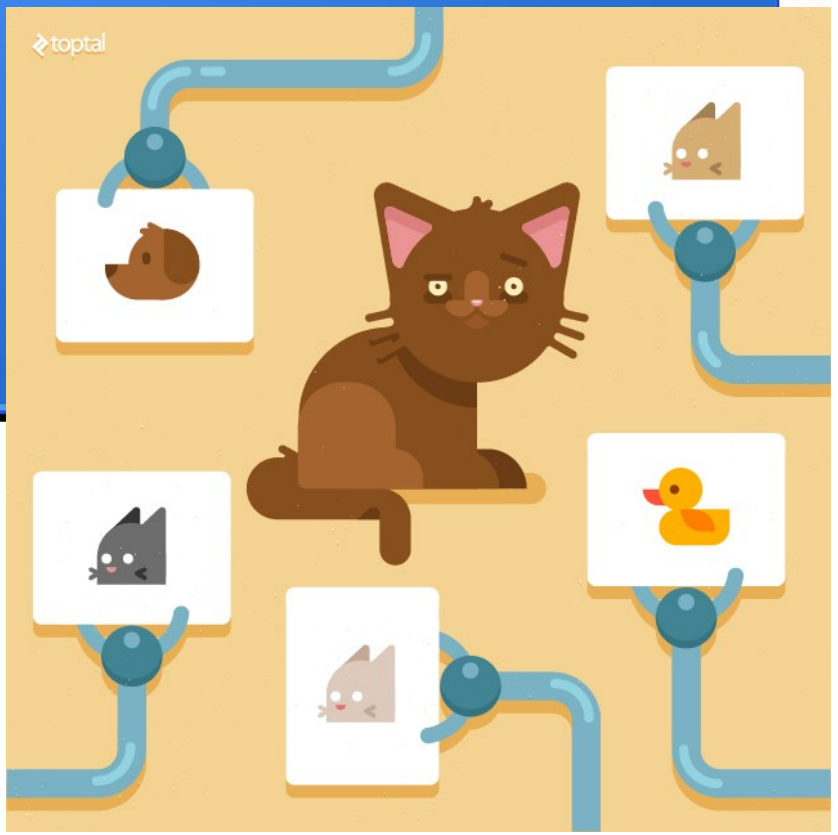
ICC204 - Aprendizagem de Máquina e Mineração de Dados

Ensembles



Prof. Rafael Giusti
rgiusti@icomp.ufam.edu.br

Objetivos da aula

- Compreender o que são *ensembles* e por que eles provavelmente funcionam para um problema de classificação
- Compreender as condições para que um classificador-base seja adequado em um *ensemble*
- Aprender os métodos utilizados para introduzir diversidade em um *ensemble*
- Explorar algoritmos conhecidos que introduzem diversidade em *ensembles*

Leitura recomendada

- Sobre o aprendizado de *ensembles* e *boosting*
  - Rusell e Norvig. "Inteligência Artificial: uma Abordagem Moderna". 2ª Edição. Seção 18.4.
  - Bishop. "*Pattern Recognition and Machine Learning*". Capítulo 14, em particular as seções 14.2 e 14.3.

Motivação

- **Realidade**: muitos problemas práticos podem ser resolvidos com o uso de técnicas de aprendizagem de máquina:
  - Redes Neurais, SVM, Árvore de Decisão...
- **Objetivos da solução**: desenvolver um método...
  - Robusto e bem adaptado ao problema a ser solucionado.
  - Com alta taxa de reconhecimento e com pouco custo computacional

Motivação

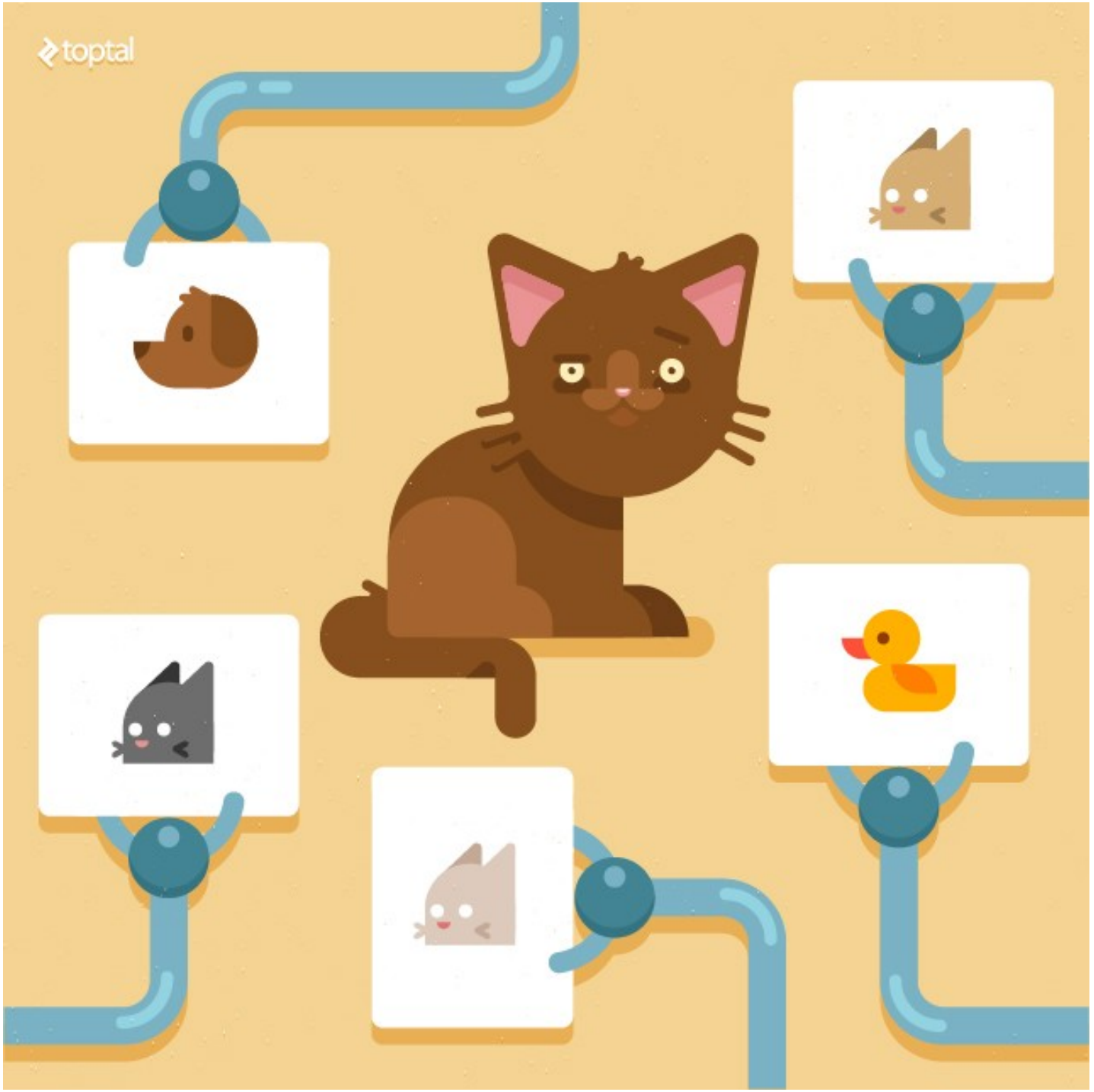
- **Problemas com a solução**:
  - A criação de um método robusto e bem adaptado ao problema a ser solucionado é uma tarefa complexa
    - Tempo de treinamento longo
    - Necessidade de muitos dados iniciais para o treinamento
  - Teorema No Free Lunch

Motivação

- **Problemas com a solução**:
  - Teorema No Free Lunch
    - Nenhum algoritmo pode ser considerado melhor do que os outros se a superioridade não for demonstrada sobre todas as possíveis classes de problemas

Motivação

- **Ensembles**, **comitês** ou **conjuntos** de classificadores
  - Combinação de diferentes modelos para uma mesma tarefa de classificação
  - Substituem um modelo extremamente robusto e preciso por modelos simples e aproximadamente corretos
    - Os modelos que compõem o *ensemble* são chamados classificadores-base



