Aprendizagem de Máquina

Alessandro L. Koerich/Alceu S. Britto

Programa de Pós-Graduação em Informática
Pontifícia Universidade Católica do Paraná (PUCPR)

Combinação de Classificadores

Filosofia

- O objetivo da combinação de classificadores é:

Buscar uma classificação mais precisa pagando o preço de uma maior complexidade.

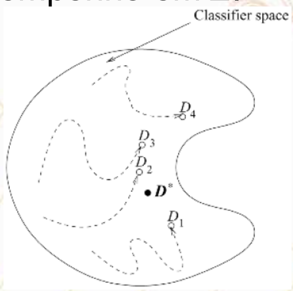
- Ao invés de buscar o melhor conjunto de características e o melhor classificador, buscamos agora **o melhor conjunto de classificadores** e então **o melhor método de combinação**.

Filosofia

- Existem três razões pelas quais uma combinação de classificadores deve ser melhor do que um único classificador:
  - Estatístico
  - Computacional
  - Representacional

Estatística

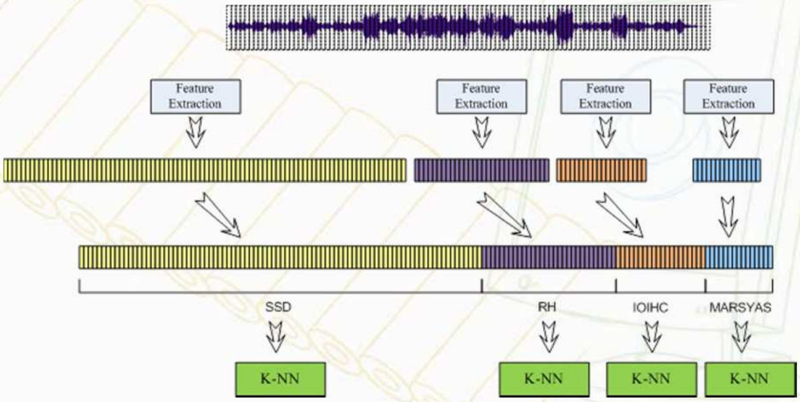
- Suponha que temos um conjunto de dados *Z* rotulado e diversos classificadores (D_1, D_2,...) com um bom desempenho em *Z*.



- Podemos pegar um único classificador como solução, correndo o risco de tomar uma má decisão para o problema.

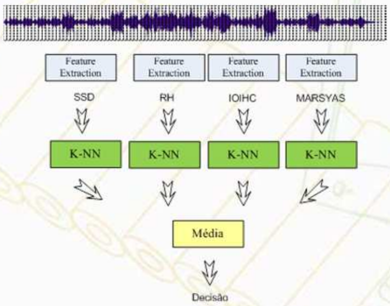
Estatística

- Exemplo: suponha que executamos um classificador 1-NN para 4 subconjuntos diferentes de características, obtendo como consequência 4 classificadores com erro zero no conjunto de treinamento.



Estatística

- Apesar destes classificadores terem todos o mesmo desempenho sobre o conjunto de treinamento
  - eles podem ter diferentes desempenhos na generalização!!!
- Ao invés de pegar somente um classificador, uma opção mais segura seria usá-los todos e fazer a média de suas saída.

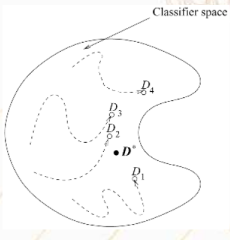


Estatística

- Mas este conjunto de classificadores pode não ser melhor do que o melhor classificador individual!
- Mas diminuirá ou eliminará o risco de escolhermos um classificador individual inadequado!

Computacional

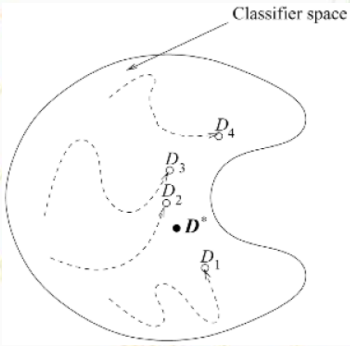
- Alguns algoritmos de treinamento executam *hill-climbing* ou alguma busca aleatória, que podem levar a diferentes ótimos locais.



- Assumimos que o processo de treinamento de cada classificador individual inicia em algum lugar no espaço de classificadores possíveis e termina próximo do classificador ótimo *D*.

