UECE - Mestrado Acadêmico em Ciência da Computação
Fundamentos e Análise de Dados

Classificação de vinhos a partir das componentes físico-químicas utilizando KNN

Lucas Coitinho Brito

lucas.brito@aluno.uece.br

8 de maio de 2019

Resumo

Inteligência Artificial (IA) é uma das áreas que mais crescem no campo de pesquisa da ciência da computação. O foco deste trabalho é a utilização de técnicas de aprendizagem de máquina, em particular, as técnicas baseadas na regra do vizinho mais próximo. O k-Nearest Neighbor (KNN) é um método supervisionado de classificação de dados baseado na proximidade de seus vizinhos em um espaço amostral, e o *dataset* escolhido está relacionado com variantes brancas do vinho "Vinho Verde". foi dividido de tal forma que: 70% das amostras fosse separadas para treinamento e o restante para teste. Foram testadas duas medidas de distância diferentes (Euclidiana e Manhattan) e avalidas sob as métricas de precisão, *recall* e f-*score*. O KNN por se tratar de um procedimento simples, não tem uma acurácia satisfatória para utilizar de forma confiável no âmbito comercial no seu estado clássico, exigindo modificações posteriores. A melhor acurácia do método para este *dataset* foi de 63%.

Palavras-chave— KNN, aprendizado de máquina, classificação, vinho

Resumo

Artificial Intelligence (AI) is one of the fastest growing areas in the field of computer science research. The focus of this work is the use of machine learning techniques, in particular, as techniques in the nearest neighbor rule. The k-Nearest Neighbor (KNN) is a supervised method of data classification based on the proximity of its neighbors in a sample space, the dataset chosen is related to white variants of the "Vinho Verde"wine. was divided in such a way that: 70% of the samples were separated for training and the remainder for testing. Two different distance measures (Euclidean and Manhattan) were evaluated and evaluated under the precision metrics, recall and f-score Because KNN is a simple procedure, it does not have a satisfactory accuracy to use reliably in the commercial scope in its classic state, requiring later modifications. The best accuracy of the method for this dataset was 63%.

Keywords— KNN, machine learning, classification, wine

1 Introdução

Inteligência Artificial (IA) é uma das áreas que mais crescem no campo de pesquisa da ciência da computação. Ela pode ser definida de forma simplista como "a inteligência exibida por qualquer coisa que tenha sido construída pelo homem", no entanto, este conceito leva ao questionamento do que seria "inteligência".

A Inteligência Artificial tem demonstrado bastante sucesso em áreas onde é possível se construir abstrações do mundo real. Geralmente, tais abstrações se baseiam em modelos simples que buscam modelar sistemas complexos do mundo real, tentando imitar os sistemas naturais em algumas de suas características. A área de IA é subdividida em diversos ramos, sendo os principais: a aprendizagem de máquina, IA simbólica, as redes neurais artificiais (RNA), dentre outros.

O foco deste trabalho é a utilização de técnicas de aprendizagem de máquina, em particular, as técnicas baseadas na regra do vizinho mais próximo. Técnicas de aprendizagem de máquina têm como objetivo a classificação de padrões. Um algoritmo de aprendizagem de máquina para classificação de padrões tem como meta classificar padrões desconhecidos dentre as várias classes possíveis de um determinado problema.

Em 1966, a regra do vizinho mais próximo, Nearest Neighbor Rule (NN), foi proposta por Cover e Hart. Eles mostraram que essa técnica era muito eficiente para problemas de classificação de padrões. Um outro motivo para a utilização da regra do vizinho mais próximo seria que ela é simples e de fácil implementação, mas possui alguns pontos fracos, tais como: (a) requerer uma grande quantidade de memória, pois necessita armazenar todos os padrões de treinamento; (b) o algoritmo requer uma grande quantidade de tempo computacional; (c) é sensível a ruído e falsos padrões.

Os algoritmos baseados na regra do vizinho mais próximo necessitam de recursos computacionais significantes, dentre esses os mais requisitados são recursos de memória e de processamento, como é o caso do algoritmo K-Nearest Neighbor (KNN) [Webb 2003].