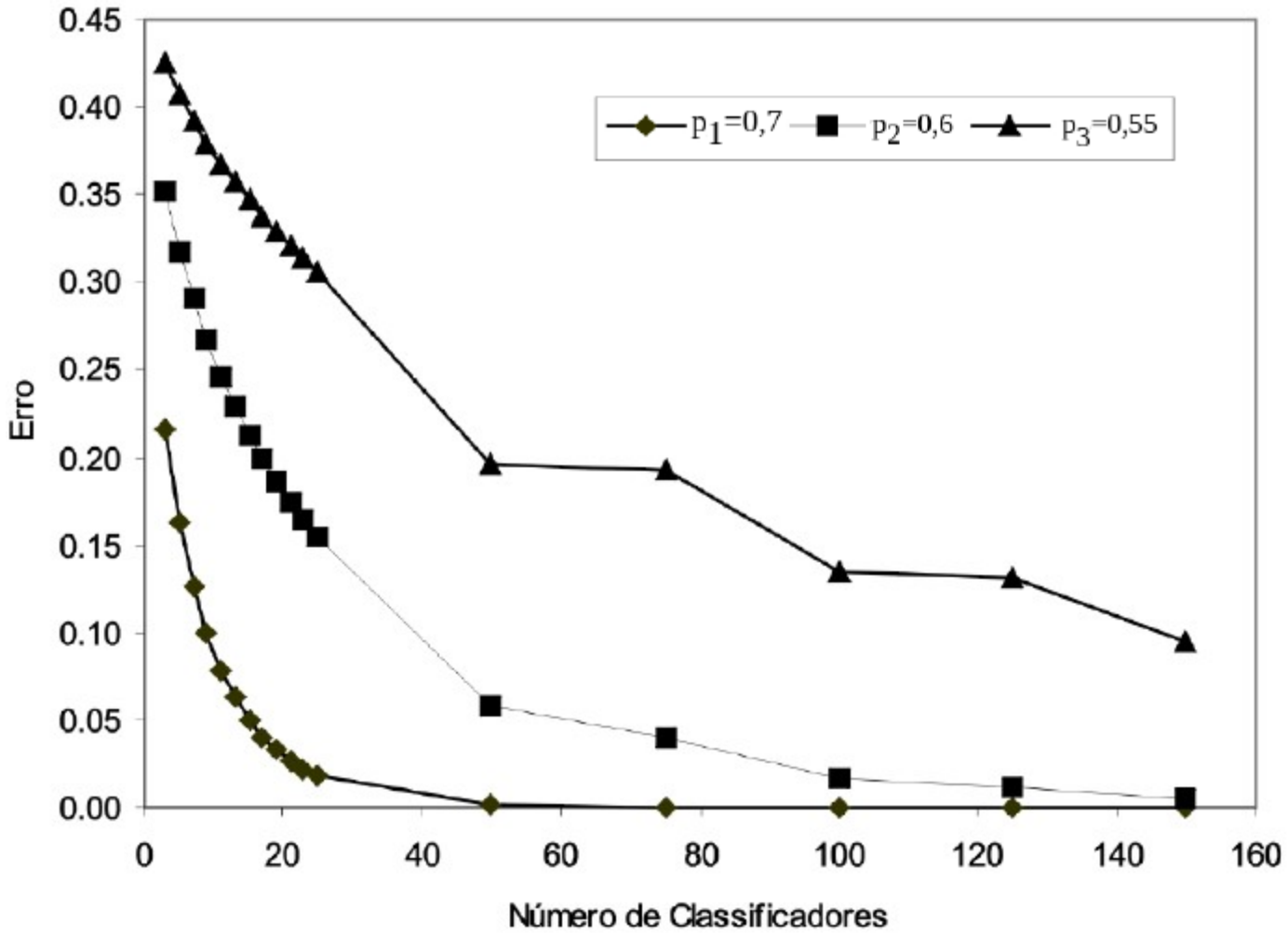
Fonte: https://www.toptal.com/machine-learning/ensemble-methods-machine-learning

Motivação

- Intuitivamente, a motivação para *ensembles* parece trivial
  - Considere um conjunto com 5 classificadores--base com voto majoritário simples
    - Isto é, cada classificador vota em uma classe e o *ensemble* responde com o voto mais frequente
    - Esse *ensemble* só irá classificar incorretamente um exemplo se ao menos 3 classificadores-base cometerem erros

Motivação

Probabilidade de um ensemble cometer erro em função da probabilidade de erro dos classificadores-base e do número de classificadores-base.



Motivação

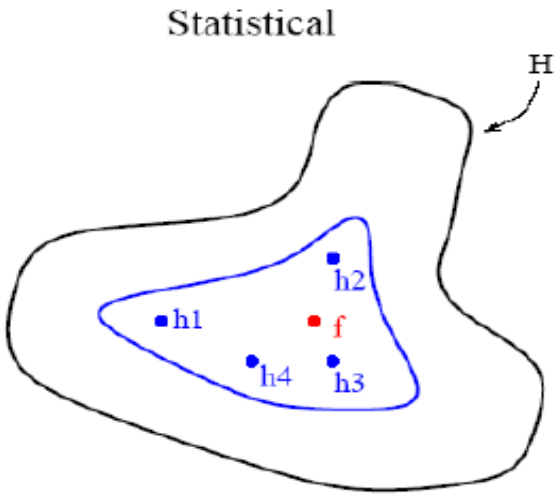
- Essa análise é bastante simplificada, pois assume que os erros dos classificadores-base são **independentes**
  - Porém, se houver uma **expectativa razoável** de que a correlação entre os erros seja baixa, nós nos **aproximamos** dessa situação ideal
- Mais formalmente, existem justificativas, baseadas no conceito de aprendizado como busca, para o sucesso de um *ensemble*

Ensembles: por quê?

- Existem três justificativas para o uso de um *ensemble*
  - **Estatística**: combinação de soluções que parecem ser igualmente boas
  - **Computacional**: combinação de diferentes mínimos locais
  - **Representational**: combinação de aproximações que, aparentemente, não são individualemente boas

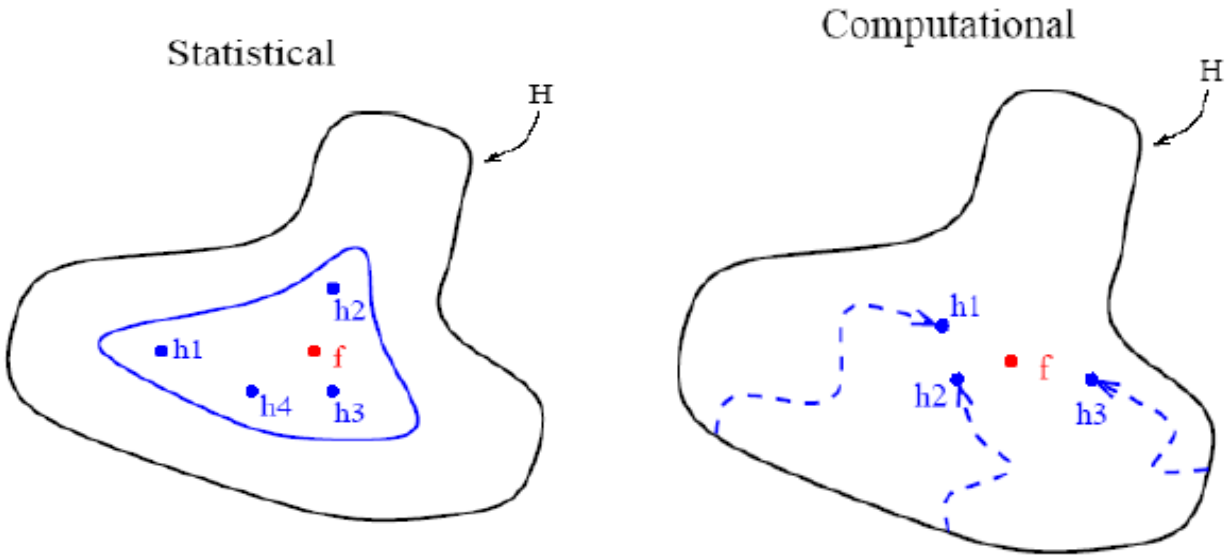
Ensembles: por quê?

