first search*).

Ranqueamento de variáveis: cuidado!

- Variáveis altamente correlacionadas podem ser redundantes, mas:
  - Em alguns casos, manter variáveis redundantes ajuda a reduzir ruído.
  - Dois atributos fracos separadamente podem ser úteis em conjunto.
- Quando um atributo não é selecionado, isso **não** significa que seja inútil para sempre:
  - Pode ser importante em outro modelo;
  - Pode ter relevância para o especialista de domínio.

Seleção de atributos: aspectos relevantes

- Em muitos problemas reais, a seleção:
  - Melhora a precisão do modelo;
  - Simplifica a interpretação;
  - Reduz custo de armazenamento e processamento.
- Ajuda a formular e entender melhor o problema:

Princípio de Economia Científica

"Quanto menos se sabe a respeito de um fenômeno, maior o número de variáveis exigidas para explicá-lo."

Limitações da seleção de atributos

- Em datasets com **muitos** atributos, a seleção pode:
  - Levar a **overfitting** se a validação não for adequada.
  - Ser computacionalmente cara (especialmente wrappers).
- Métodos gulosos podem ficar presos em ótimos locais.
- Atributos descartados podem ser relevantes para:
  - Outras tarefas;
  - Interpretação por especialistas.

Desafios atuais

- Desenvolver heurísticas eficientes para bases com milhares de atributos.
- Combinar seleção de atributos com:
  - Técnicas de regularização (LASSO, Elastic Net).
  - Métodos de ensemble e *stacking*.
- Definir critérios de relevância que:
  - Sejam pouco dependentes do algoritmo de aprendizado;
  - Considerem custos e benefícios distintos para cada atributo.
- Integrar seleção de atributos com conhecimento de domínio (especialistas humanos no ciclo).

Resumo

- Seleção de atributos busca um **subconjunto compacto** e **informativo**.
- Três famílias principais:
  - Filtros (estatísticos), embutidos e wrappers.
- Métodos como ganho de informação, χ^2, correlação e CFS ajudam a ranquear e escolher variáveis.
- É fundamental combinar:
  - Critérios quantitativos (métricas, validação cruzada);
  - Conhecimento de domínio.