Classificação de Documentos

Wendel Melo
Faculdade de Computação
Universidade Federal de Uberlândia

Recuperação da Informação

Classificação de Documentos

- Dada uma coleção de documentos e um conjunto de categorias, a missão é classificar cada documento em uma ou mais categorias apropriadas;
- Quando há limitação de classificar cada documento em uma única categoria, o problema é dito ser de *classificação única* (*rótulo único*);
- Quando é permitido que cada documento pertença a mais de uma categoria, temos um problema de *multiclassificação* (*multirótulo*);
- O problema de classificação de documentos tem sido tratado com técnicas de *Aprendizado de Máquina* (*Machine Learning*)

Aprendizado de Máquina

- O *aprendizado de máquina* é um sub-ramo da Inteligência Artificial que estuda e desenvolve algoritmos voltados a assimilar (aprender) padrões presentes em conjuntos dados;

- A atividade de assimilação de padrões presentes em dados (processo de aprendizagem) é denominada como *treinamento*;

- A ideia é que, após treinar com um determinado conjunto de dados, uma ferramenta possa usar os padrões aprendidos para fazer predições sobre dados ainda não vistos;

Aprendizado de Máquina

- Exemplos de aplicação de aprendizado de máquina:
  - Aproximador de funções desconhecidas;
  - Classificação de dados;
  - Reconhecimento de imagens: objetos, pessoas, caracteres...
  - Sistemas de previsão;
  - Correção de dados imprecisos ou com ruídos em imagens, áudio, sinais, etc;
  - Reconhecimento de padrões em geral.

Tipos de Treinamento

- **Supervisionado**: Temos um conjunto de amostras de dados onde já se conhece a resposta ideal. Nesse caso, o algoritmo ajusta seus parâmetros para que, em média, apresente respostas satisfatórias para essas amostras de dados;

- **Não supervisionado**: O treinamento deve ser feito então com amostras de dados cuja resposta ideal não está disponível. Esse tipo de treinamento ocorre, por exemplo em problemas de agrupamento (*clustering*);

- **Semisupervisionado**: O treinamento é realizado com um pequeno números de amostras com resposta ideal conhecida e um grande número de amostras sem conhecimento da resposta.

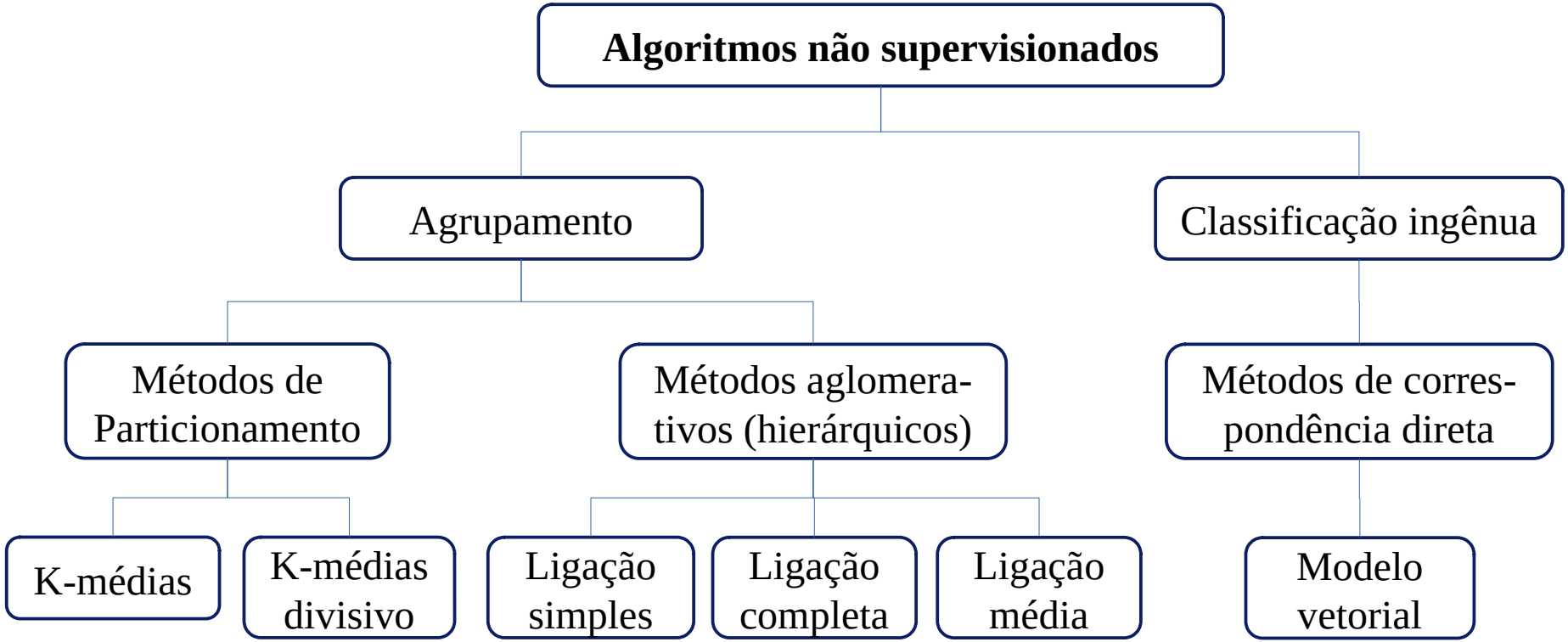
Classificação de Documentos

- Algoritmos de aprendizado supervisionado podem ser sair bem ao classificar de documentos, dependendo da qualidade do conjunto de amostras de treinamento;
- Todavia, em muitos casos, não há dados conhecidos a priori para realizar o treinamento supervisionado.
- Nessas situações, é necessário apelar para algoritmos de aprendizado não supervisionado;

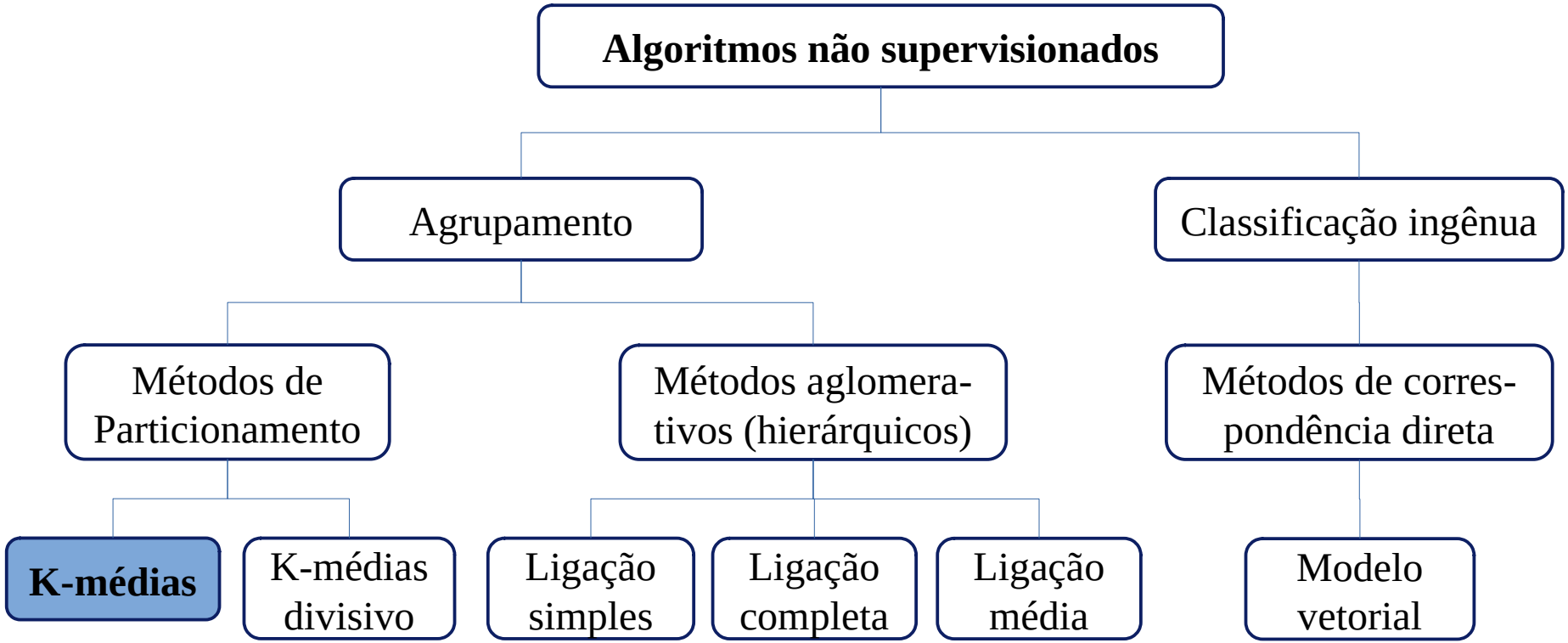
Classificação de Documentos

- Os algoritmos de aprendizado não supervisionado podem ser subdivididos em duas classes:
  - **Agrupamento (*clustering*)**: quando não se conhece nem mesmo as categorias de classificação. Nesse caso, é preciso agrupar documentos por alguma forma de similaridade (ou várias);
  - **Classificação ingênua**: quando se conhece as categorias de classificação, mas a falta de dados para treinamento torna a classificação mais “simplista”.

Classificação de Documentos – Algoritmos não supervisionados



Classificação de Documentos – Algoritmos não supervisionados



Algoritmo de agrupamento k-médias

- O algoritmo k-médias parte da representação dos documentos como pontos (vetores) em um espaço algébrico;
- Um esquema de ponderação como TF-IDF (ou qualquer outro) pode ser utilizado para representar documentos como vetores;
- Embora não se tenha informações sobre quais seriam as categorias de classificação, na versão básica do algoritmo o número *k* de categorias deve ser conhecido;
- O algoritmo então criará k grupos de documentos observando a proximidade entre seus vetores de representação.

Algoritmo de agrupamento k-médias

1) Seja $x^{(j)}$ o vetor de representação do documento j;

2) Selecione *k* documentos e coloque cada um em um agrupamento distinto. Esses docs são usados como centroides iniciais de cada um dos agrupamentos A_i.

3) Atribua cada um dos N docs ao agrupamento de centroide mais próximo.

4) Recalcule o centroide de cada agrupamento *i* como sendo a média dos vetores que foram colocados em *i*.

$$C_i = \frac{1}{|A_i|} \sum_{x^{(j)} \in A_i} x^{(j)}$$

5) Volte para o passo 3 até que os centroides não mudem mais.

Algoritmo de agrupamento k-médias

- Ilustração do algoritmo k-médias com k = 2 (2 categorias):

Cada ponto x^j é a representação vetorial do documento *j*

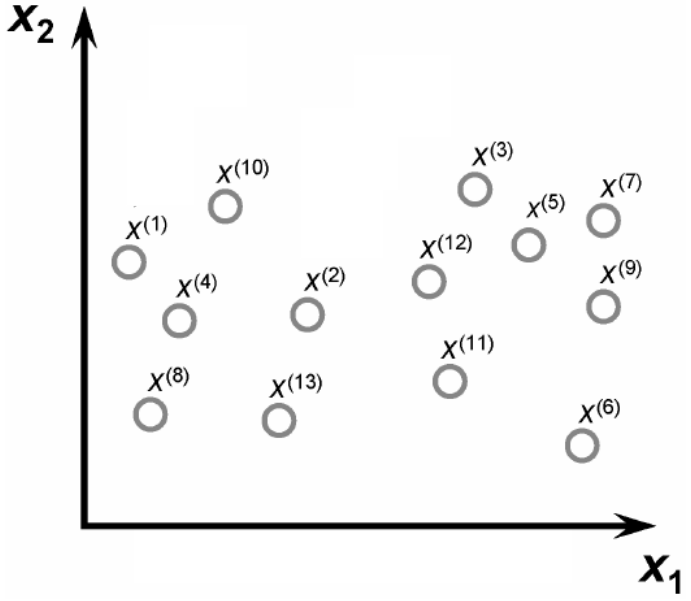


Figura adaptada de [2]

Algoritmo de agrupamento k-médias

- Ilustração do algoritmo k-médias com k = 2 (2 categorias):

O vetor C_i representa o agrupamento *i*.