- Existem três justificativas para o uso de um *ensemble*
  - **Estatística**: combinação de soluções que parecem ser igualmente boas



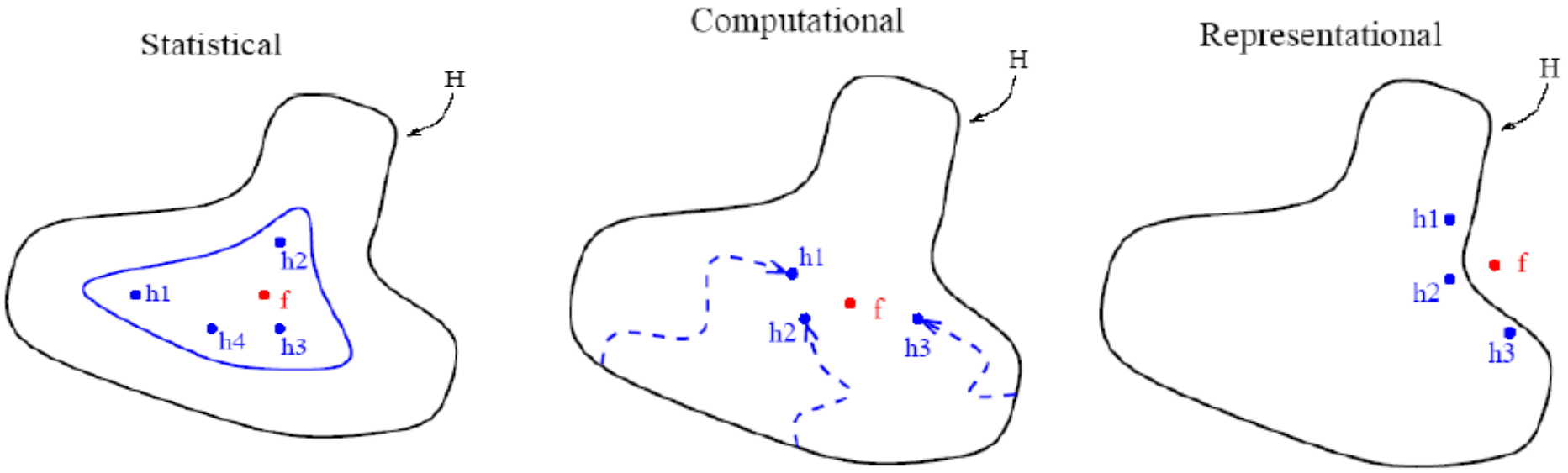
Ensembles: por quê?

- Existem três justificativas para o uso de um *ensemble*
  - **Computacional**: combinação de diferentes mínimos locais



Ensembles: por quê?

- Existem três justificativas para o uso de um *ensemble*
  - **Representational**: combinação de aproximações que, aparentemente, não são individualemente boas



Ensembles: quando

- Para que possamos nos aproximar da situação ideal de *ensembles*, é necessário que
  - A probabilidade de erro de cada classificador-base seja razoavelmente baixa
  - A correlação entre os erros dos classificadores--base também deve ser razoavelmente baixa
- Em outras palavras, classificadores-base devem ser corretos e diversos

Ensembles: quando

- Os *ensembles* devem ser coleções de classificadores que sejam
  - **Diversos**
    - Se todos os classificadores-base aproximarem a função-conceito da mesma forma, então o *ensemble* terá o mesmo desempenho que qualquer um dos classificadores-base

Ensembles: quando

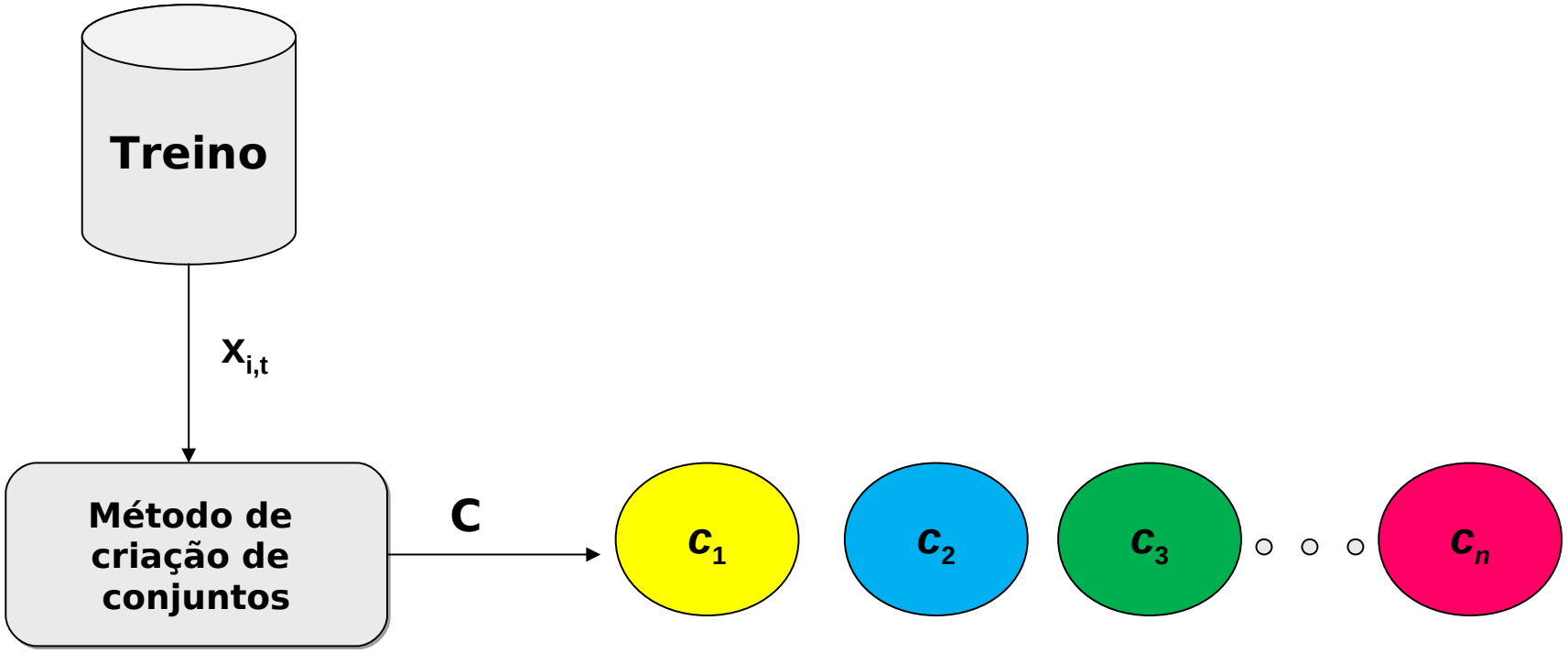
- Os *ensembles* devem ser coleções de classificadores que sejam
  - **Corretos**
    - Se todos os classificadores-base tiverem um erro muito elevado, então o *ensemble* também irá produzir uma classificação incorreta

