K-NEAREST

Para conjuntos de dados bidimensionais, também podemos ilustrar a previsão para todos os pontos de teste possíveis no plano xy. Colorimos o plano de acordo com a classe que seria atribuída a um ponto nesta região. Isso nos permite visualizar a fronteira de decisão, que é a divisão entre onde o algoritmo atribui a classe 0 em comparação com onde atribui a classe 1.

Considerar cada vez mais vizinhos leva a uma fronteira de decisão mais suave. Uma fronteira mais suave corresponde a um modelo mais simples. Em outras palavras, usar poucos vizinhos corresponde a uma alta complexidade do modelo (como mostrado no lado direito da Figura 2-1), e usar muitos vizinhos corresponde a uma baixa complexidade do modelo (como mostrado no lado esquerdo da Figura 2-1). Se considerarmos o caso extremo em que o número de vizinhos é igual ao número de todos os pontos de dados no conjunto de treinamento, cada ponto de teste teria exatamente os mesmos vizinhos (todos os pontos de treinamento) e todas as previsões seriam iguais: a classe que é mais frequente no conjunto de treinamento.

Embora os gráficos do mundo real raramente sejam muito suaves, ainda podemos reconhecer algumas das características de overfitting (sobreajuste) e underfitting (subajuste) (observe que, como considerar menos vizinhos corresponde a um modelo mais complexo, o gráfico está espelhado horizontalmente em relação à ilustração na Figura 2-1). Ao considerar um único vizinho mais próximo, a previsão no conjunto de treinamento é perfeita. Mas quando mais vizinhos são considerados, o modelo se torna mais simples e a precisão no treinamento diminui. A precisão no conjunto de testes ao usar um único vizinho é menor do que ao usar mais vizinhos, indicando que o uso do único vizinho mais próximo resulta em um modelo muito complexo. Por outro lado, ao considerar 10 vizinhos, o modelo é muito simples e o desempenho é ainda pior. O melhor desempenho está em algum lugar no meio, usando cerca de seis vizinhos. Ainda assim, é bom ter em mente a escala do gráfico. O pior desempenho é em torno de 88% de precisão, o que ainda pode ser aceitável.

A princípio, há dois importantes parâmetros para o classificador KNeighbors: o número de vizinhos e como você mede a distância entre dois pontos. Na prática, usar um número pequeno de vizinhos como três ou cinco funciona bem, mas você pode certamente ajustar estes parâmetros.

Distância euclidiana é usado, que funciona bem em vários settings.

Um dos pontos fortes do k-NN é que o modelo é fácil de entender, e usualmente nos dá uma boa performance sem muitos ajustes. Usando este algoritmo é um bom método inicial para tentar antes de considerar modelos mais avançados. Construir este modelo é bem rápida, mas quando o conjunto de treinamento é muito grande, a previsão pode ser bem lenta. Esta abordagem normalmente não performa tão bem em datasets com muitas características, e performa bem mal quando as características são nulos a maior parte do tempo.

ENSEMBLES OF DECISION TREES

Ensembles são métodos que combinam múltiplos modelos de machine learning para criar modelos mais poderosos. Há muitos modelos que pertencem a esta categoria, sendo o mais conhecido deles o Random Forests. Como vimos, o principal ponto fraco do modelo de decision trees é sua capacidade de overfiting da conjunto de dados de treinamento. Um modelo RF nada mais é do que uma coleção de discion trees independentes entre si. A ideia por de trás é que cada árvore faça relativamente um bom trabalho em prever, mas ele irá provavelmente se “sobreajustar” aos dados de treinamento. Se construirmos muitas árvores, podemos mitigar este problema facilmente pegando a média dos resultados.

O que significa uma árvore ser diferente da outra (ou ser independente)?

O modelo ganha este nome pois aleatoriza a construção das árvores para garantir que a independência entre elas.

Há duas formas de aleatorizar as árvores:

1. Selecionando os pontos dos dados usados para construir a árvore.
2. Selecionando as características em cada rodada de teste.

RF se sobreajusta menos do que qualquer árvore individualmente, e provê um limite de decisão muito mais intuitivo.