1. Apresente os principais fatores de complexidade de um algoritmo de geração de regras de associação. Ilustre a sua resposta com exemplos.

Considerando que antes da iteração de um algoritmo de geração de regras, foram definidos os parâmetros/valores mínimos de **suporte** e **confiança**, um algoritmo deste tipo pode ser definido pela iteração de dois estados: A geração de conjuntos de itens (*itemsets*) com o suporte mínimo especificado e, para cada um dos conjuntos levantados, determinar as regras que apresentem uma confiança igual ou superior à confiança mínima indicada.

A primeira iteração, partindo de um conjunto vazio, começa por criar conjuntos de tamanho 1 que respeitem o suporte mínimo e, para cada um destes, cria incrementalmente os seus superconjuntos, começando por um conjunto com 2 itens, depois de 3 itens, e por aí em diante. Cada uma destas iterações, obriga a uma passagem pelo conjunto de dados, para contar o número de ocorrências de cada conjunto, como forma de validar se o mesmo possui ou não o suporte mínimo necessário para ser aceite.

Apenas considerando esta pequena descrição do algoritmo, alguns fatores de complexidade podem ser levantados:

a) Escolha do *threshold* para o valor do suporte mínimo: Se o valor de suporte mínimo for muito baixo, facilmente se vão criar muitos *itemsets* frequentes, o que aumenta o número de candidatos para junção dos superconjuntos em cada iteração. Por exemplo, se o suporte mínimo for 0, todos os *itemsets* possíveis serão considerados frequentes e com isso, a coleção de procura será bastante grande.
No esquema 1, vemos a dimensão que o espaço de procura toma caso o suporte mínimo seja por exemplo o valor 2 e no esquema 2, podemos ver a simplificação que se realiza ao espaço de procura caso o suporte mínimo seja 5.

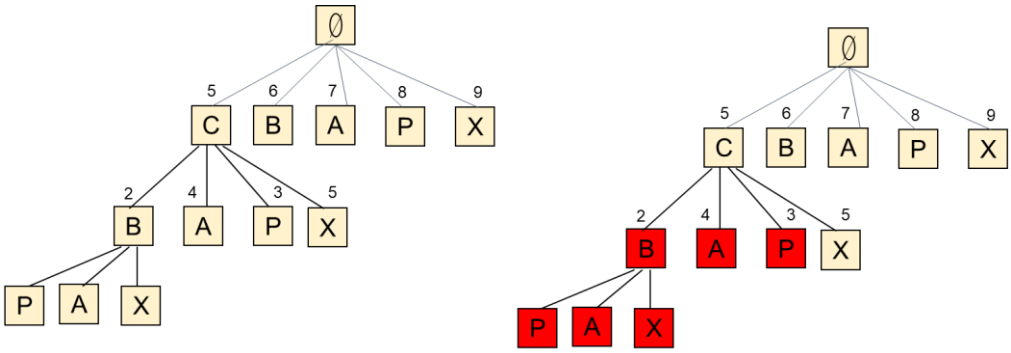

Figura 1 - Exemplo da influencia no suporte mínimo no crescimento do espaço de procura

b) Número de itens (atributos) no *dataset* inicial, pois mais espaço será necessário para armazenar a contagem e a representação dos conjuntos gerados;

c) Tamanho geral do *dataset* devido ao número de passagens que o algoritmo realiza sobre os dados para a contagem do suporte de cada conjunto. Se considerarmos que os dados em vez de contidos num ficheiro estiverem numa base de dados, os custos destas sucessivas transições em BD de grandes dimensões será ainda mais pesados.

2. Descreva as principais fraquezas da medida de interesse *confiança*. Apresente exemplos de regras para ilustrar essas fraquezas. Sugira formas de ultrapassar os problemas levantados pelo usa da *confiança*.

Seja a regra **R: A -> C**, a medida de confiança apresenta-se como um estimador de certeza da inferência feita pela regra, indicando a % de vezes que o consequente **C** surge junto com o antecedente **A.**

Apresenta como fraqueza o facto de favorecer regras entre itens independentes, porque não deteta esta possível situação entres os itens de uma regra. Estas situações ocorrem quando uma regra produz uma inferência que não se traduz numa situação de **causalidade**, mas sim naquilo que se pode chamar de uma coincidência, ou seja, existe uma forte co-corrência entre os itens da regra, mas isto não implica necessariamente que "se **A** então temos **C**".

Numa análise formal, a confiança mede $\mathrm{P(C|A)} = \frac{P(C \cup A)}{P(A)}$ (probabilidade de **C** sabendo **A**).

No cenário em que **C** é independente de **A**:

$$P(C \cup A) = P(C)*P(A) \text{ e pontanto } \frac{P(C \cup A)}{P(A)} = \frac{P(C)*P(A)}{P(A)} = P(C).$$

Ou seja, quando existir um item **C** com elevada probabilidade de ocorrência nos registos, qualquer regra que o relacione com itens independentes, terá valor de confiança alto e igual a P(C), independentemente do antecedente. Estas regras serão aceites pelo critério de confiança mínima mas não traduzem um conhecimento útil no contexto.