Ensembles: quando

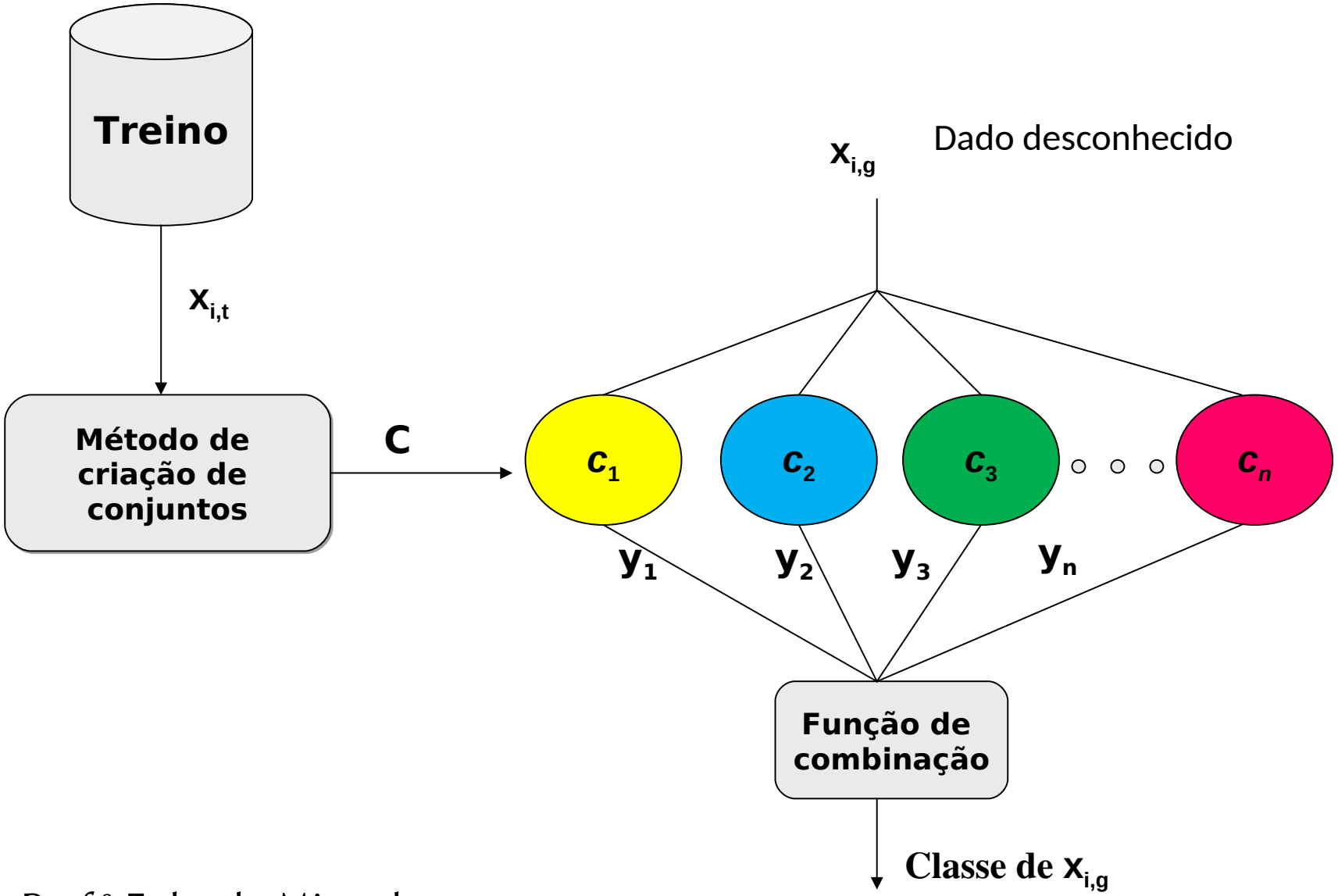
- O grande desafio da construção de *ensembles* é manter um equilíbrio entre acurácia e diversidade
- Métodos de criação de *ensembles* frequentemente incorporam algum mecanismo de manutenção da diversidade

Métodos de Criação

Métodos de Criação



Métodos de Criação



Métodos de Criação

Inputs:	$D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$	*%Training data set*
	$L_1, \ldots, L_T$	*%Learning methods*
Learning Process:	for i=1...T	
	$\{C_i = L_i(D)$	*%Train one base classifier*
	calculate $\alpha_i\}$	*%Calculate weight of the base classifier*
	end if	
Output:	$C(x) = \sum_{i=1}^{T} \alpha_i C_i(x)$	*%Final classifier*

An example of the representation of a simple ensemble method

Fonte: ANNA JUREK , YAXIN BI , SHENGLI WU and CHRIS NUGENT. A survey of commonly used ensemble-based classification techniques. The Knowledge Engineering Review, Vol. 29:5, 551–581, 2013.

Métodos de criação

- Devemos manipular os dados ou os modelos de forma a **introduzir diversidade** no *ensemble*
- Podemos manipular...
  - **Os dados de entrada**, utilizando reamostragem dos exemplos de treinamento
  - **O espaço de características**, utilizando reamostragem do espaço de atributos
  - **Os membros do conjunto**, utilizando modelos heterogêneos

Métodos de criação

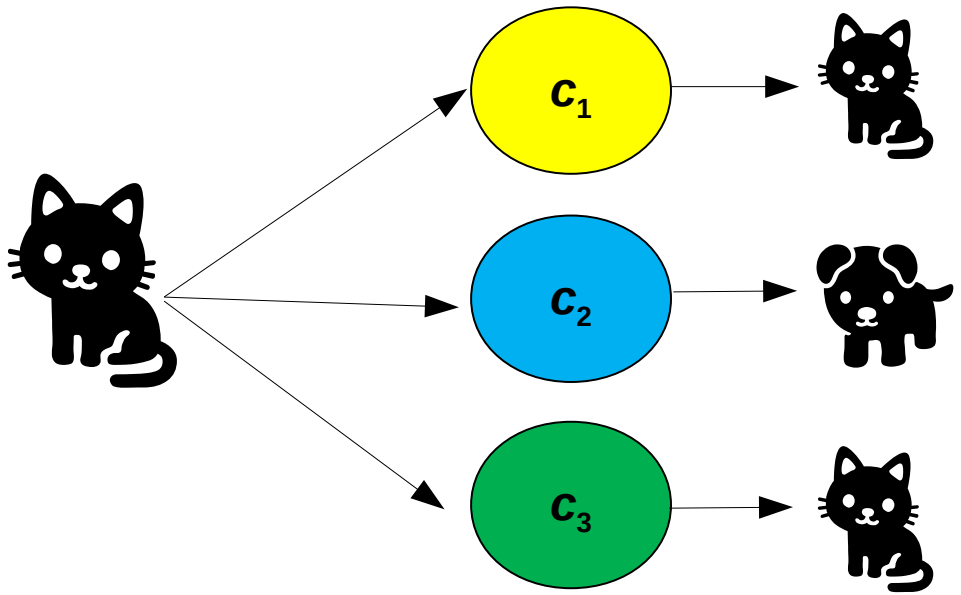
- O *ensemble* necessita também de uma **estratégia de combinação** das classificações individuais
  - Para problemas em que os exemplos possuem um único rótulo, normalmente recorre-se a **estratégias de voto**
  - Cada classificador vota em uma classe
  - Os votos são combinados
    - Exemplos: moda e *argmax* de pesos sobre os votos

Métodos de criação

- O *ensemble* também pode ter como base modelos de **regressão**
  - Nesse caso a estratégia de combinação deve ser uma função que obtenha uma variável-alvo numérica
    - Exemplo: média simples

Funções de combinação

- As rotulações individuais dos classificadores-base devem ser combinadas em uma classificação do *ensemble*
  - Uma simples estratégia é o **voto majoritário**



2 votos para "gato"
1 voto para "cachorro"

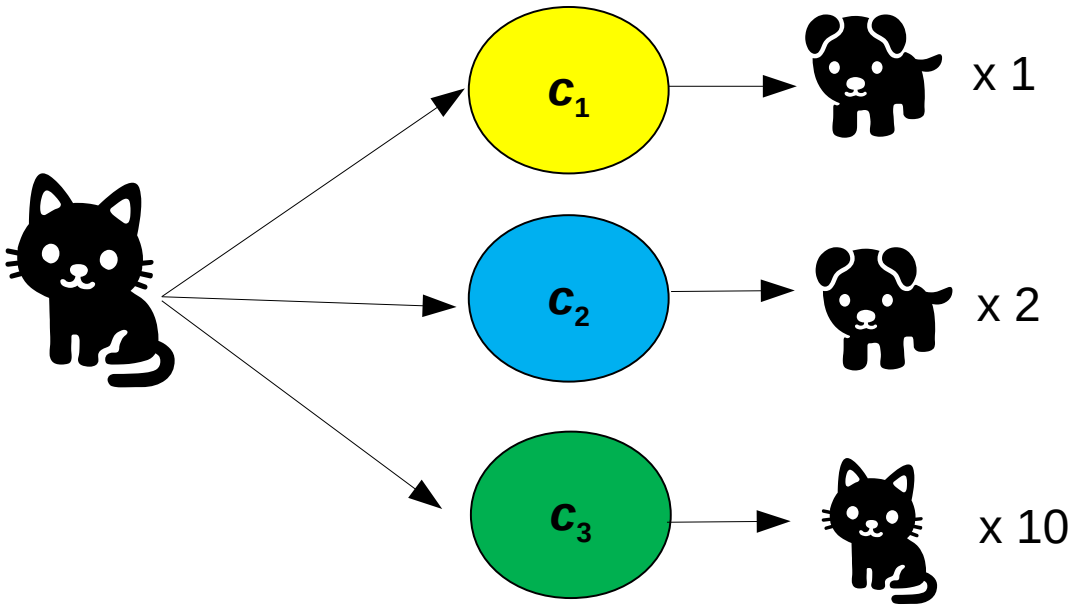
Resultado: "gato"

Funções de combinação

- Alternativamente, pode-se empregar um **voto ponderado**
  - Nem todos os classificadores são igualmente bons em resolver um problema
  - Os classificadores com maior capacidade de generalização podem ter maior peso na determinação do *ensemble*

Funções de combinação

- No voto ponderado, cada classificador atribui um **peso** ao seu voto



"Cachorro" é votado com peso 3

"Gato" com peso 10

Resultado: "gato"