Os vetores C_i dos k agrupamentos são inicializados com os vetores dos primeiros k documentos. (Pode-se escolher k documentos aleatórios também para inicializar os C_i)

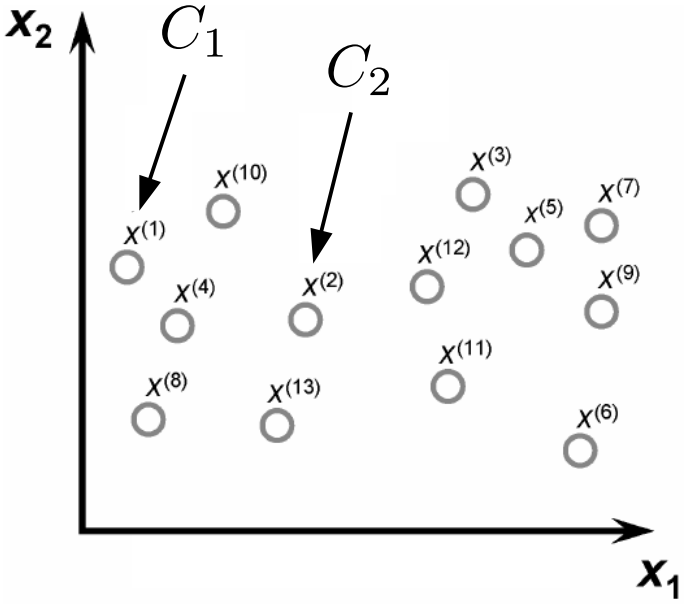


Figura adaptada de [2]

Algoritmo de agrupamento k-médias

- Ilustração do algoritmo k-médias com k = 2 (2 categorias):

Montam-se agrupamentos atribuindo cada amostra ao grupo do vetor C_i mais próximo

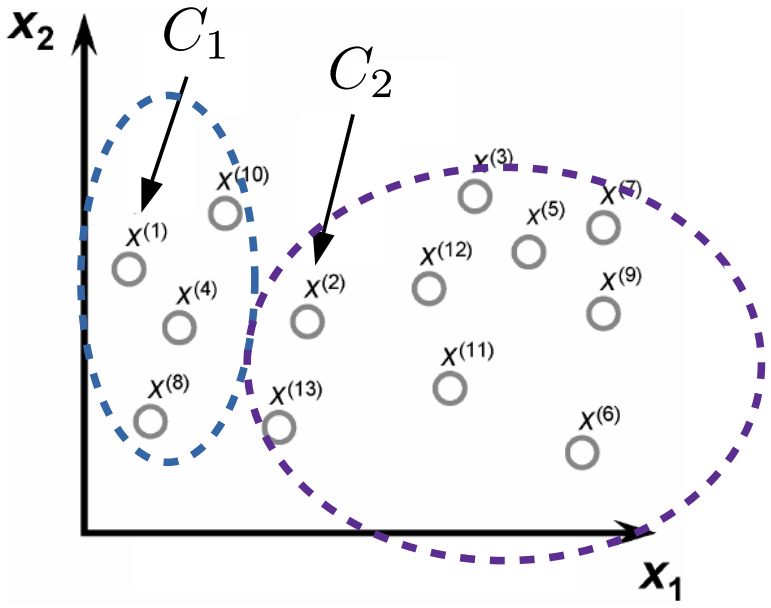


Figura adaptada de [2]

Algoritmo de agrupamento k-médias

- Ilustração do algoritmo k-médias com k = 2 (2 categorias):

Cada vetor C_i é movido para o centroide do agrupamento (média das coordenadas dos membros do agrupamento)

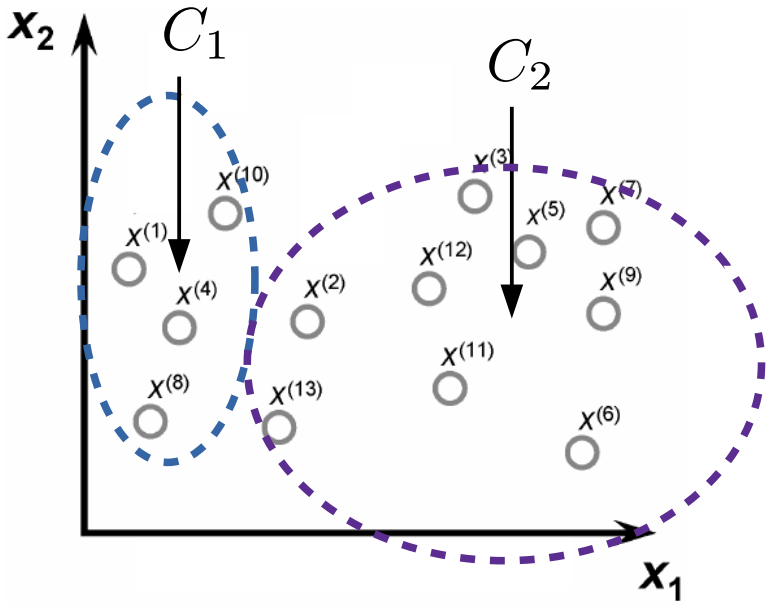


Figura adaptada de [2]

Algoritmo de agrupamento k-médias

- Ilustração do algoritmo k-médias com k = 2 (2 categorias):

Os agrupamentos são remontados, novamente com o critério de pertencer ao grupo do vetor C_i mais próximo

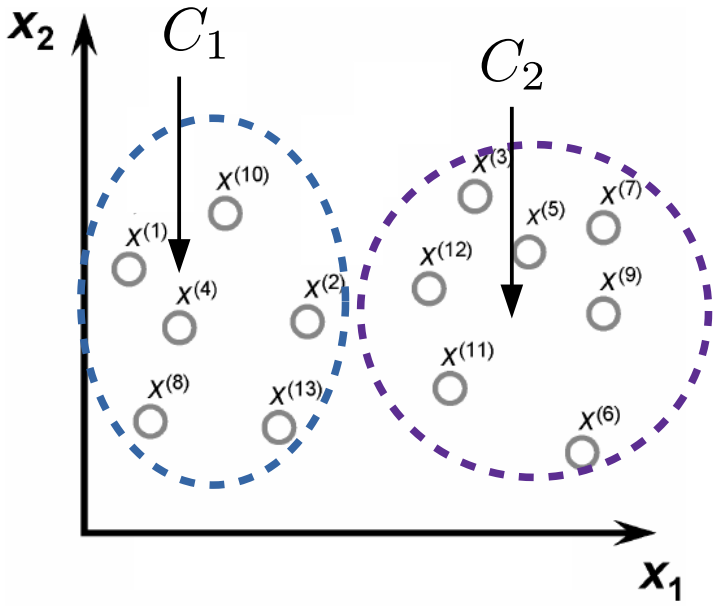


Figura adaptada de [2]

Algoritmo de agrupamento k-médias

- Ilustração do algoritmo k-médias com k = 2 (2 categorias):

Cada vetor C_i é novamente movido para o centroide do agrupamento (média das coordenadas dos membros do agrupamento)

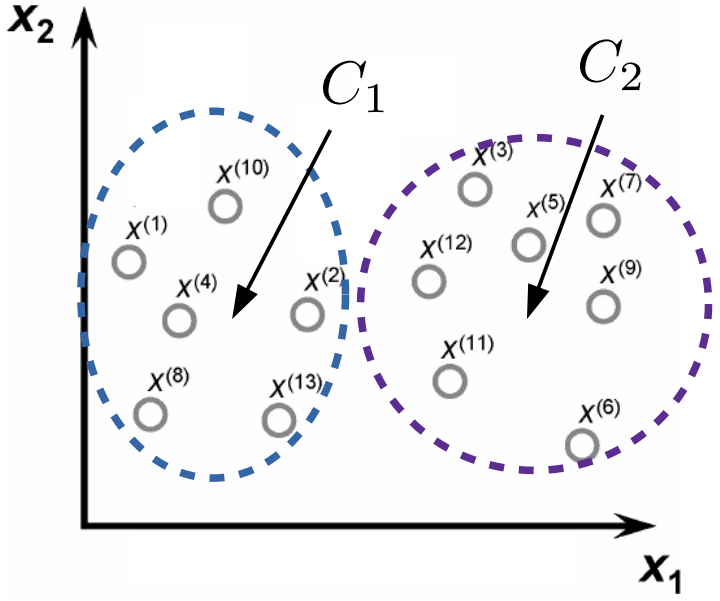


Figura adaptada de [2]

Algoritmo de agrupamento k-médias

- Ilustração do algoritmo k-médias com k = 2 (2 categorias):

Os agrupamentos são remontados segundo o critério do C_i mais próximo.

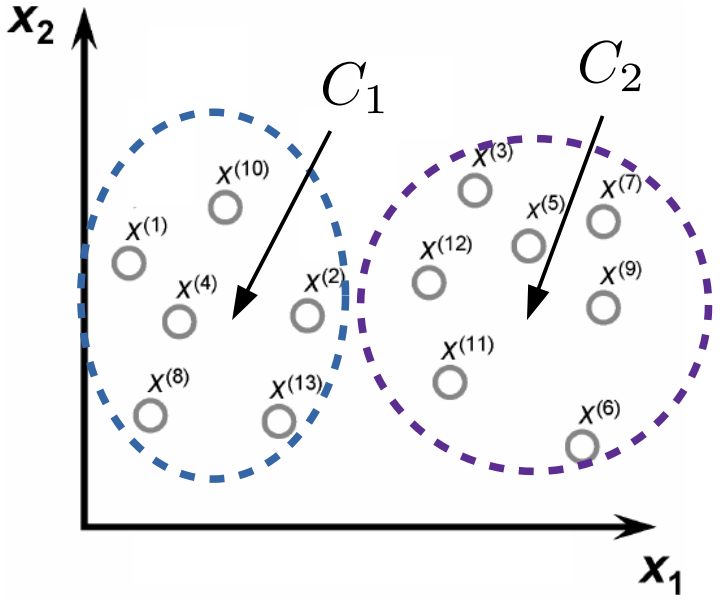


Figura adaptada de [2]

Algoritmo de agrupamento k-médias

- Ilustração do algoritmo k-médias com k = 2 (2 categorias):

O algoritmo pára quando não há mudanças nas composições dos agrupamentos de uma iteração para outra.

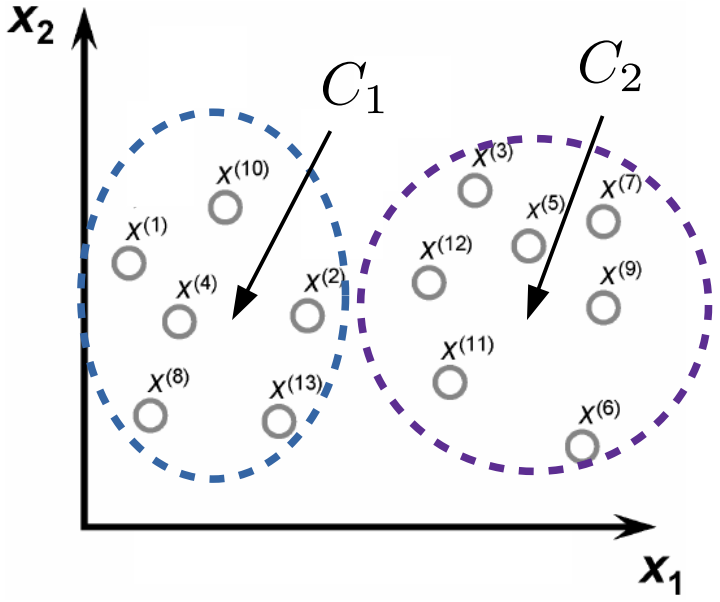


Figura adaptada de [2]

Algoritmo de agrupamento k-médias

- Para determinar o agrupamento em cada doc será incluído, no lugar da distância geométrica (euclidiana), pode-se usar o conceito de similaridade do modelo vetorial (cosseno do ângulo entre vetores).