Se P("salmão") = 0.8, e a compra de salmão é independente da compra de "cartolinas", então poderá existir uma regra "cartolinas -> salmão" com confiança de 0.8, mas que não traduz nenhuma causalidade porque os dois produtos são independentes e não se relacionam.

Uma solução poderia ser usar em conjunto com a medida de confiança a medida ***Lift***, pois esta medida mede a distância para a independência entre o antecedente e o consequente. Assim, se o valor se *Lift* for superior a 1, a relação entre a regra é boa e podemos dar credibilidade a um valor de confiança alto. Se o valor do lift for menor ou igual a 1, então o antecedente não influencia o consequente e não podemos guiar o estudo da regra pelo valor alto da confiança.

3. Considere o dataset de disciplinas (student_courses.bas). Apresente para este dataset exemplos de regras redundantes, produtivas e significativas. Comente os exemplos obtidos/usados.

1. Uma regra é redundante se uma regra mais geral existe com uma confiança igual ou superior à regra inicial. Seja **A -> C**, se existe um **A´** subconjunto de **A**, então a regra **A -> C** é redundante se:

   confiança(**A´->C**) >= confiança(**A -> C**).

(usando o comando ***caren adult.data 0.05 0.5 -s, -ovrt -Hclass -Att***)

Exemplo de regra redundante: A linha 25 representa uma regra com um antecedente mais pequeno e subconjunto do antecedente da regra existente na linha 26, tendo ambas a mesma confiança e suporte. Desta forma devemos optar por usar a regra mais genérica da linha 25 e descartar a regra redundante da linha 26, onde a especificação com mais itens no antecedente não aumentou a confiança da regra produzida.

```
25  Sup = 0.05055   Conf = 0.67653   class=>50K   <--   education=Bachelors  &  relationship=Husband  &  sex=Male
26  Sup = 0.05055   Conf = 0.67653   class=>50K   <--   education=Bachelors  &  relationship=Husband  &  education.num=13
```

2. Uma regra é produtiva quando se introduz um conceito de *improvement* face à especificação do antecedente. Se este improvement de uma regra com um antecedente mais especifico for superior a um improvement mínimo, então a regra é aceite e produtiva.

(usando o comando ***caren adult.data 0.05 0.5 -s, -ovrt -imp0.00001 -Hclass -Att***)

Exemplo de uma regra produtiva: A regra na linha 139, é mais especifica que a regra na linha 141, ou seja, o seu antecedente é um superconjunto do antecedente da linha 141, mas a confiança aumenta com esta especificação. Por isso, a regra é considerada produtiva.

```
139  Sup = 0.05801   Conf = 0.98181   class=<=50K   <--   marital.status=Never-married  &  hours_per_week=40  &  sex=Female  &  capital.gain=0  &  capital.loss=0
140  Sup = 0.05313   Conf = 0.98073   class=<=50K   <--   marital.status=Never-married  &  hours_per_week=40  &  sex=Female  &  native_country=United-States  &  capital.gain=0
141  Sup = 0.05970   Conf = 0.98033   class=<=50K   <--   marital.status=Never-married  &  hours_per_week=40  &  sex=Female  &  capital.gain=0
```

Um outro exemplo desta situação, onde a especificação do antecedente da regra da linha 297 levou a um aumento da confiança face à regra mais geral da linha 298.

```
297  Sup = 0.05322   Conf = 0.93980   class=<=50K   <--   marital.status=Divorced  &  sex=Female  &  workclass=Private  &  capital.loss=0
298  Sup = 0.05485   Conf = 0.93803   class=<=50K   <--   marital.status=Divorced  &  sex=Female  &  workclass=Private
299  Sup = 0.05381   Conf = 0.93740   class=<=50K   <--   occupation=Adm-clerical  &  sex=Female  &  workclass=Private  &  capital.gain=0
```

3. Uma regra é significativa quando alem do conceito de improvement mínimo, se aplica um teste de análise de significância estatística.

(usando o comando ***caren adult.data 0.05 0.5 -s, -ovrt -imp0.00001 -Hclass -Att -fisher***)

```
16  Sup = 0.05153   Conf = 0.68322   class=>50K   <--   education.num=13  &  marital.status=Married-civ-spouse  &  native_country=United-States
17  Sup = 0.05153   Conf = 0.68322   class=>50K   <--   education=Bachelors  &  marital.status=Married-civ-spouse  &  native_country=United-States
18  Sup = 0.05117   Conf = 0.68167   class=>50K   <--   occupation=Exec-managerial  &  marital.status=Married-civ-spouse
19  Sup = 0.05206   Conf = 0.67827   class=>50K   <--   education.num=13  &  marital.status=Married-civ-spouse  &  race=White
20  Sup = 0.05206   Conf = 0.67827   class=>50K   <--   education=Bachelors  &  marital.status=Married-civ-spouse  &  race=White
```