Funções de combinação

- Pesos estáticos
  - Ao construir os modelos, utiliza-se reamostragem para estimar o desempenho do classificador
  - A acurácia de cada classificador é o peso do seu voto
  - Classificadores que generalizaram melhor ao reamostrarmos os exemplos de treinamento têm mais peso

Funções de combinação

- Pesos dinâmicos
  - Ao classificar um novo exemplo, cada classificador calcula um peso dinamicamente com base na sua "confiança" da rotulação
    - Por exemplo, em um modelo de vizinhos mais próximos, o inverso da distância entre o exemplo e os protótipos pode ser um peso
    - Podemos também estimar a probabilidade de cada classe para o Naive Bayes

Introdução de diversidade

- Podemos introduzir diversidade no *ensemble* manipulando os dados, as características ou os classificadores-base
- Algumas estratégias incluem
  - Usar modelos distintos para introduzir diversidade nos membros do conjunto
  - Reamostrar os exemplos (e.g., *bagging*) para obter diferentes "visões" do espaço de decisão

Introdução de diversidade

- Algumas estratégias incluem
  - Ponderar os exemplos (e.g., *boosting*) para aprender modelos cada vez mais aptos a corrigir os erros detectados na fase de aprendizado
  - Combinar classificadores em sequência (e.g., *stacking*) para obter meta-classificadores que aprendem a ponderar o viés de cada modelo
  - Amostrar o espaço de características (e.g., florestas aleatórias) para obter modelos focados em diferentes dimensões dos dados

Ensembles heterogêneos

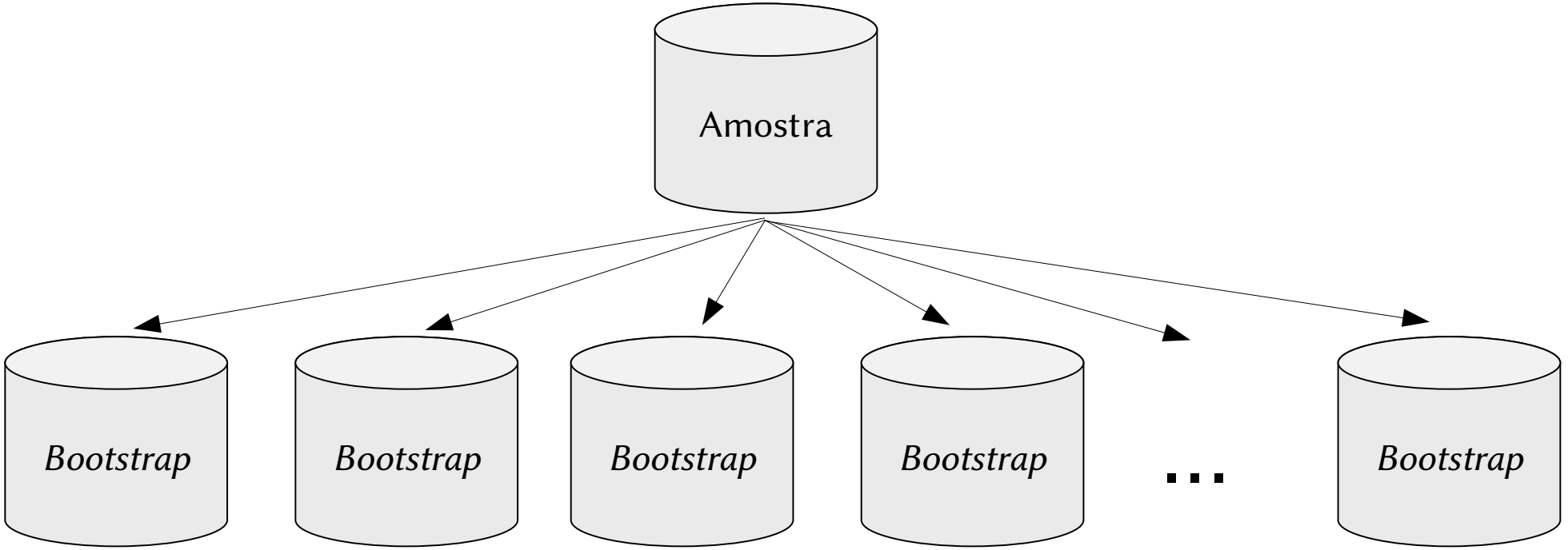
- Introduz diversidade ao *ensemble* empregando manipulando os **membros do conjunto**
- Espera-se que diferentes modelos tenham baixa correlação entre seus erros
  - Por exemplo, combinar um modelo de separação linear com um modelo probabilístico
  - Combinar modelos com diferentes capacidades para obter diferentes graus de generalização do mesmo problema

Ensembles heterogêneos

- Os classificadores-base do *ensemble* podem ser
  - Modelos diferentes
    - Por exemplo, um *ensemble* de árvores de decisão, SVM e *k*-NN
  - Induzidos com hiperparâmetros distintos
    - Exemplo: todos os classificadores-base são modelos SVM, mas cada um é induzido com um parâmetro distinto de C

Bagging

- Manipulação dos dados de entrada
  - Baseado em *boostrap*
  - *Bagging = bootstrap aggregation learning*



Bagging

- De cada iteração *bootstrap*, gera-se um classificador
  - Em vez de apenas utilizar os classificadores para estimar o erro médio, guarda-se cada classificador gerado
  - Esses classificadores comporão o *ensemble*
    - Pode-se utilizar o erro estimado de cada classificador como peso de seus votos

Bagging

- O *bagging* pode produzir *ensembles* bastante robustos, porém é necessário utilizar **classificadores instáveis**
  - Um classificador é **instável** se uma pequena alteração nos dados podem alterar substancialmente o modelo induzido
  - Exemplo: árvores de decisão

Florestas aleatórias

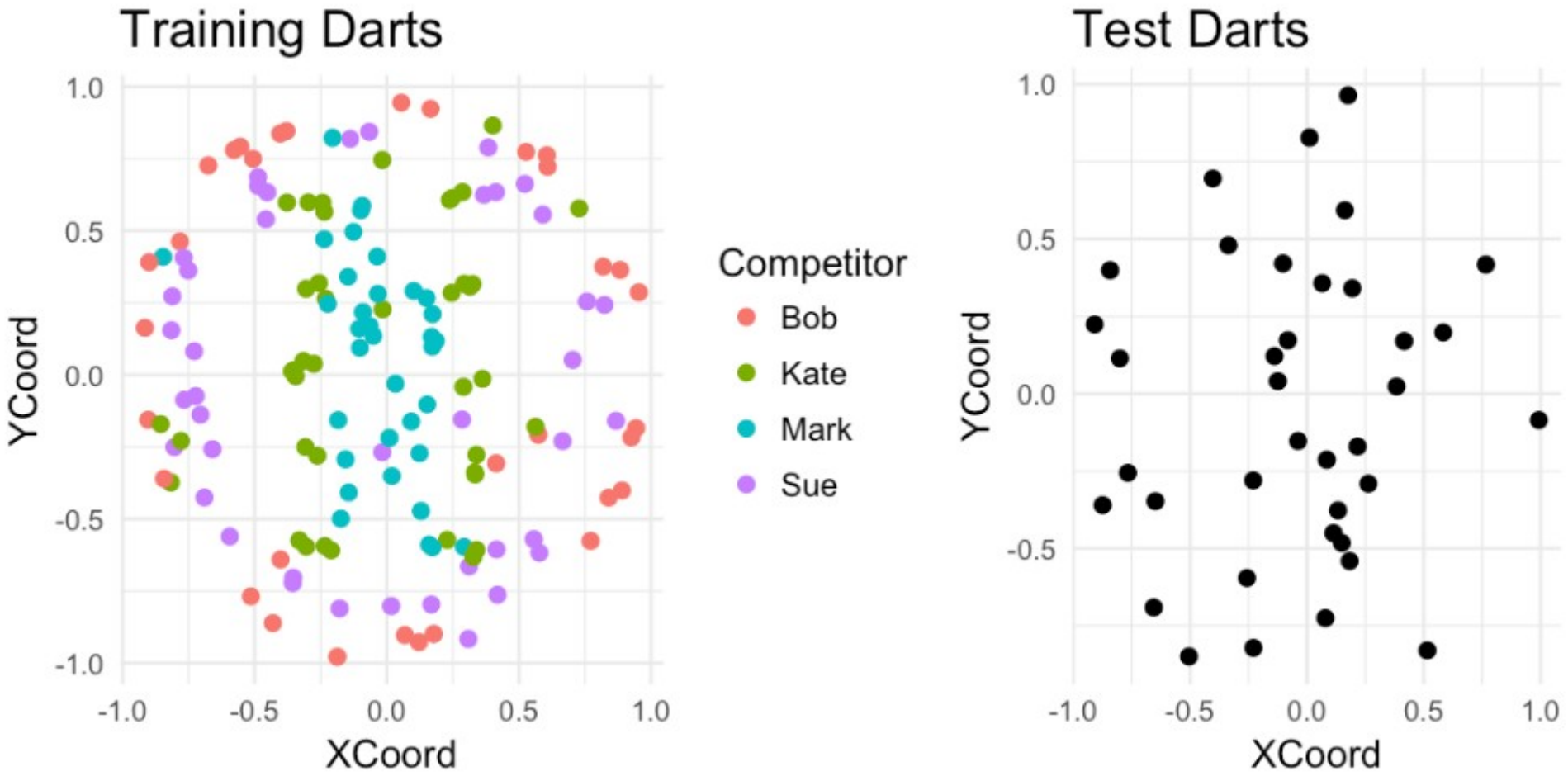
- Introduz diversidade manipulando o **espaço de características**
- Deriva do *bagging*
  - Utiliza sub-amostragem de exemplos
  - Utiliza **sub-amostragem de atributos**
    - Para cada nó da árvore, seleciona um conjunto de atributos aleatórios
    - Para um espaço com p atributos, pode-se selecionar $\sqrt{p}$ atributos