- Nesse caso, será preciso considerar o inverso do cosseno, pois quanto menor a similaridade, maior a distância:

$$dist(x, C) = \frac{1}{cos\theta} = \frac{||x||||C||}{\sum_{p=1}^{T} x_p C_p}$$

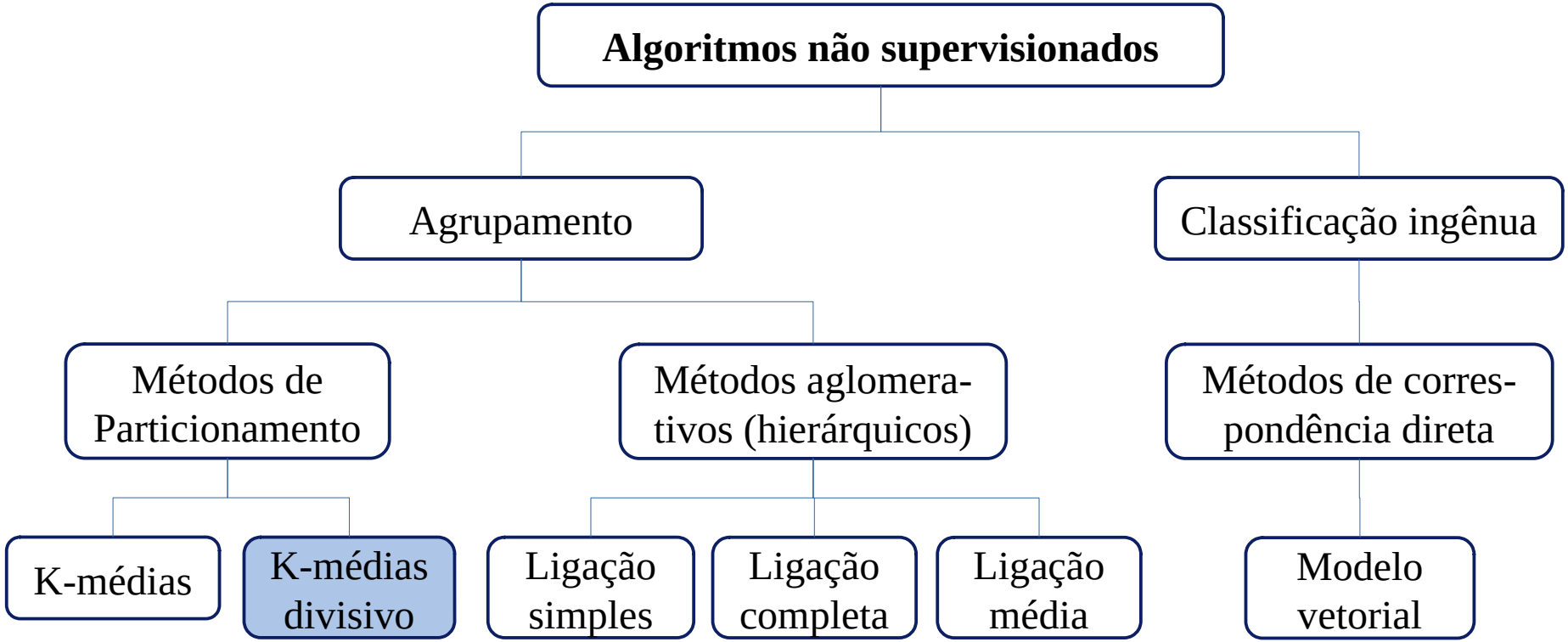
Algoritmo de agrupamento k-médias

- A versão inicial do algoritmo k-médias é denominada *em lote*, pois cada centroide é atualizado após a atribuição de **todos** os documentos a algum agrupamento;

- Uma versão alternativa do algoritmo usa atualização *online*, onde, tão logo **um** documento é incluído em algum agrupamento, seu centroide é imediatamente atualizado para incluir também o vetor do último documento incluído;

- Testes tem mostrado melhor desempenho da versão *online* do algoritmo.

Classificação de Documentos – Algoritmos não supervisionados



Algoritmo de agrupamento k-médias divisivo

- A ideia do k-médias divisivo (*bisecting k-means*) é construir uma hierarquia de agrupamentos onde um agrupamento é dividido em dois a cada passo;
- Isso é feito aplicando-se k-médias repetidamente com k = 2;
- Dessa forma, após aplicação, teremos uma espécie de árvore de agrupamentos, com classes e subclasses.

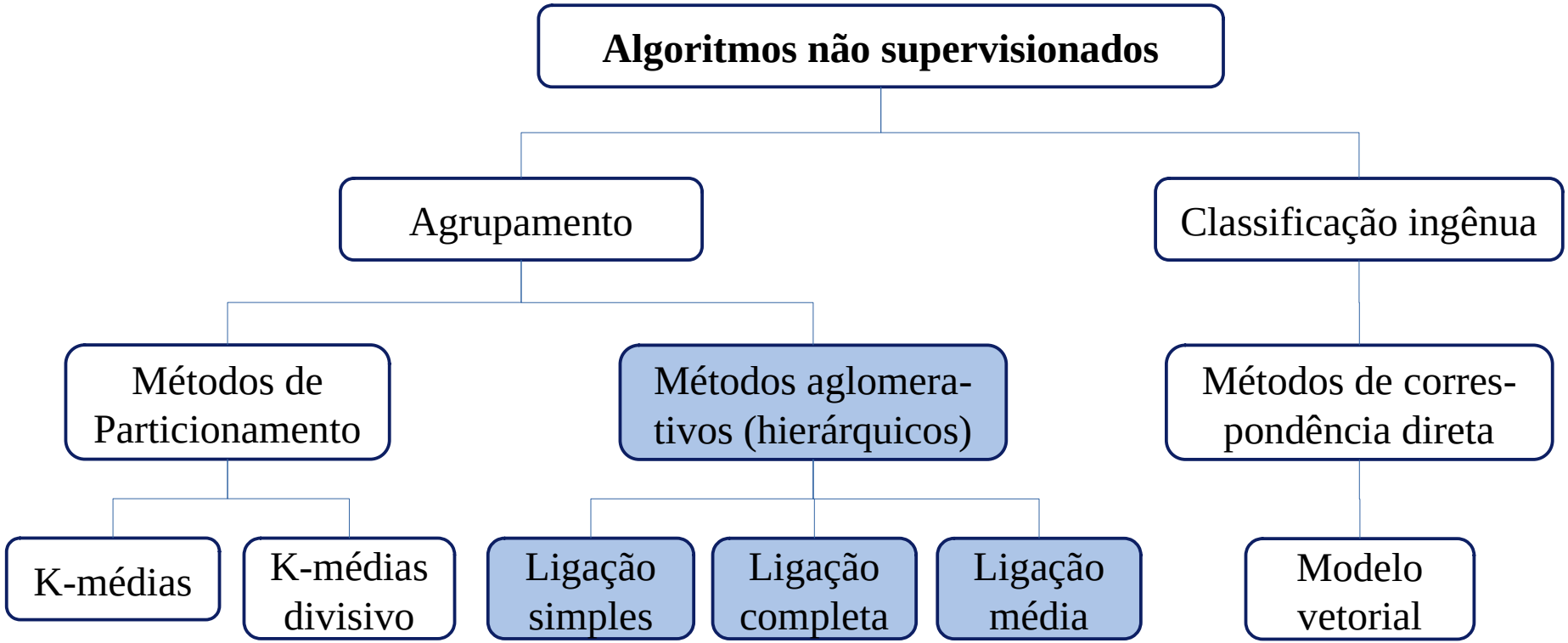
Algoritmo de agrupamento k-médias divisivo

1) Atribua todos os documentos a único agrupamento;

2) Aplique o k-médias, com k= 2 ao agrupamento maior

3) Se há algum agrupamento maior que um determinado parâmetro *p*, volte ao passo 2.

Classificação de Documentos – Algoritmos não supervisionados



Agrupamento aglomerativo hierárquico

- Algoritmos nessa categoria geram uma hierarquia de agrupamentos;
- Podem operar tanto decompondo agrupamentos em agrupamentos menores quanto unindo agrupamentos em outros maiores.

Agrupamento aglomerativo hierárquico – Algoritmo (da base em direção ao topo)

1) Dada a coleção de *N* docs, calcule uma matriz de similaridade *NxN* entre os docs (ex: com a fórmula de similaridade do modelo vetorial);

2) Crie um agrupamento para cada doc. Esses agrupamentos podem ser vistos como folhas da árvore de hierarquia de classificação;

3) Encontre o par de agrupamentos mais similar e una-os em um único agrupamento. Esse novo agrupamento é representado como um nó de nível acima na árvore que contém todos os agrupamentos;

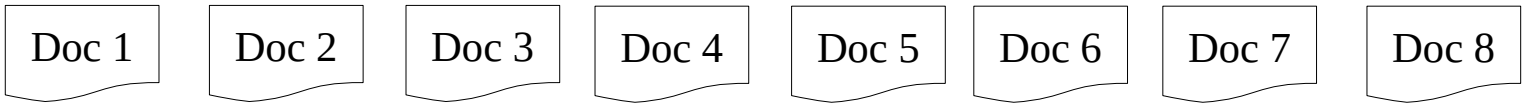
4) Recalcule as similaridades entre o novo agrupamento e os demais;

5) Volte para o passo 3 até que todos os docs estejam em um único agrupamento de tamanho *N* (o nó raiz da árvore de agrupamentos).

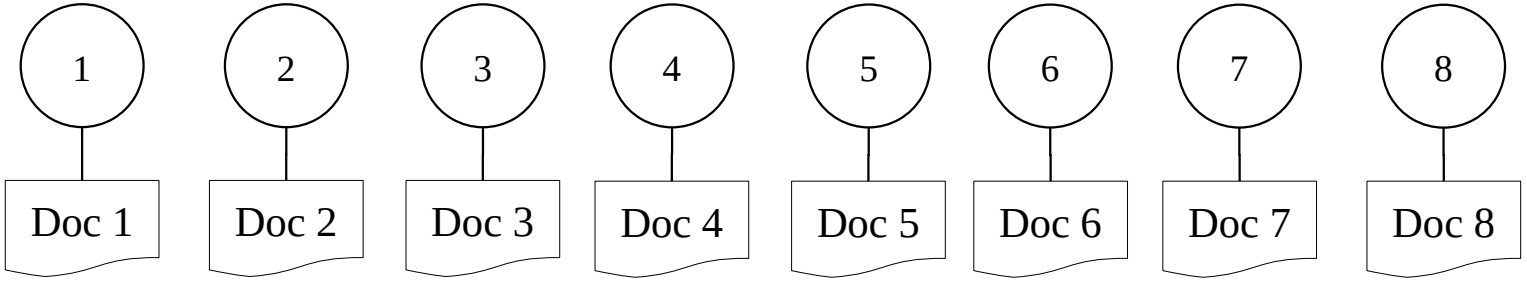
Agrupamento aglomerativo hierárquico

- Observe que o algoritmo constrói uma árvore de agrupamentos, isto é, uma hierarquia de agrupamentos.

