Exercícios de Comitês

Aprendizado de Máquina

Prof. Me. Otávio Parraga

Exercício 1 - Comitês de Aprendizes

Responda as questões abaixo:

(a) Por que *tree stumps* são utilizados em comitês de aprendizes baseados em Boosting?
Porque tree stumps são modelos de alto bias e baixa variância, o que os torna adequados para um processo voltado a diminuir apenas o bias.

(b) Por que Árvores de Decisão são utilizadas em comitês de aprendizes baseados em Bagging?
Porque Árvores de Decisão são modelos de alta variância e baixo bias, o que os torna adequados para um processo voltado a diminuir apenas a variância.

(c) Você treinou 5 classificadores utilizando algoritmos de aprendizagem diferentes, sendo que cada um desses classificadores possui acurácia próxima de 0.6. Dada essa situação, responda:

- **(I)** Supondo que os erros sejam altamente correlacionados, você espera melhoria na acurácia ao realizar um *ensemble* entre esses classificadores?
- Não, se os erros são altamente correlacionados, isso significa que os classificadores estão cometendo os mesmos erros, e um ensemble não ajudará a melhorar a acurácia.
- **(II)** Supondo que os erros sejam pouco correlacionados, você espera melhoria na acurácia ao realizar um *ensemble* entre esses classificadores?
- Sim, se os erros são pouco correlacionados, um ensemble pode ajudar a melhorar a acurácia, pois os classificadores podem compensar os erros uns dos outros.

(d) Explique como introduzir diversidade em um comitê de aprendizes a partir de:

- **(I)** Manipulação do *dataset*.
- Pode-se usar técnicas como amostragem com reposição (bootstraping) para criar diferentes subconjuntos de dados para treinar cada aprendiz, ou usar diferentes conjuntos de dados de treinamento.
- **(II)** Manipulação de *features*.

- Pode-se usar técnicas como seleção de características (feature selection) ou criação de características (feature engineering) para introduzir diversidade entre os aprendizes.
- **(III)** Manipulação do algoritmo de aprendizagem.
- Pode-se usar diferentes algoritmos de aprendizagem ou variar os hiperparâmetros do mesmo algoritmo para criar diversidade entre os aprendizes.

(e) Qual a diferença do método de combinação de aprendizes por seriação e por votação?
A combinação por votação é geralmente usada para tarefas de classificação, onde cada aprendiz vota na classe que acredita ser a correta, e a classe com mais votos é escolhida como a previsão final. A combinação por seriação é geralmente usada para tarefas de regressão, onde as previsões dos aprendizes são combinadas (por exemplo, pela média) para produzir uma previsão final.

Exercício 2 - Bagging e Boosting

Identifique cada asserção abaixo como verdadeira ou falsa:

(a) Bagging introduz diversidade no conjunto de dados por manipulação de *features*.
Falso. Bagging introduz diversidade no conjunto de dados por manipulação do dataset, especificamente através de amostragem com reposição (bootstraping).

(b) Ao usar Bagging por mais de uma iteração (*round*), mesmo que o aprendiz base utilizado tenha expressividade para aproximar apenas funções lineares, é possível aproximar uma função não-linear.
Verdadeiro. Através da combinação de múltiplos aprendizes, cada um treinado em um subconjunto diferente dos dados, é possível capturar padrões mais complexos e não-lineares.

(c) Em Bagging, é esperado que alguns dos aprendizes treinados tenham uma taxa de erro elevada devido ao peso aplicado em observações cujo erro dos aprendizes anteriores foi elevado.
Falso. Em Bagging, todos os aprendizes são treinados de forma independente e com igual peso, sem considerar o desempenho dos aprendizes anteriores. A técnica que ajusta os pesos com base no desempenho anterior é o Boosting.

(d) O processo de amostragem com reposição que é feito em cada rodada de Bagging se chama *bootstraping*.
Verdadeiro. Bootstraping é a técnica de amostragem com reposição usada em Bagging para criar diferentes subconjuntos de dados para treinar cada aprendiz.

(e) *Random Forests* são diferentes do algoritmo de Bagging básico pois introduzem também uma manipulação das *features* do conjunto de dados para inserir variabilidade.

Verdadeiro. *Random Forests* introduzem variabilidade ao selecionar aleatoriamente um subconjunto de características em cada divisão de árvore, além de usar amostragem com reposição nos dados.

(f) *Out of Bag Error (OOB)* é computado em *Random Forests* utilizando apenas as observações que não foram consideradas na *bootstrap sample* que foi usada para treinar cada uma das árvores. OOB é um bom estimador para o erro em validação.
Verdadeiro. OOB é uma forma eficaz de estimar o erro de validação em *Random Forests*, pois utiliza dados que não foram usados no treinamento de cada árvore.

(g) O termo de importância, usualmente denotado por α_i, do algoritmo de Boosting determina o peso do voto de cada um dos aprendizes treinados. Essa importância é inversamente proporcional à taxa de erro desse aprendiz.
Verdadeiro. No Boosting, aprendizes com menor taxa de erro recebem maior peso (importância) na combinação final das previsões.