Stacking

- Manipulação **dos dados de entrada**
- Utiliza múltiplos níveis de classificadores e um **meta-classificador**
  - Classificadores no nível 0 são os classificadores--base
  - No nível 1 está um meta-classificador que irá decidir a classe do *ensemble* observando as respostas individuais dos classificadores-base

Stacking

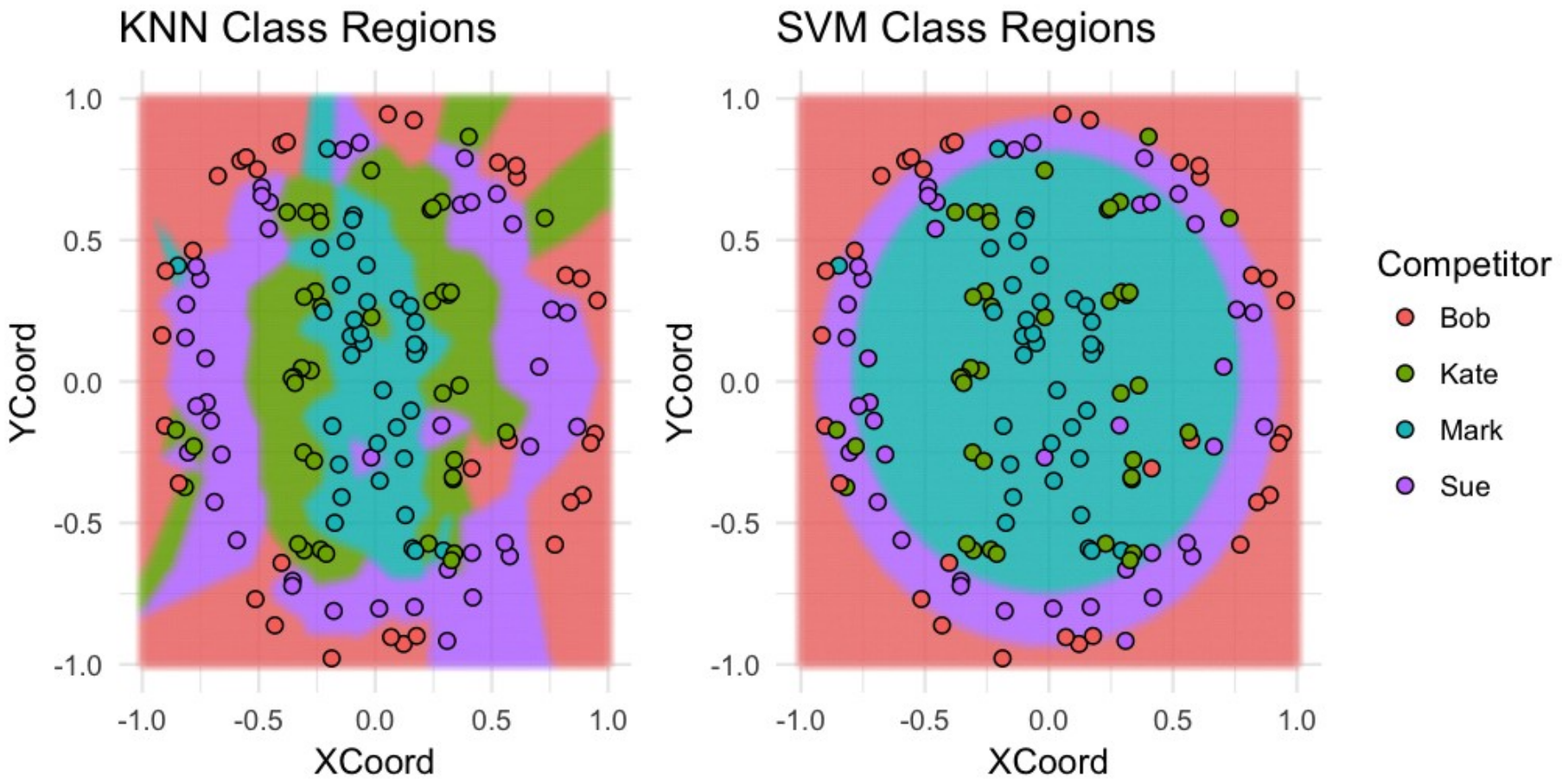
- Exemplo sintético



http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice/

Stacking

- Exemplo sintético



http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice/

Stacking

- Exemplo sintético
  - O *k*-NN e o SVM compõem o nível 0 do *ensemble*
  - No nível 1 poderíamos ter um segundo modelo (exemplo: outro SVM) que utiliza os atributos originais **e também** as saídas dos classificadores-base como atributos de entrada

Boosting

- O termo *boosting* refere-se a um conjuto de estratégias utilizadas para tornar aprendizes fracos em aprendizes melhores
  - Em particular, um dos métodos mais populares para *ensembles* é o AdaBoost (*Adaptive Boosting*)
- Para compreender o *boosting*, precisamos entender o conceito de **conjunto de dados com peso**

Boosting: Adaboost

- Técnica de *ensemble* baseada em manipulação dos dados de entrada
  - O conjunto de dados com peso $D = \{(X_1, y_1, w_1), (X_2, y_2, w_2), ..., (X_N, y_N, w_N)\}$ é, inicialmente, equivalente a um conjunto não ponderado
    - Isto é, $w_i = 1/N$ para todos os exemplos
  - Em cada iteração, um classificador y^j é induzido e os pesos são ajustados de acordo com o erro de y^j

Boosting: Adaboost

- Técnica de *ensemble* baseada em manipulação dos dados de entrada
  - O objetivo é que cada classificador y^j "foque no erro" cometido pelos classificadores anteriores
    - O peso pode ser diretamente utilizado na função de perda do modelo ou os exemplos com maior peso podem ser duplicados
  - Os pesos dos exemplos incorretamente classificados são aumentados em cada passo

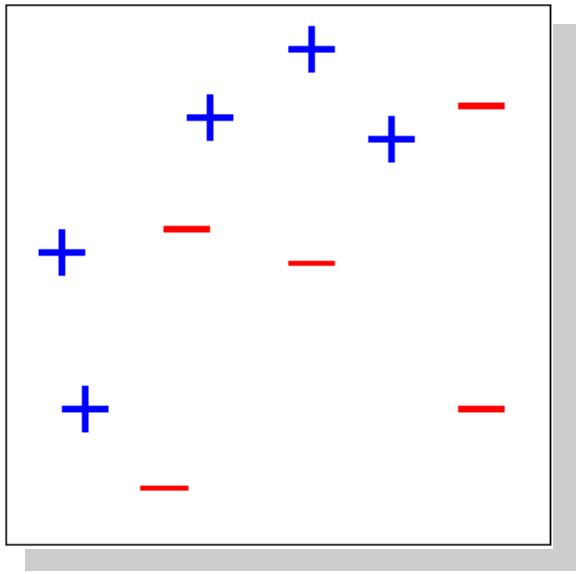
Boosting: Adaboost

- Exemplo:
  - A cada iteração, uma amostra de treinamento é selecionada aleatoriamente com repetição
    - Semelhante ao *bootstrap*, mas com distribuição dependente dos pesos
    - Amostras com maior peso têm maior probabilidade de serem selecionadas!