Computacional

- Alguma forma de combinação pode levar a um classificador que é uma melhor aproximação para *D* do que qualquer outro D_i individual.

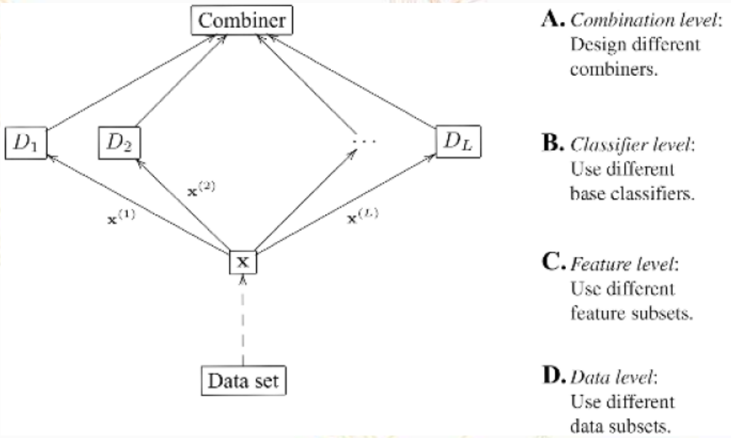


Representacional

- É possível que o espaço do classificador considerado para o problema não contenha o classificador ótimo.
- Por exemplo, classificador ótimo para dados não-lineares será também, não-linear.
- Se restringirmos o espaço dos classificadores possíveis a somente classificadores lineares, o classificador ótimo não pertencerá a este espaço.
- Entretanto, um conjunto (*ensemble*) de classificadores lineares pode aproximar qualquer fronteira de decisão com alguma precisão.

Terminologia e Taxonomia

- Quatro maneiras de construir combinação de classificadores



Fusão e Seleção

- Existem duas estratégias principais para combinar classificadores:
  - Fusão
  - Seleção

- Na fusão de classificadores, supõe-se que cada membro do *ensemble* possui conhecimento do espaço de características completo.

- Na seleção de classificadores, supõe-se que cada membro do ensemble conhece bem uma parte do espaço de características e é responsável por objetos nesta parte.

Otimização da Decisão

- Otimização da decisão refere-se a métodos para escolher e otimizar a combinação para um conjunto fixo de classificadores de base.
- Otimização de cobertura refere-se a métodos para criação de uma diversidade de classificadores de base assumindo uma combinação fixa.

Treinamento ou não?

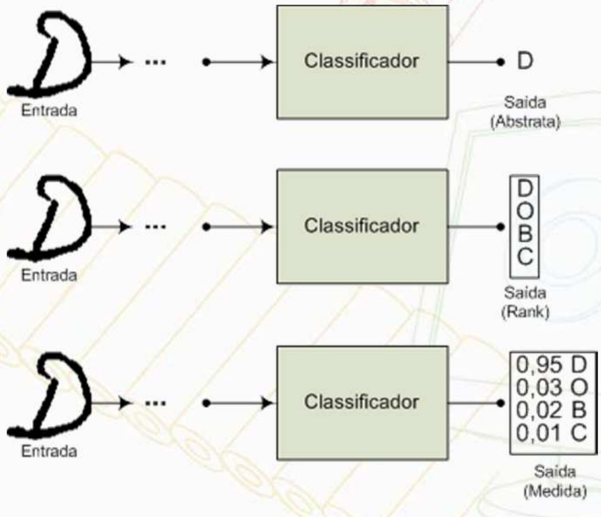
- Algumas combinações não necessitam treinamento após os classificadores do ensemble serem treinados individualmente. Um exemplo é a combinação “votação da maioria”.
- Outra combinações necessitam de treinamento adicional, por exemplo, a combinação média ponderada
- Uma terceira classe de ensembles desenvolve a combinação durante o treinamento dos classificadores individuais → AdaBoost.

Tipos de Saídas de Classificadores

- As maneiras possíveis de combinar as saídas de *L* classificadores em um *ensemble*, depende da informação obtida dos membros individuais.

- Podemos distinguir três (3) tipos de saídas de classificadores:
  - Tipo 1: Nível Abstrato: Cada classificador produz um rótulo
  - Tipo 2: Nível de Rank: Cada classificador produz uma lista ordenada de alternativas de acordo com a plausibilidade.
  - Tipo 3: Nível de Medidas: Cada classificador produz um vetor de c dimensões onde cada valor representa o suporte para o vetor.