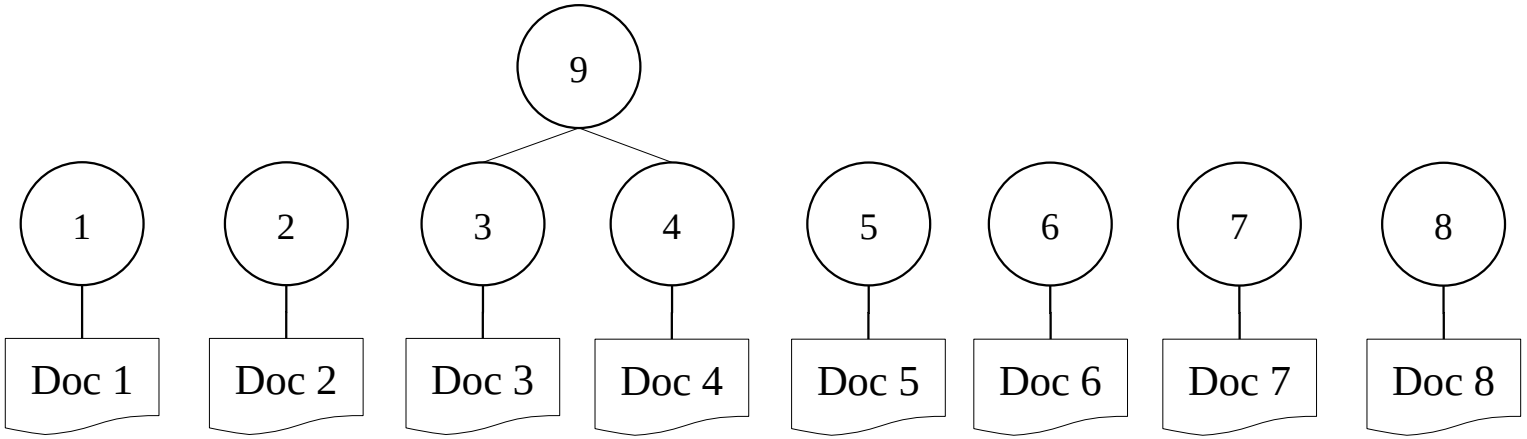
Agrupamento aglomerativo hierárquico



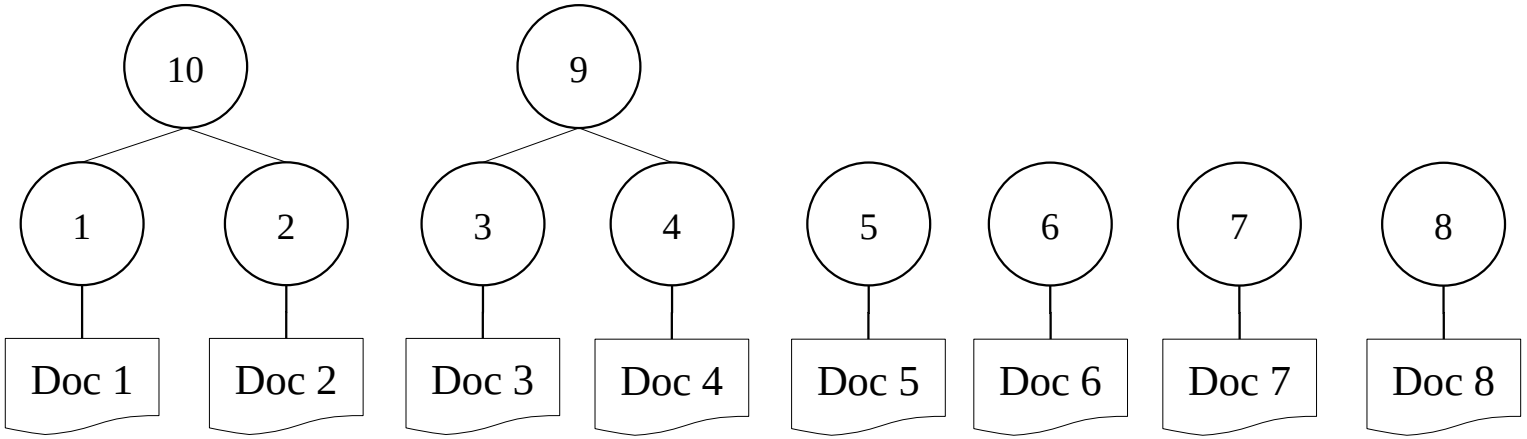
Agrupamento aglomerativo hierárquico



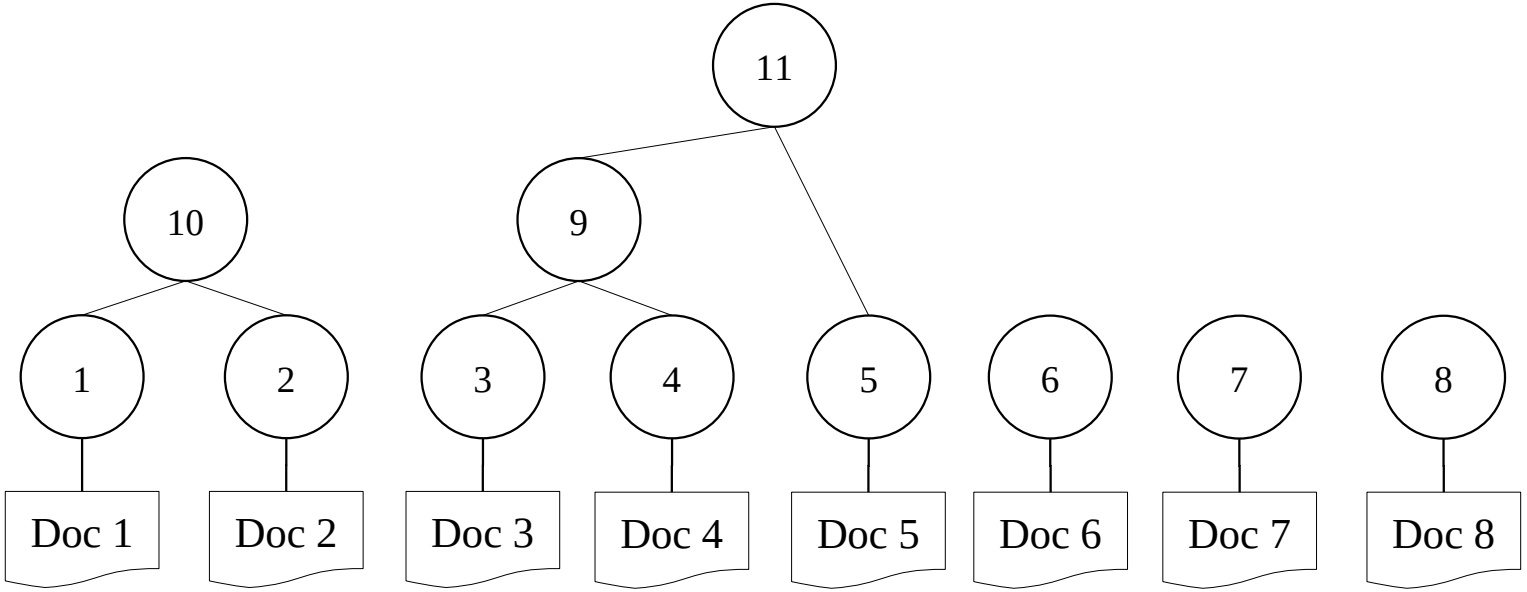
Agrupamento aglomerativo hierárquico



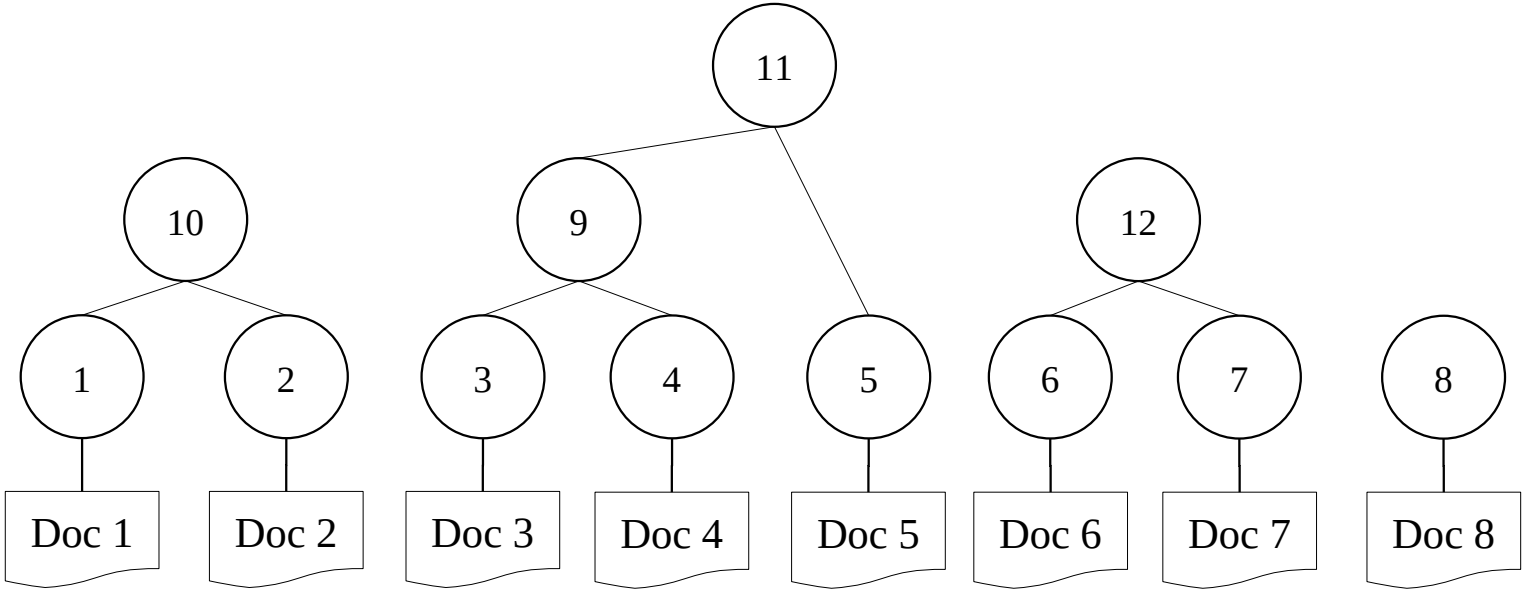
Agrupamento aglomerativo hierárquico



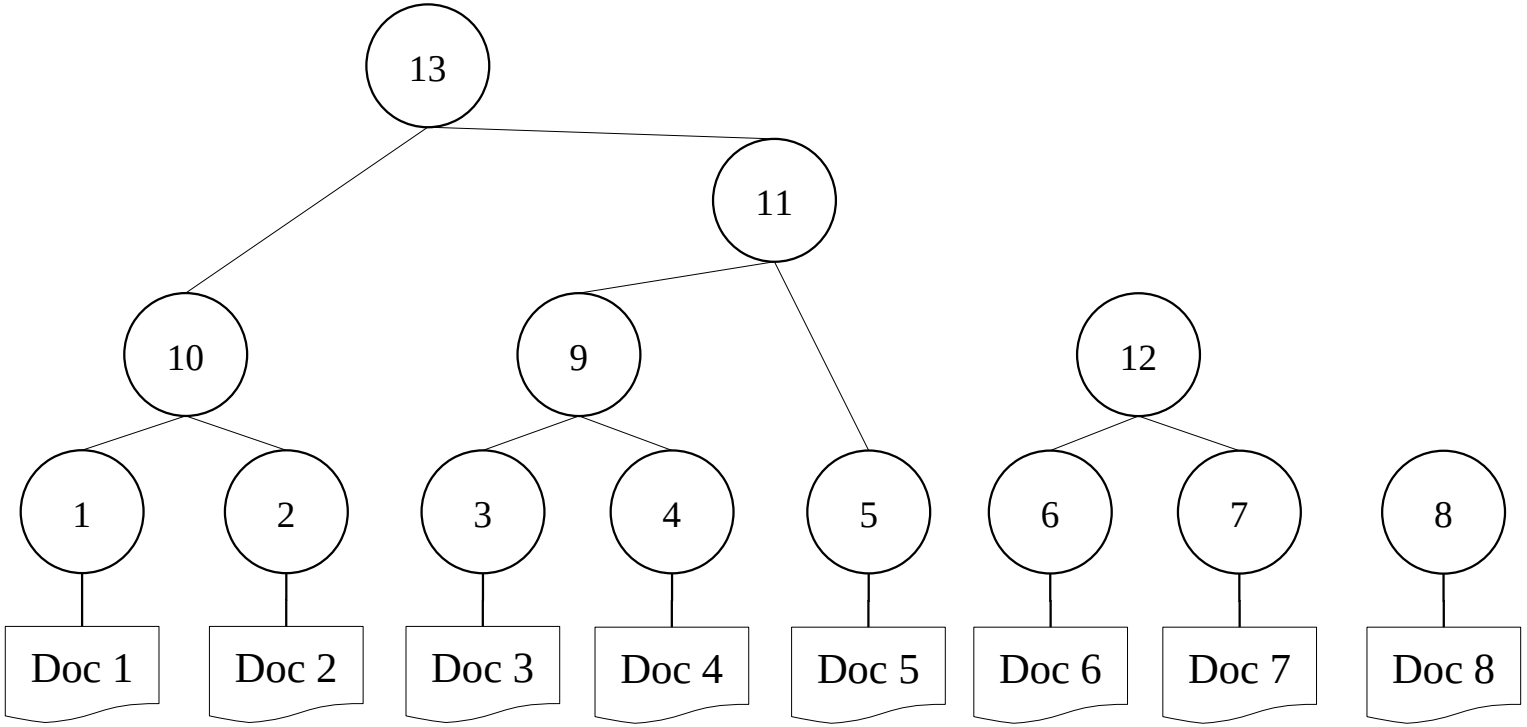
Agrupamento aglomerativo hierárquico



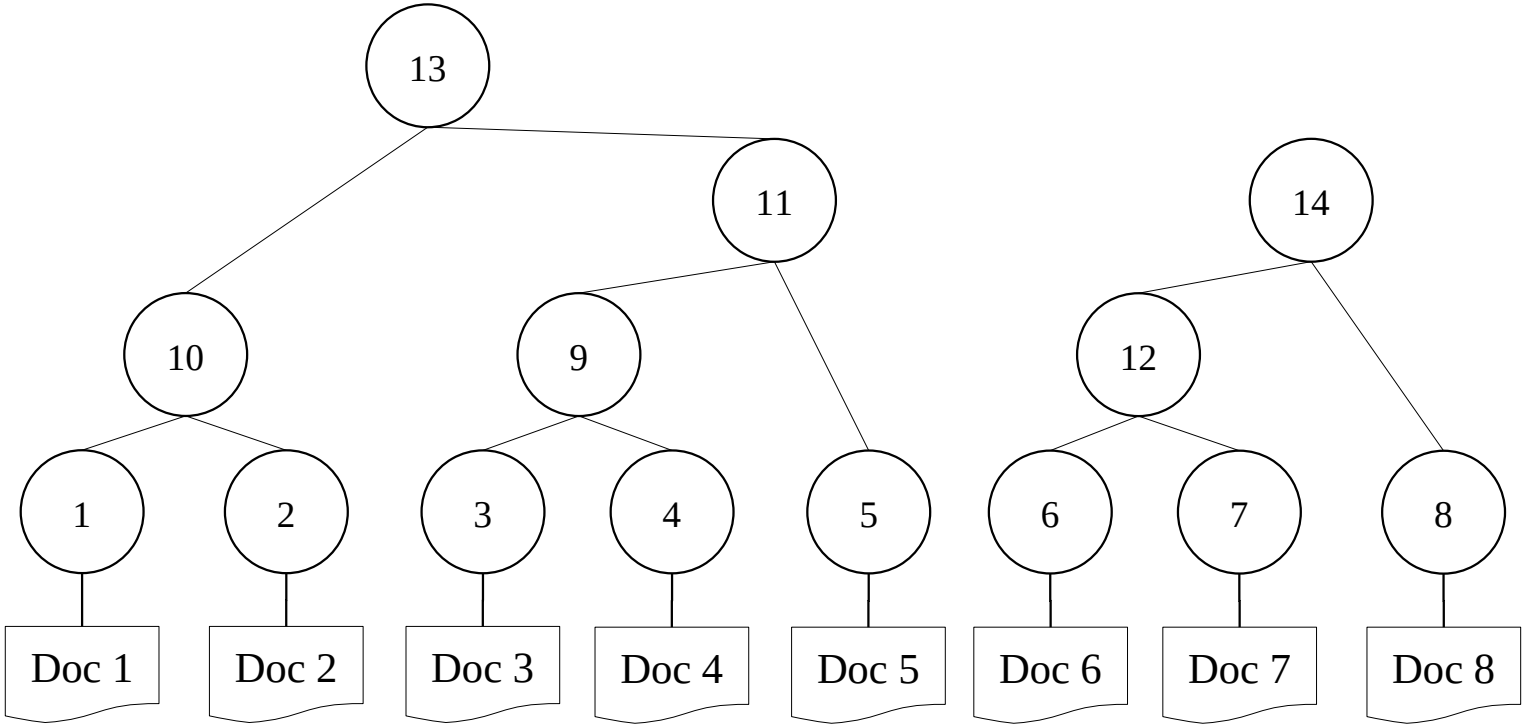
Agrupamento aglomerativo hierárquico



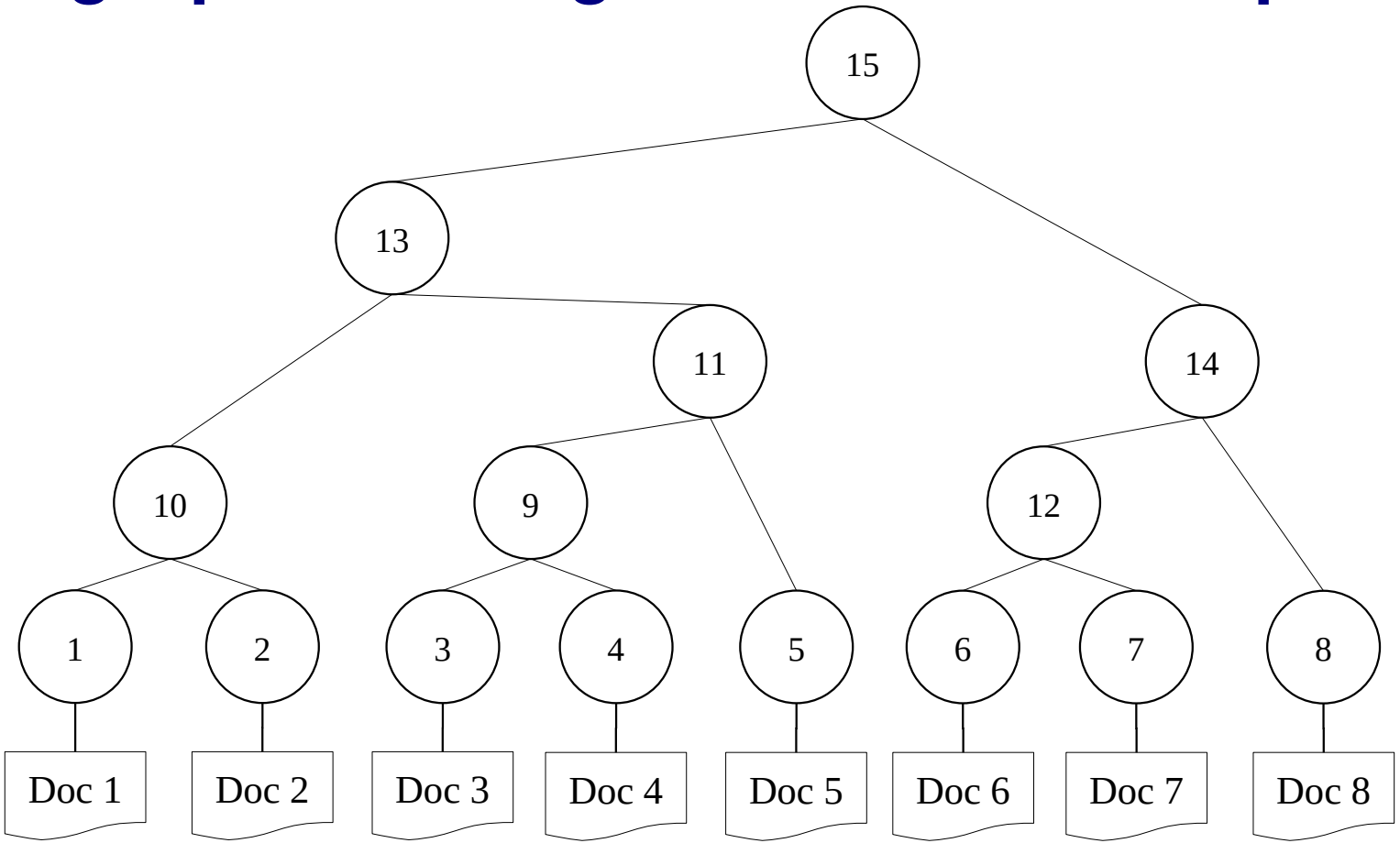
Agrupamento aglomerativo hierárquico



Agrupamento aglomerativo hierárquico



Agrupamento aglomerativo hierárquico



Agrupamento aglomerativo hierárquico

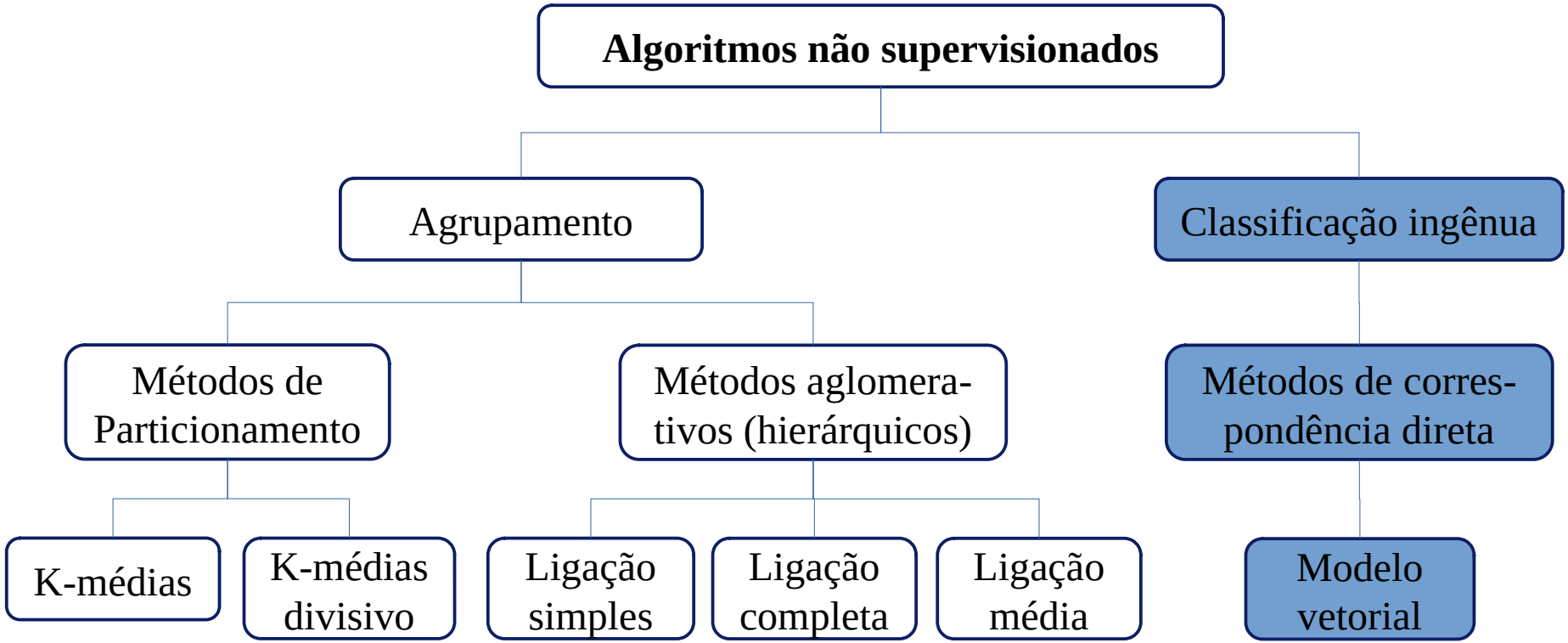
- O passo 4 calcula similaridade entre diferentes agrupamentos. De modo geral, existem 3 formas de realizar este cálculo:

- **Ligação simples**: a distância entre agrupamentos é igual a menor distância (ou maior similaridade) entre qualquer documento de um agrupamento e qualquer documento do outro agrupamento;

- **Ligação completa**: como o anterior, mas agora considerando a maior distância, no lugar da menor, entre os agrupamentos;

- **Ligação média**: como os anteriores, mas agora considerando a distância média entre os agrupamentos;

Classificação de Documentos – Algoritmos não supervisionados



Classificação Ingênua

- Se as categorias de documentos forem pré-definidas, mas não há amostras para realizar treinamento supervisionado, pode-se utilizar o **Algoritmo de Classificação Ingênua**;

- A ideia é definir, para cada categoria, um conjunto de *palavras-chaves* ou *rótulos*, que são utilizados para compor um vetor de pesos para a mesma;

- Assim, para cada doc, calcula-se a similaridade entre seu vetor de pesos e vetor da categoria, segundo a fórmula do modelo vetorial;