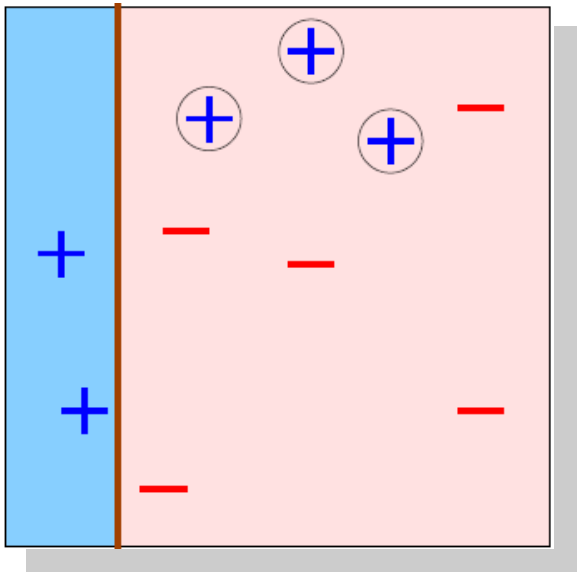
Boosting: Adaboost

- Exemplo:
  - Um modelo y^j é induzido e guardado
  - As instâncias incorretamente classificadas por y^j são ajustadas para que os pesos $w_i^{(j+1)}$ sejam **maiores**
  - Na iteração seguinte, o classificador $y^{(j+1)}$ será induzido para "aprender os conceitos que y^j não conseguiu absorver"