Tipos de Saídas de Classificadores

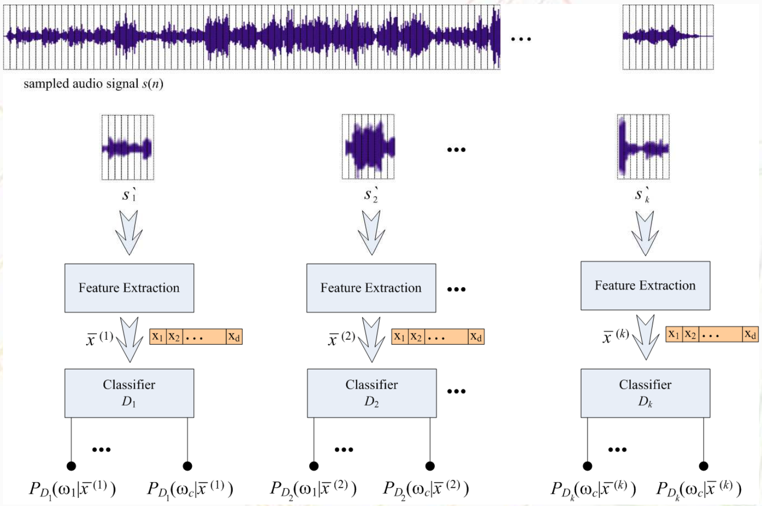


Voto da Maioria (Majority Vote)

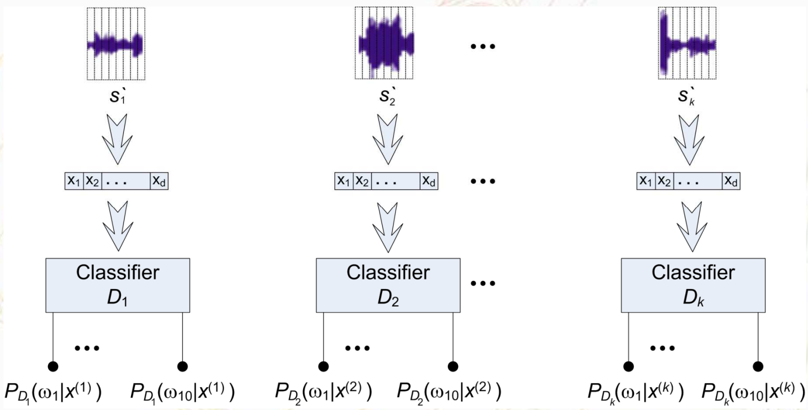
Estratégias de Combinação

- Assumimos um vetor *d*-dimensional de características, *x.*
- Assumimos que existem *c* classes possíveis rotuladas w_1 a w_c organizado como um conjunto de rótulos.
- Assumimos um conjunto de *k* classificadores D_1... D_k onde cada classificador D_i produz na saída [P_{Di} (w_1|x), P_{Di} (w_2|x), ..., P_{Di} (w_j|x)], onde P_{Di} (w_j|x) representa o suporte para que hipótese de que o vetor x seja da classe w_j.