- Associa-se então cada doc com a(s) categoria(s) de maior(s) similaridade(s).

Classificação Ingênua

- Para melhorar os resultados, pode-se usar apenas um subconjunto de termos para representar vetores de categorias e documentos (*redução de dimensionalidade*);
- Para isso é necessário selecionar os termos de interesse segundo alguma abordagem específica;
- A Classificação Ingênua pode render bons resultados para coleções verticais focadas em áreas específicas do conhecimento, onde há uma hierarquia com operações de especialização/generalização;
- Em coleções muito amplas, a Classificação Ingênua pode produzir resultados ruins;

Algoritmos de Treinamento Supervisionado

- Os algoritmos de treinamento supervisionado partem de uma conjunto de dados de entrada para a qual já se conhece a resposta ideal. Em nosso caso, documentos cujas categorias já foram definidas por seres humanos;

- As amostras de dados com resposta conhecida devem ser particionadas em dois conjuntos:

  - **Conjunto de treinamento**: amostras que serão efetivamente utilizadas para treinar o algoritmo (algo em torno de 70%);

  - **Conjunto de teste ou validação**: Amostras que serão utilizadas para testar o algoritmo treinado (algo em torno de 30%).

Algoritmos de Treinamento Supervisionado

- Uma vez que o algoritmo foi treinado e o ajuste foi validado, ele está pronto para receber novos documentos e classificá-los automaticamente;

- É importante frisar que o treinamento supervisionado tem por objetivo ajustar parâmetros internos do algoritmo para que ele produza resposta para as amostras de treinamento o mais próximo possível da resposta conhecida;

- Após o treinamento, os valores dos parâmetros internos não são mais alterados, e o ajuste é utilizado para classificar os documentos cuja resposta não é conhecida.

