# Aprendizado Descritivo

## 1 Definição

Aprendizado descritivo se enquadra na seguinte classificação dos métodos de aprendizado de máquina:

**Aprendizado preditivo:** supervisionado ou não, tem como objetivo predizer dada característica (*target*).

**Aprendizado descritivo:** supervisionado ou não, tem como objetivo obter uma descrição para os dados.

Contrastando, o aprendizado preditivo pretende prever dados futuros ou desconhecidos, enquando o apredizado descritivo prentende trazer uma instrospecção sobre os dados. Tal diferença pode se apresentar de forma tênue. Um exemplo são os algorítmos de agrupamento, que podem ser utilizado em ambas formas. Particularmente, o algorítmo *K-means* se apresenta:

**Preditivo:** *K-means*: $P \to C$

**Descritivo:** *K-means*: $A \to C$

Em aprendizado preditivo, o *K-means* possui como domínio toda a população, ainda que o treinamento seja realizado apenas com uma amostra. Já no aprendizado descritivo, nos limitamos à amostra, afinal ela constitui o alvo total de análise.

## 2 Mineração de itens frequentes

Para um conjunto de dados:

1. Chamamos de itens os elementos do conjunto de variáveis de análise $I$.
2. Um conjunto $X \subseteq I$ é denominado *itemset*
3. O conjunto de todos *k-itemsets* é denotado port $I^{(k)}$.
4. A amostra de transações é denominada por $T$.
5. Cada transação é identificada unicamente por um `tid`.
6. Um conjunto $Y \subseteq T$ é denominado *tidset*.
7. Cada transação consiste de um identificador, e um conjunto de itens: $(\text{tid}, X)$, $X \subseteq I$

Formalmente, um conjunto de dados será uma tripla $(T, I, D)$, onde $D \subseteq T \times I$ é uma relação binária:

$$(t, i) \in D \iff \left[i \in X \text{ na transação } (t, X)\right]$$

Uma transação pode **conter** um *itemset*, e tal relação é definida da seguinte forma:

$$X \subseteq t \iff \forall\, i \in X : (t, i) \in D$$

O conjunto de transações que contém um *itemset* $X$ é denominado **extensão** ou **cobertura** de $X$.
Tal conjunto é definido pela seguinte operação:

$$c : \mathcal{P}(I) \to \mathcal{P}(T)$$
$$c(X) = \big\{ t \in T \mid \forall\, i \in X : (t, i) \in D \big\}$$

Analogamente, o maior conjunto de itens comuns à um *tidset* $Y$ é chamado de **intensão** de $Y$.

$$i : \mathcal{P}(T) \to \mathcal{P}(I)$$
$$i(Y) = \big\{ x \in I \mid \forall\, t \in Y : (t, x) \in D \big\}$$

Desta forma, podemos representar um conjunto de dados de duas formas:

**Horizontal:** o conjunto de transações e suas intesões: $\big\{ \big(t, i(t)\big) \mid t \in T \big\}$

**Vertical :** o conjunto de itens e suas coberturas: $\big\{ \big(x, c(x)\big) \mid x \in I \big\}$

## 2.1 Metodologia

A identificação de regras de associação, em geral, envolve duas etapas:

1. Mineração de conjuntos de itens frequentes
2. Descoberta de regras de associação

Devido à natureza computacionalmente intensa da primeira etapa, nossos estudos a focam.

O limiar que separa os itens frequentes dos infrequentes é chamado de **suporte mínimo**.
O suporte mínimo de um *itemset* é o tamanho de sua cobertura:

$$\sup(X) = \big|c(X)\big|$$

Admite-se também a definição de **suporte relativo**:

$$\mathrm{rsup}(X) = \frac{\big|c(X)\big|}{|T|}$$

### 2.1.1 Algoritmos

O espaço de busca do problema é o conjunto potência do conjunto de itens. Se considerarmos a relação de subconjuntos como uma relação de ordem parcial, temos que o espaço de busca é estruturado como um reticulado. Este reticulado pode ser visualizado como um grafo, onde somente as relações diretas são representadas.

$\rightarrow$ Se $A \subseteq B \wedge |A| = |B| - 1$, então existe uma aresta entre $A$ e $B$.

Assim, a mineração de conjunto de itens frequentes é resolvida por uma busca neste reticulado, seja em largura ou em profundidade. De fato, existem abordagens baseadas em ambas as buscas.

No entanto, a maioria das abordagens compartilham a mesma estrutura de busca:

1. Identificam candidatos navegando o espaço de busca
2. Computam o suporte desses candidatos, descartando os infrequentes

Um algorítmo ingênuo é definido: enumerar cada *itemset* possível, e verificar no conjunto de dados quais transações contêm esse *itemset*.

- A computação do suporte de um *itemset* requer uma passada sobre o conjunto de dados: $\mathcal{O}(|T|)$
- Verificar se uma dada transação contém um *itemset*: $\mathcal{O}(|I|)$
- Portanto, o custo total de computação do suporte é $\mathcal{O}(|I| \cdot |T|)$
- O espaço de busca, por sua vez, é o conjunto potência de $I$.

Logo, a complexidade do algoritmo ingênuo é $\mathcal{O}\big(2^{|I|} \cdot |I| \cdot |T|\big)$.

Vale notar que, devido aos seus tamanhos, a memória principal tipicamente não comporta o conjunto de dados. Tal característica agrava fortemente a ineficiência deste algorítimo, onde o componente $\mathcal{O}(|I| \cdot |T|)$ corresponde à passadas no conjunto de dados.