2 Metodologia

2.1 Treinamento e Teste

A fase de treinamento é bastante simples e consiste em, dado um conjunto de treinamento $\mathbf{T} = t_1, \ldots, t_n$ onde n é o numero de amostras de treino armazenadas em $\mathbf{T}$ e $\mathbf{P} = p_1, \ldots, p_m$ onde m é o número de amostras armazenadas no conjunto de teste. Seja um X qualquer, que pertença à base de treinamento ou à base de teste, este é um vetor de atributos que contem uma classe cuja amostra pertence.

Durante a fase de teste, o algoritmo é que se avalia o desempenho do classificador para a classificação de novos padrões. Para isso, dada uma base de teste P deve-se calcular a taxa de erro ou acurácia do classificador para um dado conjunto de teste.

2.2 *K-Nearest Neighbor*

O k-Nearest Neighbor (KNN) é um método supervisionado de classificação de dados baseado na proximidade de seus vizinhos em um espaço amostral [Dakhlaoui et al. 2012]. Para estimar a classe de um padrão desconhecido X, o algoritmo *K-Nearest Neighbor*, KNN daqui em diante, calcula os k-vizinhos mais próximos a X e classifica-o como sendo da classe mais frequente dentre os seus k-vizinhos.

Durante a fase de classificação do KNN, algumas vezes ocorre um problema, onde, dado um padrão de teste X, os seus k-vizinhos mais próximos são de uma mesma classe e o algoritmo não consegue decidir com qual classes dos k-vizinhos ele deve comparar o padrão X. Sendo k o número de vizinhos, podemos aumentar esse número impar de forma a evitar empates.

Segundo [Webb 2003], "O KNN é um método simples de estimação de densidade". O KNN recebe essa denominação pelo ao fato dele estimar a densidade local de padrões de treinamento na vizinhança de um padrão desconhecido durante a classificação.

Nesse algoritmo, existe um parâmetro chamado `k`, que indica o número de vizinhos que serão usados pelo algoritmo durante a fase de teste. O parâmetro `k` faz com que algoritmo consiga uma classificação mais refinada, porém o valor ótimo de `k` varia de um problema para o outro, o que faz com que, para cada base de dados, sejam testados vários valores diferentes de forma a descobrir qual o melhor valor de `k` para determinado problema.

O KNN determina um volume V que contém os k-vizinhos mais próximos centrados em um padrão X, o qual se deseja classificar. Por exemplo, se X_k é o k-ésimo vizinho de X, então V será uma esfera centrada em X e com raio igual à distância Euclidiana entre X e X_k, ou seja, $||X - X_k||$.

Um exemplo simples de como funciona a classificação feita pelo algoritmo KNN pode ser visto na Figura 1, onde o padrão desconhecido representando pelo círculo vermelho está entre os padrões da classe amarela, e os padrões da classe roxa. A tarefa do KNN é classificar o padrão sinalizado em vermelho como sendo pertencente a uma das classes exibidas no exemplo.

Para tal tarefa, temos estratégias que selecionam os k vizinhos com menor dissimilaridade, aplicando pesos positivos na sua classe correspondente afim de classificar o padrão desconhecido. Note que, quando o número de $k = 3$ podemos afirmar, baseado em votos, que o padrão desconhecido é da classe roxa. Porém, quando $k = 6$ a classe que melhor representa é a amarela.

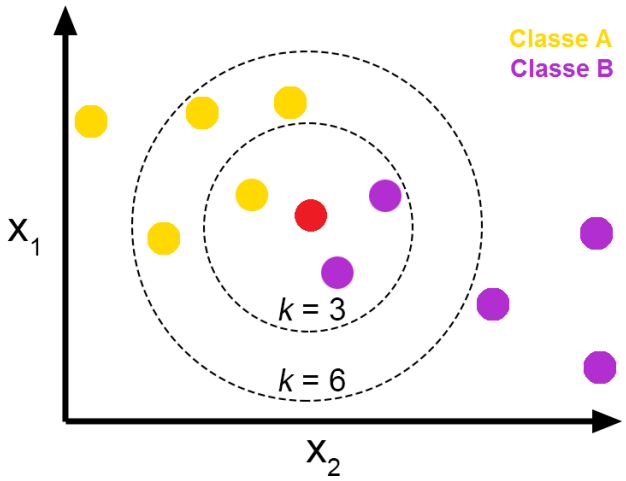

Figura 1: Exemplo de classificação com duas classes: k=3 e k=6

No algoritmo 1 podemos observar o primeiro passo é selecionar quantas amostras do *dataset* farão parte do em treinamento e teste, de tal forma que possamos predizer a classe de uma amostra do conjunto de teste a partir do conjunto de treinamento utilizando métricas de distância e escolha das classes mais promissoras.