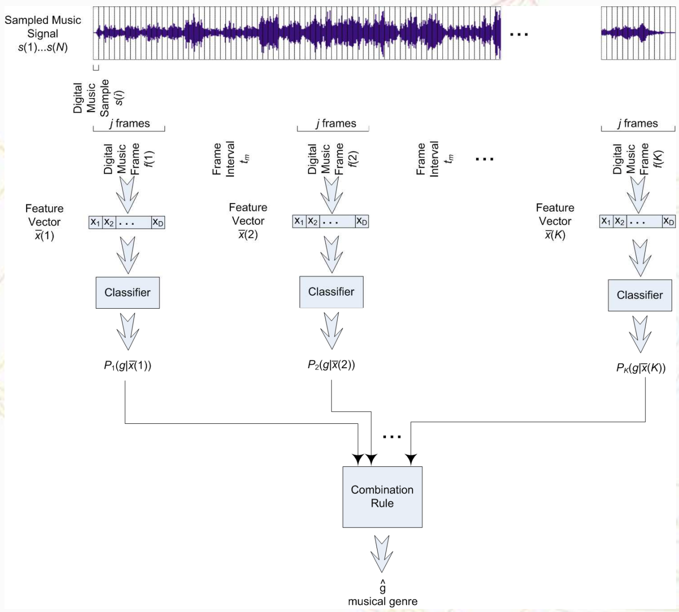
Estratégias de Combinação



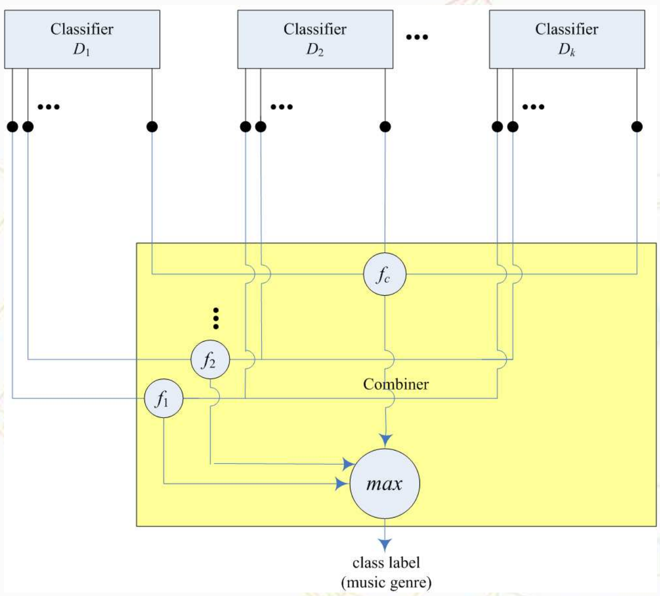
Estratégias de Combinação



Estratégias de Combinação



Detalhes da Combinação



Votação da Maioria

- Regra de decisão simples, onde somente os rótulos atribuídos pelo classificador são levados em conta e aquele que tiver mais votos é o vencedor.

$$\hat{\omega} = \max_{i \in [1,k]} count\left[\arg\max_{\omega \in \Omega} P_{D_i}(\omega \mid x)\right]$$

- onde *i* indica o índice do classificador e P_{Di} indica o nível de confiança fornecido na saída do classificador D_i

Máx

- Regra de decisão simples, onde a classe com o nível de confiança mais elevado é a vencedora, pouco importando o classificador.

$$\hat{\omega} = \arg\max_{\substack{i \in [1,k] \\ \omega \in \Omega}} P_{D_i}(\omega \mid x)$$

Soma

- A regra da soma é baseada no somatório dos níveis de confiança fornecidos pelos classificadores.
- Os níveis de confiança são somados para cada classe e a classe cuja soma resultante for a mais elevada, é declarada vencedora.

$$\hat{\omega} = \arg\max_{\omega \in \Omega} \sum_{i=1}^{k} P_{D_i}(\omega \mid x)$$

Soma Ponderada

- Ao invés de utilizar somente a regra da soma, é possível adicionar pesos às saídas dos classificadores, dando mais podes aos classificadores mais competentes.
- Neste caso, o nível de confiança de cada classificador é multiplicado por *k* pesos w_i, e os pesos são específicos para cada classificador.

$$\hat{\omega} = \arg\max_{\omega \in \Omega} \sum_{i=1}^{k} w_i P_{D_i}(\omega \mid x)$$

Produto

- A regra do produto é baseada na multiplicação dos níveis de confiança fornecidos pelos classificadores.
- Os níveis de confiança são multiplicados, para cada classe, e a classe cujo produto resultante for o mais elevado, é declarada vencedora.

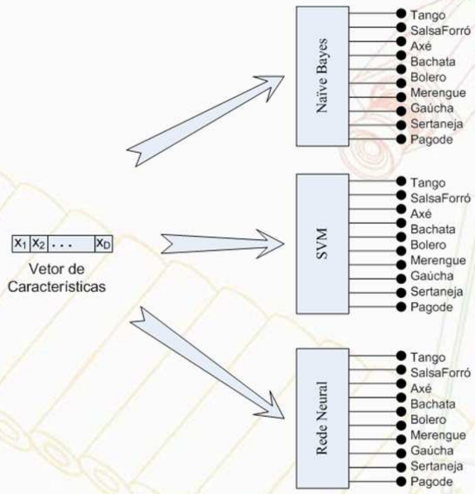
$$\hat{\omega} = \arg\max_{\omega \in \Omega} \prod_{i=1}^{k} P_{D_i}(\omega \mid x)$$