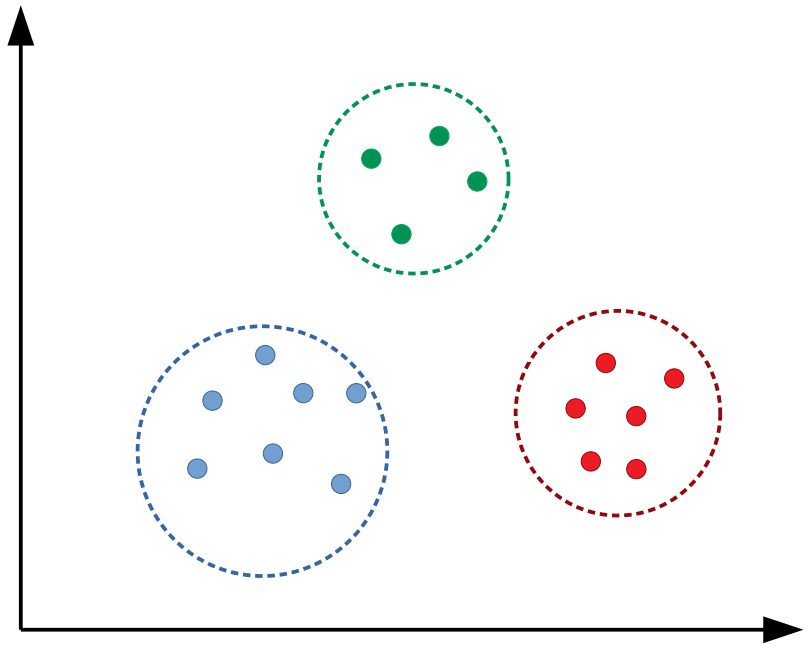
Algoritmos de Treinamento Supervisionado

- Tipos de algoritmos de treinamento supervisionado para classificação de documentos:
  - Árvores de Decisão;
  - Vizinhos mais próximos: (*K-Nearest Neighbor*);
  - Realimentação de relevância: Rochio;
  - Bayes ingênuo;
  - Máquinas de vetores de suporte;
  - Ensemble;
  - ...

K Vizinhos mais Próximos (k-NN)

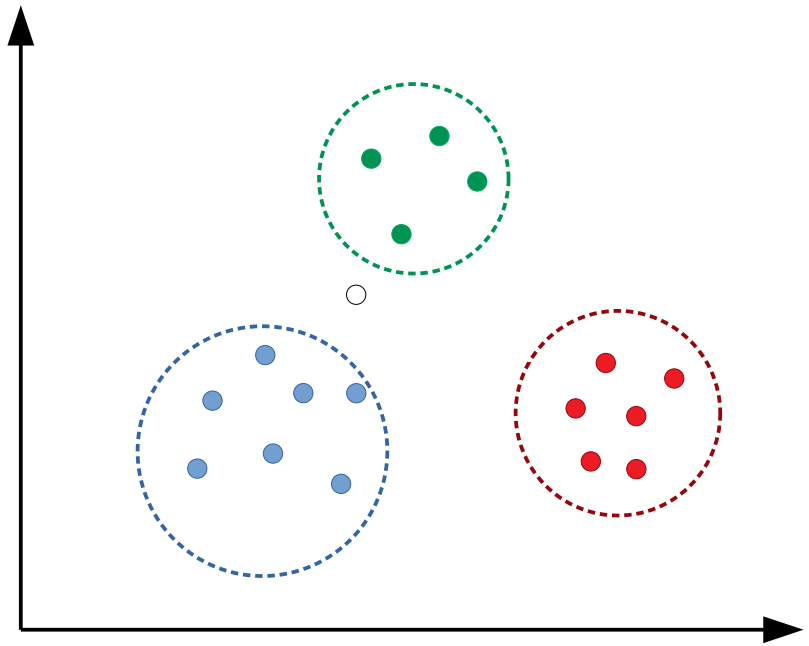
- O algoritmo *K Vizinhos mais Próximos* (K-NN) parte da representação dos documentos como vetores numéricos;
- Cada documento é então classificado baseado na classificação dos *k* vizinhos mais próximos dentre os docs do conjunto de treinamento (k é parâmetro a ser ajustado)

K Vizinhos mais Próximos (k-NN)



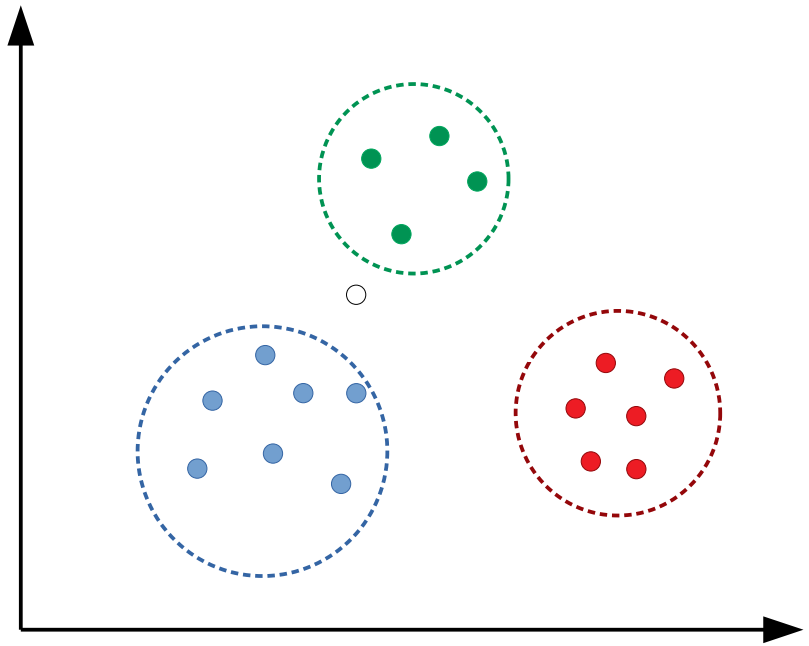
- Suponha que temos 3 categorias de documentos (verde, azul e vermelho) e a seguinte disposição do conjunto de treinamento;

K Vizinhos mais Próximos (k-NN)



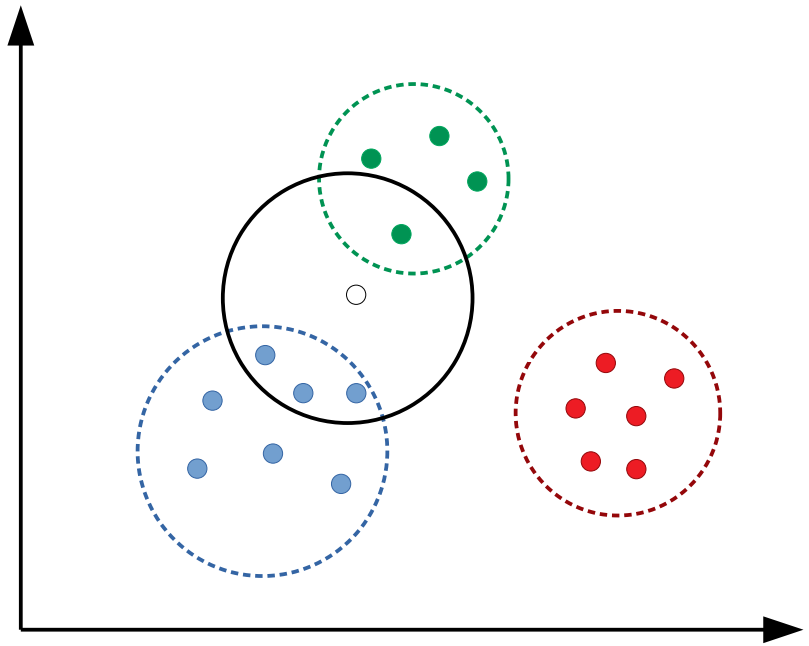
- Suponha que temos 3 categorias de documentos (verde, azul e vermelho) e a seguinte disposição do conjunto de treinamento;
- Temos agora que classificar um novo documento cuja categoria não se conhece;

K Vizinhos mais Próximos (k-NN)



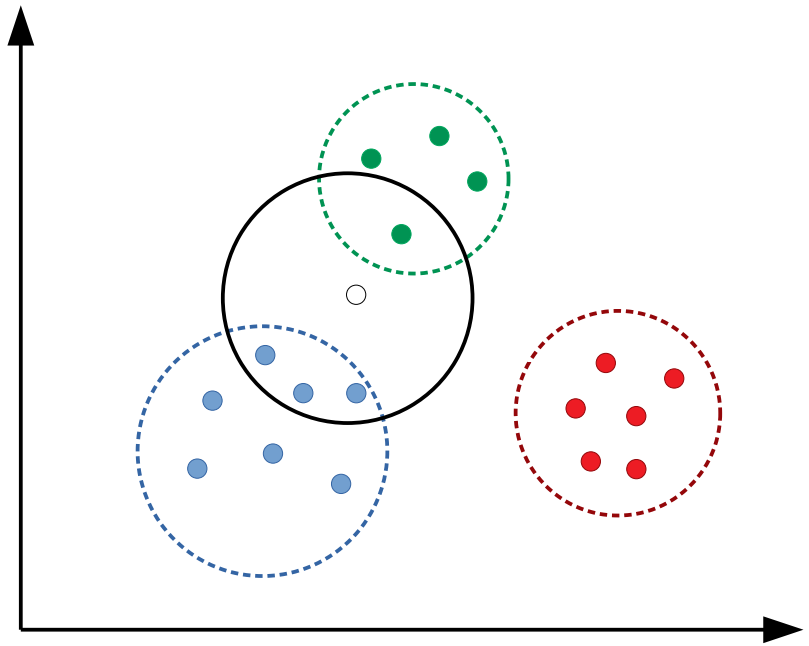
- Suponha que temos 3 categorias de documentos (verde, azul e vermelho) e a seguinte disposição do conjunto de treinamento;
- Temos agora que classificar um novo documento cuja categoria não se conhece;
- Vamos usar K-NN com k = 4, isto é, a classificação será dada pelos 4 vizinhos mais próximos.

K Vizinhos mais Próximos (k-NN)



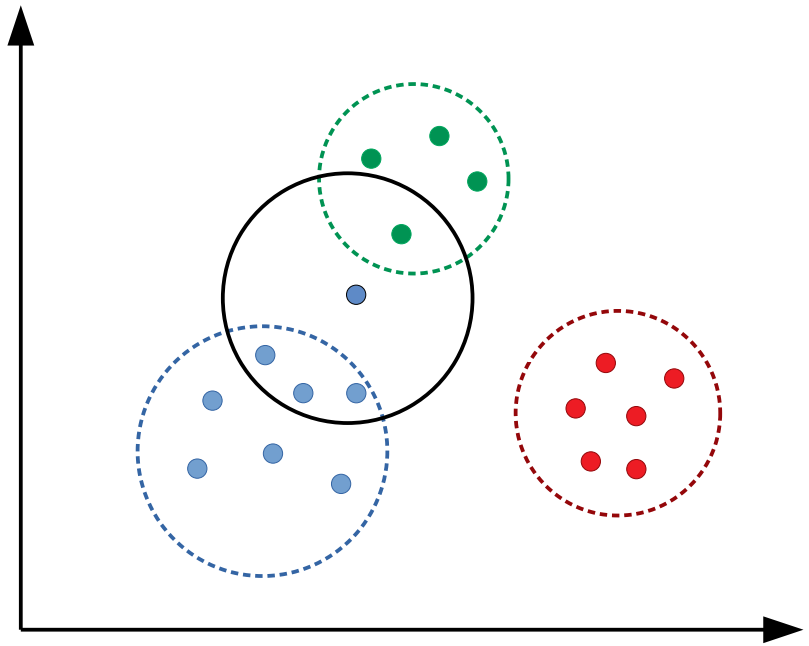
- Suponha que temos 3 categorias de documentos (verde, azul e vermelho) e a seguinte disposição do conjunto de treinamento;
- Temos agora que classificar um novo documento cuja categoria não se conhece;
- Vamos usar K-NN com k = 4, isto é, a classificação será dada pelos 4 vizinhos mais próximos

K Vizinhos mais Próximos (k-NN)



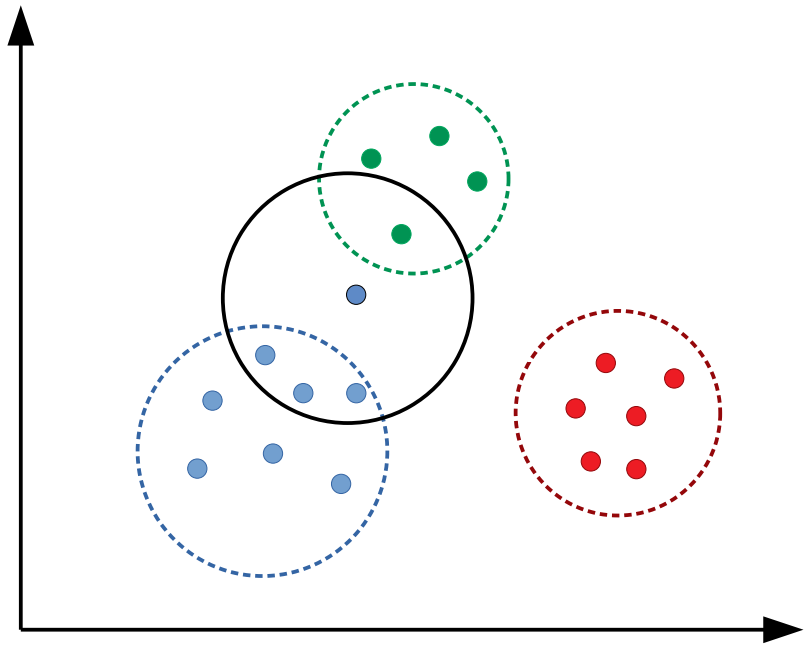
- Dos 4 vizinhos mais próximos do novo documento, 3 pertencem a classe azul;