Boosting: Adaboost

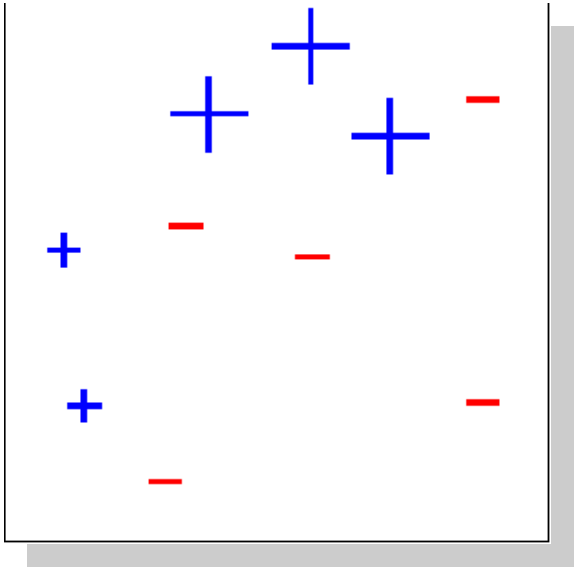


Exemplos

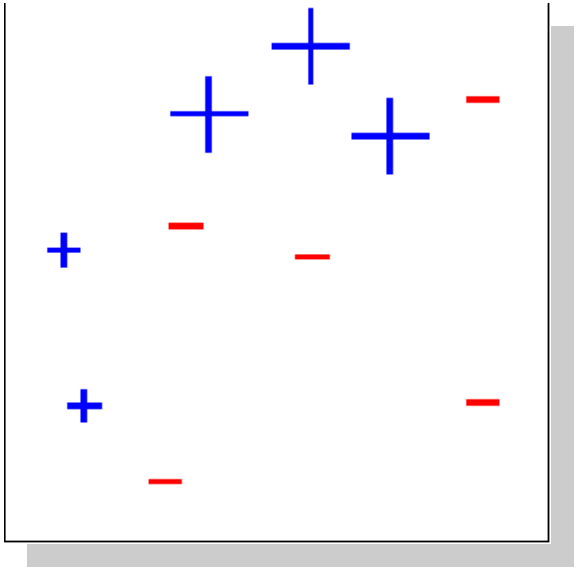


Modelo

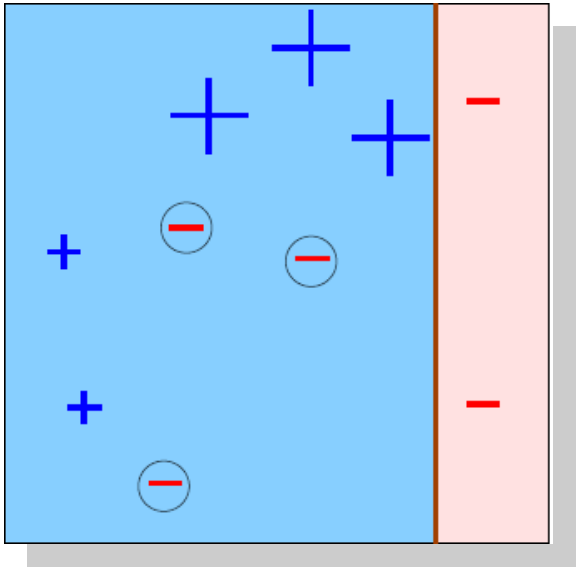
Pesos reajustados



Boosting: Adaboost

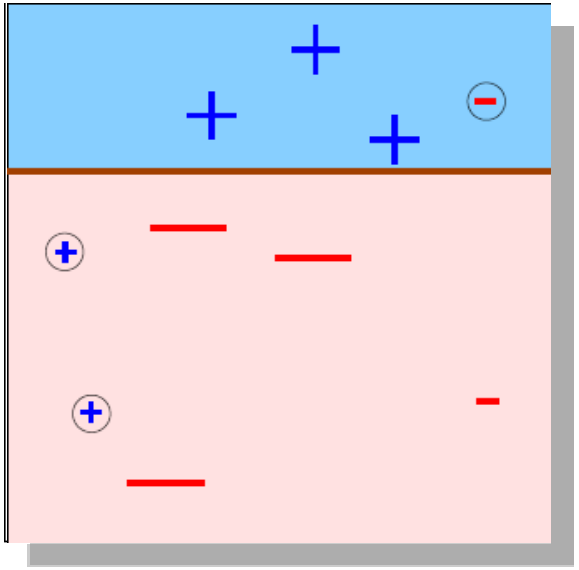


Exemplos



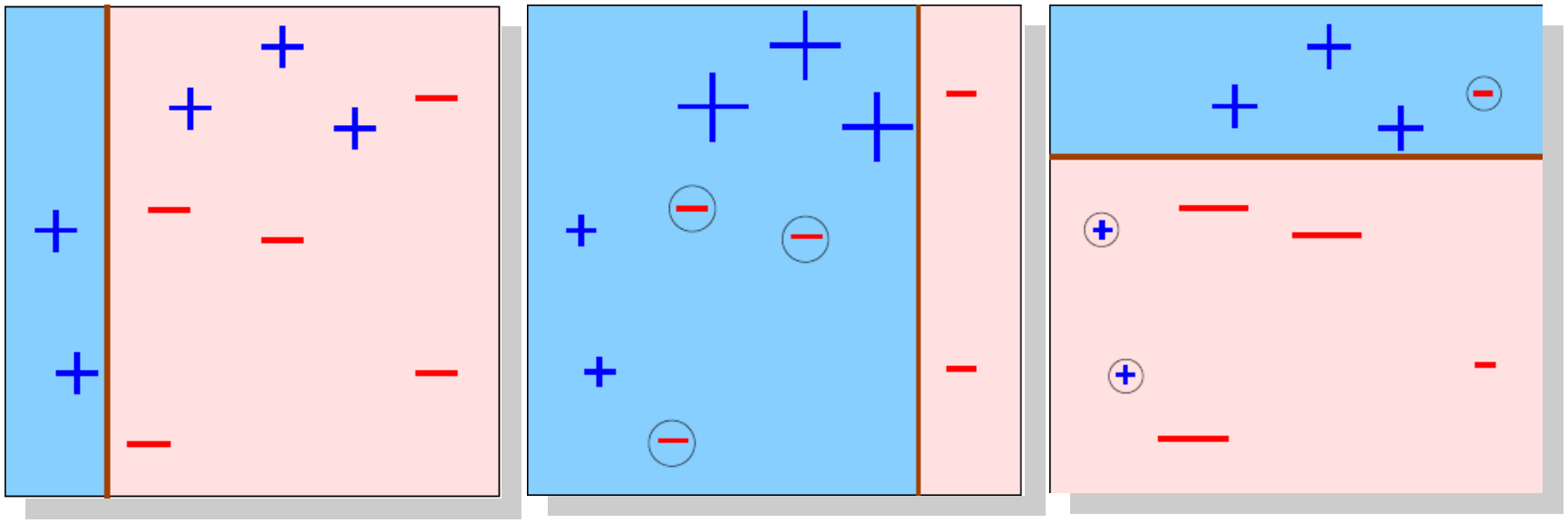
Modelo

Pesos reajustados



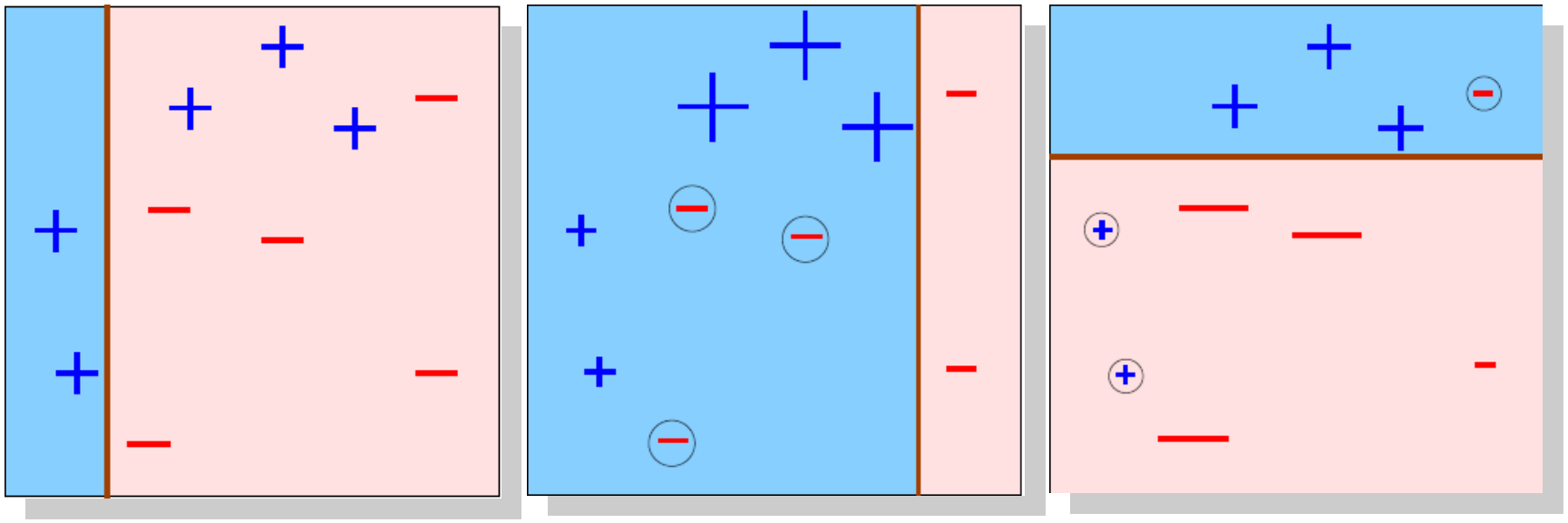
Boosting: Adaboost

- Ao final, o Adaboost terá produzido uma coleção de classificadores-base
- O resultado do *ensemble* pode ser o voto simples ou ponderado de cada classificador



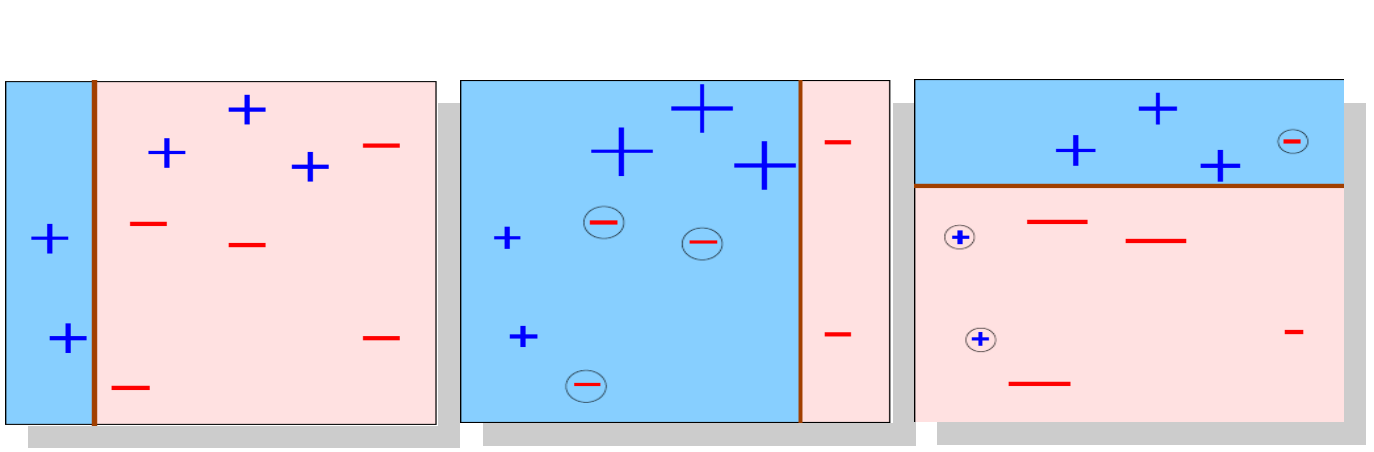
Boosting: Adaboost

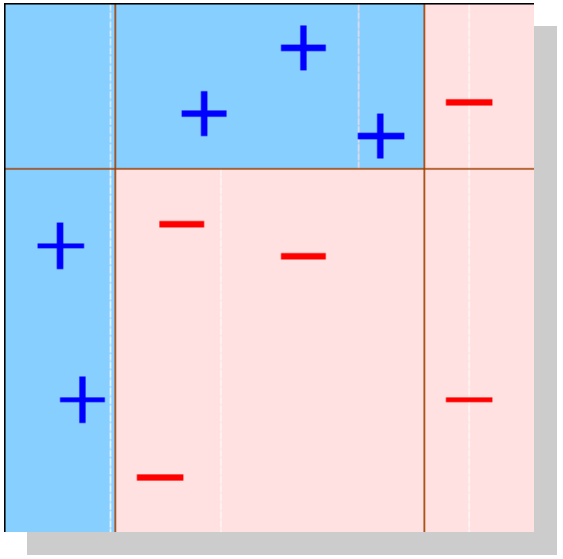
- Ao final, o Adaboost terá produzido uma coleção de classificadores-base
- O resultado do *ensemble* pode ser o voto simples ou ponderado de cada classificador



Boosting: Adaboost

- A combinação de classificadores com baixa acurácia através do AdaBoost pode produzir um *ensemble* de alta acurácia





function ADABOOST($examples$, L, K) **returns** a weighted-majority hypothesis
  inputs: $examples$, set of N labeled examples $(x_1, y_1), \ldots, (x_N, y_N)$
    L, a learning algorithm
    K, the number of hypotheses in the ensemble
  local variables: **w**, a vector of N example weights, initially $1/N$
    h, a vector of K hypotheses
    z, a vector of K hypothesis weights

  for $k = 1$ **to** K **do**
    $\mathbf{h}[k] \leftarrow L(examples, \mathbf{w})$
    $error \leftarrow 0$
    for $j = 1$ **to** N **do**
      if $\mathbf{h}[k](x_j) \neq y_j$ **then** $error \leftarrow error + \mathbf{w}[j]$
    for $j = 1$ **to** N **do**
      if $\mathbf{h}[k](x_j) = y_j$ **then** $\mathbf{w}[j] \leftarrow \mathbf{w}[j] \cdot error/(1 - error)$
    $\mathbf{w} \leftarrow$ NORMALIZE(**w**)
    $\mathbf{z}[k] \leftarrow \log(1 - error)/error$
  return WEIGHTED-MAJORITY(**h**, **z**)

Ensembles

- Resumo
  - Produzem coleções de classificadores que podem exceder a capacidade individual de um indutor de
    - Evitar o *overfitting*
    - Induzir modelos que exigiriam treinamento excessivamente longo
    - Representar modelos mais complexos do que a capacidade da linguagem

Ensembles

- Dificuldades
  - É extremamente importante garantir que os classificadores-base serão diversos
  - Deve-se tomar cuidado para que a introdução de diversidade não prejudique a acurácia dos classificadores-base
    - Os classificadores devem ter **erros independentes**