Produto Ponderado

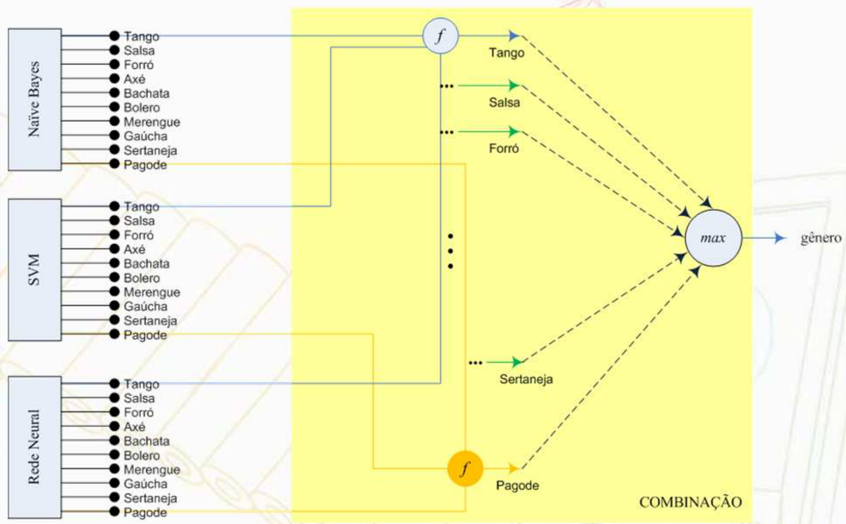
- Segue a mesma ideia da soma ponderada.
- Neste caso, o nível de confiança de cada classificador é elevado à *k* pesos w_i, e os pesos são específicos para cada classificador.

$$\hat{\omega} = \arg\max_{\omega \in \Omega} \prod_{i=1}^{k} \left[P_{D_i}(\omega \mid x) \right]^{w_i}$$

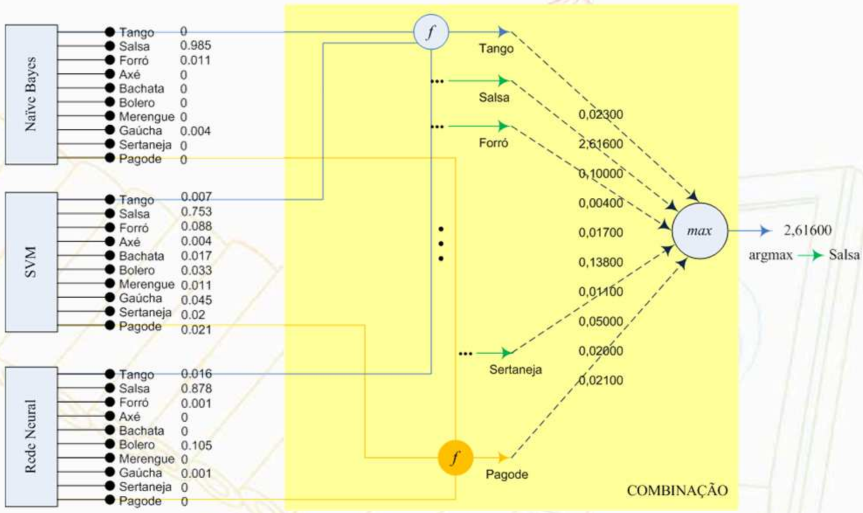
Exemplo



Exemplo



Exemplo



Naïve Bayes
Tango 0
Salsa 0.985
Forró 0.011
Axé 0
Bachata 0
Bolero 0
Merengue 0
Gaúcha 0.004
Sertaneja 0
Pagode 0
SVM
Tango 0.007
Salsa 0.753
Forró 0.088
Axé 0.004
Bachata 0.017
Bolero 0.033
Merengue 0.011
Gaúcha 0.045
Sertaneja 0.02
Pagode 0.021
Rede Neural
Tango 0.016
Salsa 0.878
Forró 0.001
Axé 0
Bachata 0
Bolero 0.105
Merengue 0
Gaúcha 0.001
Sertaneja 0
Pagode 0
f
Tango
Salsa
Forró
Sertaneja
Pagode
0,02300
2,61600
0,10000
0,00400
0,01700
0,13800
0,01100
0,05000
0,02000
0,02100
max
2,61600
argmax
Salsa
COMBINAÇÃO