K Vizinhos mais Próximos (k-NN)



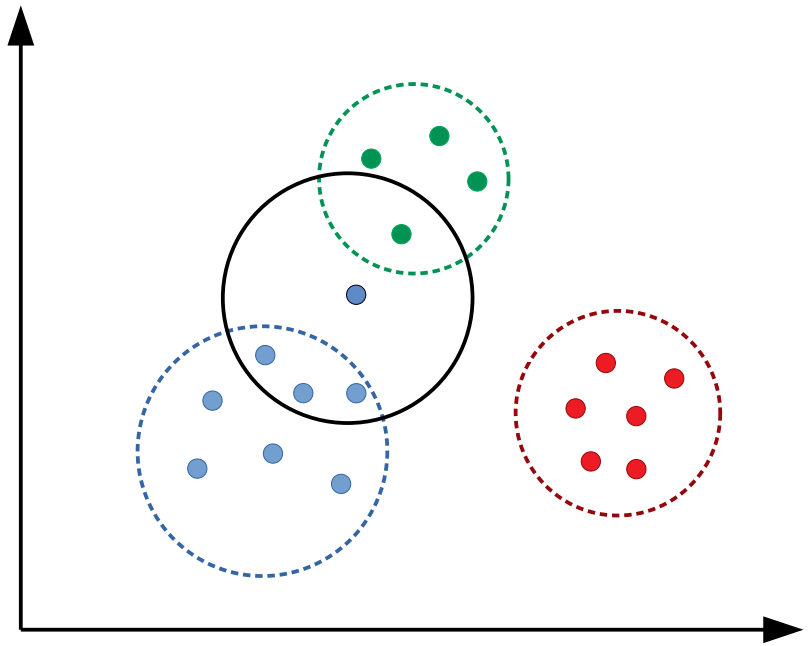
- Dos 4 vizinhos mais próximos do novo documento, 3 pertencem a classe azul;
- Isso sugere atribuir o novo documento à classe azul;

K Vizinhos mais Próximos (k-NN)



- Dos 4 vizinhos mais próximos do novo documento, 3 pertencem a classe azul;
- Isso sugere atribuir o novo documento à classe azul;
- Críticos e *haters* podem argumentar, todavia que o documento mais próximo do novo doc é da classe verde!

K Vizinhos mais Próximos (k-NN)

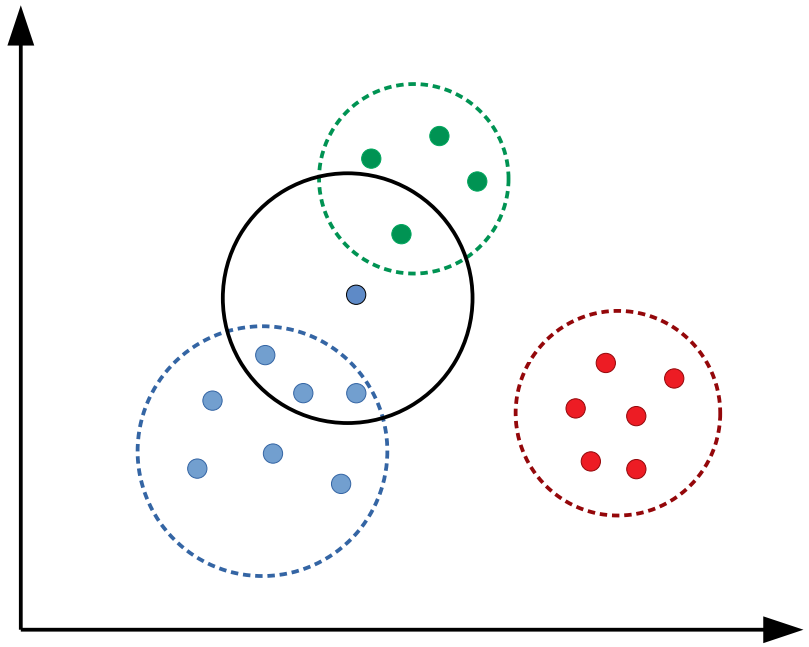


- Podemos então, para cada par doc *j* e categoria *c*, atribuir um escore de pertencimento $S_{j,c}$:

$$S_{j,c} = \sum_{t \in N_k(j)} sim(j,t) \times T(t,c)$$

- Onde $N_k(j)$ é o conjunto dos *k* vizinhos mais próximos do doc *j*, $sim(j, t)$ é a similaridade entre os docs *j* e *t*, e $T(t, c)$ é 1 se o doc *t* está na classe c, e 0 caso contrário

K Vizinhos mais Próximos (k-NN)

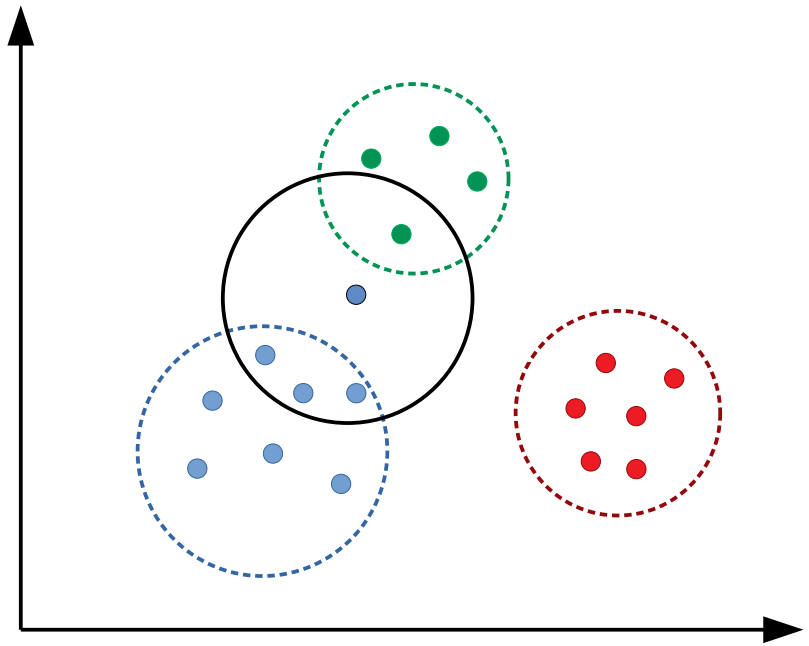


- Podemos então, para cada par doc *j* e categoria *c*, atribuir um escore de pertencimento $S_{j,c}$:

$$S_{j,c} = \sum_{t \in N_k(j)} sim(j,t) \times T(t,c)$$

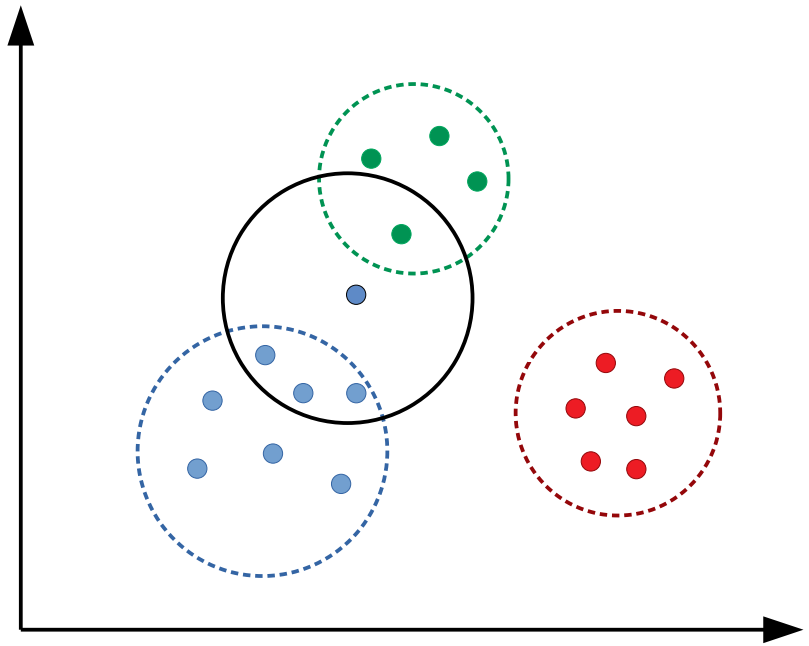
- A similaridade entre os docs j e t pode ser calculada usando a fórmula do cosseno do modelo vetorial!

K Vizinhos mais Próximos (k-NN)



- Com essa estratégia do escore de pertencimento a cada classe, o algoritmo pode classificar o documento em mais de uma categoria;
- Em nosso exemplo, o novo documento teria um escore de pertencimento a classe azul e outro escore pertencimento a classe verde.

K Vizinhos mais Próximos (k-NN)



- O algoritmo KNN é simples de se entender e de se implementar;
- Todavia, o ajuste do melhor valor para o parâmetro k pode não ser muito intuitivo.

Classificador de Rocchio

- Inspirado no método de Rocchio para expansão de consulta (realimentação de relevância);
- Novamente, parte-se da representação dos documentos como vetores numéricos (viva a álgebra linear)!
- O método trabalha com a ideia de que termos que estejam presentes em documentos que sabidamente são de uma classe p fornecem realimentação positiva, ao passo que termos presentes nos documentos fora da classe p fornecem realimentação negativa.

Classificador de Rocchio

- Assim o centroide $\bar{c}_p$ da categoria *p* é calculado de modo a se aproximar da média dos vetores dos documentos da categoria *p*, e, simultaneamente, se afastar da média dos vetores fora da categoria *p*:

$$\bar{c}_p = \frac{\beta}{|C_p|} \sum_{d_j \in C_p} d_j - \frac{\gamma}{N - |C_p|} \sum_{d_j \notin C_p} d_j$$

- Onde C_p é o conjunto de vetores dos docs da categoria *p*, d_j é o vetor de representação do doc *j*, *N* é o nº total de docs, e β e γ são parâmetros que modelam a importância dos docs dentro e fora da classe, respectivamente, na construção do centroide.

Classificador de Rocchio

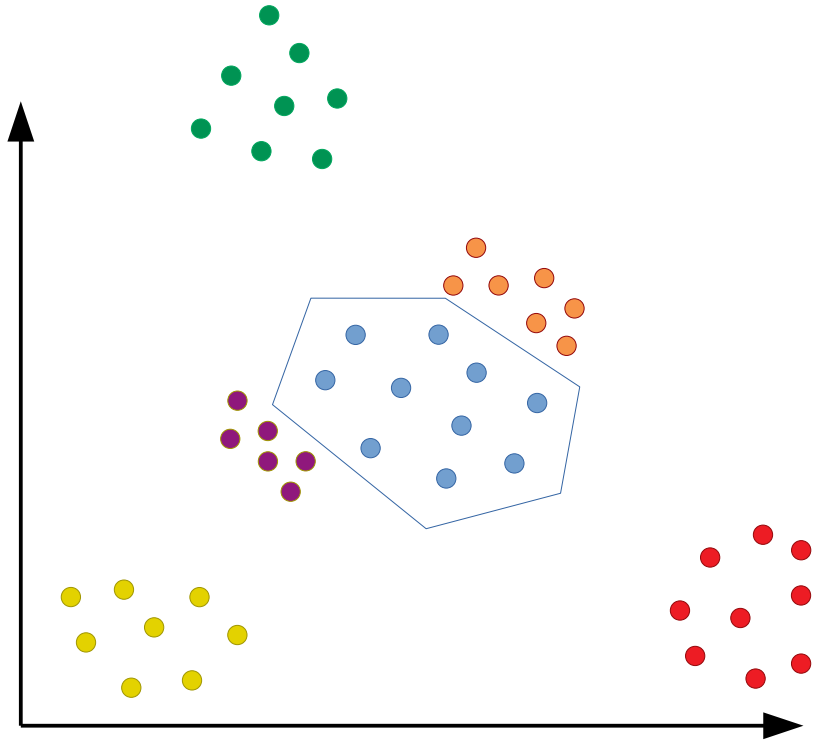
- Assim o centroide $\bar{c}_p$ da categoria *p* é calculado de modo a se aproximar da média dos vetores dos documentos da categoria *p*, e, simultaneamente, se afastar da média dos vetores fora da categoria *p*:

$$\bar{c}_p = \underbrace{\frac{\beta}{|C_p|} \sum_{d_j \in C_p} d_j}_{\text{Média dos vetores de docs da classe } p.} - \underbrace{\frac{\gamma}{N - |C_p|} \sum_{d_j \notin C_p} d_j}_{\text{Média dos vetores de docs fora da classe } p.}$$

Classificador de Rocchio

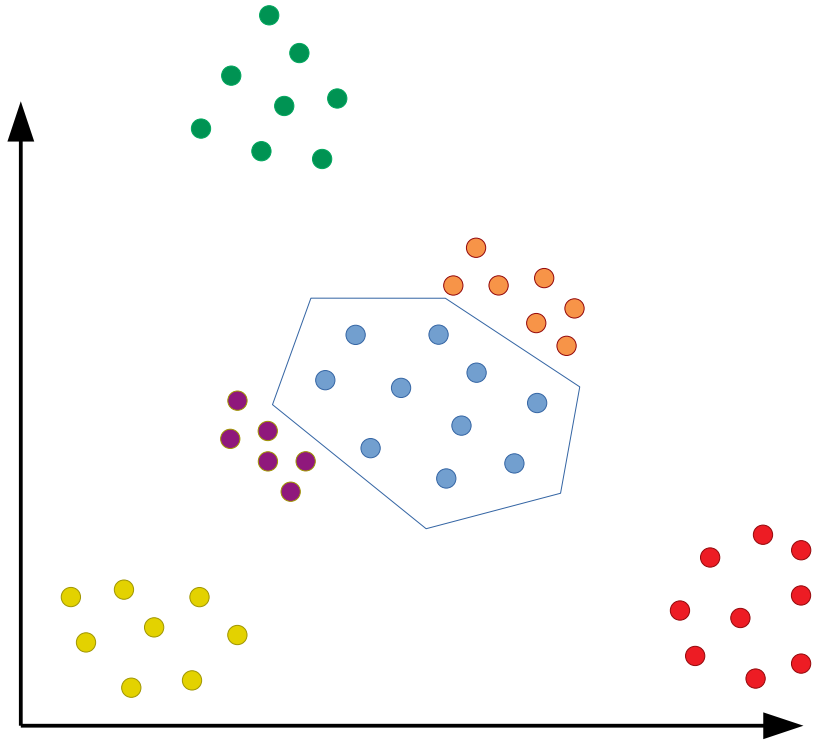
- Valores comumente utilizados para β e γ são 16 e 4. Alguns trabalhos utilizam ainda 1 e 0, respectivamente;
- Uma crítica ao classificador de Rocchio é que a consideração dos documentos fora da categoria p pode levar o centroide para longe dos documentos da categoria p;
- Uma forma de remediar essa inconveniência seria considerar apenas os documentos externos à categoria p que estejam próximos aos internos da categoria.

Classificador de Rocchio



- Note que, a tendência é que existam mais pontos fora da categoria do que dentro dela;
- Os pontos externos à categora podem assim apresentar alto grau de diversidade;
- Isso pode fazer com que centroide se afaste dos pontos da categoria;

Classificador de Rocchio



- Assim, para melhorar o desempenho do classificador, no lugar de considerar todos os pontos externos à categoria, considera-se apenas aqueles próximos aos pontos da categoria;
- No exemplo ao lado, no cálculo do centroide da classe azul, apenas os pontos externos roxos e laranjas seriam considerados.

Classificador de Rocchio

- Para classificar um novo documento *i* ainda não visto, pode-se atribuir um escore de classificação entre o doc e cada categoria *p* baseado na distância do vetor d_i para os centroides $\overline{c}_p$ das categorias;
- Assim, quanto mais próximo d_i for de $\overline{c}_p$, maior a probabilidade de se atribuir *i* à classe *p*.
- Versões apuradas do classificador de Rocchio podem apresentar desempenho competitivo aos métodos de ponta (como o *boosting*), com a vantagem de exigir menos esforço em seu treinamento

Classificadores de Bayes Ingênuos

- Para classificar um novo documento, estima-se a probabilidade do mesmo pertencer a cada uma das categorias;
- O classificador é dito ingênuo porque parte do pressuposto de que não há relação de dependência entre os termos que compõem os documentos (todavia, os modelos de RI clássicos partem dessa mesma premissa);
- Uma das variantes mais conhecidas é baseada no modelo probabilístico clássico.

Classificadores de Bayes Ingênuos

- Seja $S(d_j, c_p)$ o escore de pertencimento do documento j à classe p;
- De forma similar ao modelo probabilístico, temos:

$$S(d_j, c_p) = \frac{P(c_p|d_j)}{P(\bar{c}_p|d_j)}$$

Onde:

- $P(c_p|d_j)$ é a probabilidade de d_j pertencer à classe c_p;
- $P(\bar{c}_p|d_j)$ é a probabilidade de d_j não pertencer à classe c_p.

Classificadores de Bayes Ingênuos

- Aplicando a regra de Bayes, temos: $S(d_j, c_p) \sim \frac{P(d_j|c_p)}{P(d_j|\bar{c}_p)}$
- Onde:

$$P(d_j|c_p) = \prod_{k_i \in d_j} P(k_i|c_p) \times \prod_{k_i \notin d_j} P(\bar{k}_i|c_p)$$

$$P(d_j|\bar{c}_p) = \prod_{k_i \in d_j} P(k_i|\bar{c}_p) \times \prod_{k_i \notin d_j} P(\bar{k}_i|\bar{c}_p)$$

Classificadores de Bayes Ingênuos

- Aplicando a regra de Bayes, temos: $S(d_j, c_p) \sim \frac{P(d_j|c_p)}{P(d_j|\bar{c}_p)}$
- Onde:

Prob de k_i estar em um doc de c_p

Prob de k_i não estar em um doc de c_p

$$P(d_j|c_p) = \prod_{k_i \in d_j} P(k_i|c_p) \times \prod_{k_i \notin d_j} P(\bar{k}_i|c_p)$$

Conjunto de termos k_i presentes em d_j

Conjunto de termos k_i não presentes em d_j

$$P(d_j|\bar{c}_p) = \prod_{k_i \in d_j} P(k_i|\bar{c}_p) \times \prod_{k_i \notin d_j} P(\bar{k}_i|\bar{c}_p)$$

Classificadores de Bayes Ingênuos

- Aplicando a regra de Bayes, temos: $S(d_j, c_p) \sim \frac{P(d_j|c_p)}{P(d_j|\bar{c}_p)}$
- Onde:

$$P(d_j|c_p) = \prod_{k_i \in d_j} P(k_i|c_p) \times \prod_{k_i \notin d_j} P(\bar{k}_i|c_p)$$

Prob de k_i estar em um doc fora de c_p

Prob de k_i não estar em um doc fora de c_p

$$P(d_j|\bar{c}_p) = \prod_{k_i \in d_j} P(k_i|\bar{c}_p) \times \prod_{k_i \notin d_j} P(\bar{k}_i|\bar{c}_p)$$

Classificadores de Bayes Ingênuos

- Aplicando o mesmo raciocínio usado para deduzir a fórmula de similaridade do modelo probabilístico:

$$S(d_j, c_p) \sim \sum_{k_i \in d_j} \left(log \left(\frac{P(k_i|c_p)}{1-P(k_i|c_p)} \right) + log \left(\frac{1-P(k_i|\bar{c}_p)}{P(k_i|\bar{c}_p)} \right) \right)$$

- As probabilidades $P(k_i|c_p)$ e $P(k_i|\bar{c}_p)$ podem ser obtidas a partir do conjunto de documentos de treinamento:

Classificadores de Bayes Ingênuos

- As probabilidades $P(k_i|c_p)$ e $P(k_i|\bar{c}_p)$ podem ser obtidas a partir do conjunto de documentos de treinamento:

$$P(k_i|c_p) = \frac{1+n_{i,p}}{2+n_p} \qquad P(k_i|\bar{c}_p) = \frac{1+n_i-n_{i,p}}{2+N_t-n_p}$$

Onde:

- n_i: nº de docs com o termo k_i;
- n_p: nº de docs na classe c_p;
- $n_{i,p}$: nº de docs com o termo k_i na classe c_p;
- N_t: nº de docs de treinamento.

Classificadores de Bayes Ingênuos

- Ao receber um novo documento d_j, o algoritmo computará os escores $S(d_j, c_p)$ para cada classe c_p usando as fórmulas anteriores;
- O documento então será classificado nas classes com maiores escores.

Classificadores Ensemble

- Classificadores do tipo ensemble combinam previsões de classificadores distintos para gerar um novo escore preditivo;
- A ideia é que a combinação de classificadores se saia melhor do que seu uso individual (um classificador que falhe em uma situação poderia ser “coberto” pelos demais que acertaram);
- Assim, os classificadores ensemble podem utilizar os demais algoritmos de classificação, incluindo outros classificadores ensemble;
- Diversas estratégias podem vir a ser utilizadas para decidir a classificação de um texto, desde a votação por maioria até uma soma ponderada baseada no grau de acerto de cada classificador.

Classificadores Ensemble

- Após treinar os classificadores individuais usados na combinação, pode-se aplicá-los às próprias instâncias de treinamento e verificar o nível de acerto;

- Assim, pode-se prever o classificador individual que melhor classifica cada categoria ou calcular pesos para a construção de um classificador combinado que minimize respostas erradas.

Classificadores Ensemble

- Um tipo especial de classificador ensemble é o *boosting*, que treina K instâncias do mesmo classificador com as instâncias de treinamento;
- Assim, *boosting* executa K iterações. Em cada uma delas, uma instância do classificador é treinada usando as instâncias de treinamento.
- Para evitar que todas as instâncias do classificador fiquem idênticas, é associado um peso a cada um dos docs da base de treinamento. Esses pesos podem então variar de uma iteração para outra;
- Dessa forma, docs com pesos maiores tem maior influência no ajuste do classificador em comparação com os docs com pesos menores;

Classificadores Ensemble

- Então, a cada iteração, o *boosting* atribuiu peso maior aos docs da base treinamento incorretamente classificados na iteração anterior;
- Assim, a tendência é que as diversas instâncias do classificador treinadas acabem tendo ajustes diferentes;
- Cada uma das instâncias do classificador treinadas recebe um peso.
- Após o treinamento de todas as instâncias, o classificador *boosting* está pronto: quando um novo documento chega, todas as instâncias do classificador são usadas para classificá-lo. A resposta final é obtida considerando-se a resposta de cada instância do classificador com sua respectiva ponderação;

Métricas de Avaliação de Classificadores

1) Para avaliar a qualidade dos classificadores após o treinamento, costumam-se utilizar três conjuntos de métricas de avaliação:

Métricas de Avaliação de Classificadores

- Para avaliar a qualidade dos classificadores após o treinamento, costumam-se utilizar três conjuntos de métricas de avaliação:

1) **Acurácia e Erro**: para cada classe, mede-se o percentual de acerto (acurácia) e o percentual de falha (erro) na classificação de documentos (com classe pré conhecida);

Métricas de Avaliação de Classificadores

- Para avaliar a qualidade dos classificadores após o treinamento, costumam-se utilizar três conjuntos de métricas de avaliação:

1) **Acurácia e Erro**: para cada classe, mede-se o percentual de acerto (acurácia) e o percentual de falha (erro) na classificação de documentos (com classe pré conhecida);

2) **Precisão e Revocação**: para cada classe, calcula-se o percentual de acerto dos resultados atribuídos a classe (precisão) e o percentual de cobertura dos documentos da classe (revocação);

Métricas de Avaliação de Classificadores

- Para avaliar a qualidade dos classificadores após o treinamento, costumam-se utilizar três conjuntos de métricas de avaliação:

1) **Acurácia e Erro**: para cada classe, mede-se o percentual de acerto (acurácia) e o percentual de falha (erro) na classificação de documentos (com classe pré conhecida);

2) **Precisão e Revocação**: para cada classe, calcula-se o percentual de acerto dos resultados atribuídos à classe (precisão) e o percentual de cobertura dos documentos da classe (revocação);

3) **Medida-F1**: para cada classe, calcula a média harmônica entre precisão e revocação da classe.

Coleções padrão

- Existem coleções de referência para testes com algoritmos de classificação de documentos. Algumas das mais conhecidas são:
- **Reuters-21578**: artigos de notícias da agência Reuters de 1987;
- **Reuters Corpus Volumes**: conjunto mais amplo com mais de 800.000 notícias;
- **OHSUMED**: coleção de mais 300.000 referências médicas, organizado pela MEDLINE;
- **20 NewsGroups**: coleção com mais de 20.000 mensagens postadas em grupos de notícias da Usenet.

Coleções padrão

- Existem coleções de referência para testes com algoritmos de classificação de documentos. Algumas das mais conhecidas são:
- **WebKB**;
- **ACM-DL;**
- **Wikipedia;**
- ...