Idealmente, o ponto em que as amostras são dividas entre conjunto de teste e conjunto de treinamento é escolhido de maneira aleatória. No entanto, a proporção escolhida é em torno dos 30%.

Algoritmo 1 *K-Nearest Neighbor*

1: **Input:** *Dataset* com n amostras;
2: **Output:** Classe da amostra T_i;
3: Faça um conjunto de teste P e um de treinamento T a partir de um ponto de divisão aleatório r, sendo $r < n$, no *dataset*;
4: **for** i . . . n **do**
5: Calcular distância entre amostra x e T_i, sendo x $\in P$
6: **end for**
7: **return** conjunto dos k vizinhos de T mais próximos da amostra x

3 Experimentos

Como repositório sugerido, utilizamos a UCI Machine Learning Repository [Asuncion and Newman 2007] para se fazer o estudo comparativo de desempenho do algoritmo. Esse repositório é público e é utilizado para avaliar algoritmos de aprendizado de máquina.

O conjunto escolhido está relacionado com variantes **brancas** do vinho "Vinho Verde". Devido a questões de privacidade e logística, apenas as variáveis físico-químicas (insumos) e sensoriais (a saída) estão disponíveis (por exemplo, não há dados sobre tipos de uva, marca de vinho, preço de venda do vinho, etc.). Para mais detalhes, consulte [Cortez et al. 2009].

As entradas incluem testes objetivos (por exemplo, valores de PH) e a saída é baseada em dados sensoriais (mediana de pelo menos 3 avaliações feitas por especialistas em vinho). Cada perito classificou a qualidade do vinho entre 0 (muito ruim) e 10 (muito bom).

AF	AV	AC	AR	Cl	SO_2 L	SO_2 T	D	pH	S	C_2H_6O	Q
7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6

Tabela 1: AF: acidez fixa, AV: acidez volátil, AC: ácido cítrico, AR: açúcar residual, Cl: cloretos, SO_2 L: dióxido de enxofre livre SO_2 T: total de dióxido de enxofre, D: densidade, pH: potencial Hidrogeniônico, S: sulfatos, C_2H_6O: álcool, Q: qualidade

Durante os experimentos, visando à validação dos nossos resultados em relação ao resultado dos autores, os conjuntos de treinamento e teste foram mantidos da mesma forma que estão dispostos em sua fonte.

Na Tabela 1, podemos visualizar as características da base dados escolhida. Sendo estes 11 atributos referentes ao comportamento físico-químico dos vinhos e num total de 4898 amostras.

Para efeitos de teste, poderíamos utilizar várias métricas que nos fornecem distâncias entre as amostras vizinhas que influenciam na classificação, dentre elas: Euclidiana, Manhattan, Chebychev etc. Para esta aplicação, utilizaremos apenas as duas primeiras medidas.

Segundo [Dasarathy 1991] proporciona eficiência e produtividade. A distância é calculada como a raiz das diferenças quadradas entre coordenadas de pares de pontos de dados no espaço euclidiano. A equação 1 descreve, genericamente:

$$dist_{L_2}(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \tag{1}$$

A distância de Manhattan faz alusão ao formato quadriculado da maior parte das ruas na ilha de Manhattan. Tal configuração faz com que a menor distância a ser percorrida por um carro que vai de um ponto a outro na cidade tenha como valor aquele número fornecido pela métrica L_1. Logo, definimos:

$$dist_{L_1}(X,Y) = |x_1 - x_2| + |y_1 + y_2| \tag{2}$$

A distância entre dois pontos no espaço, pode ser afetada pela distribuição dos dados. Considere um exemplo de de classificação com duas classes, em que uma amostra de Classe 1 é escolhida (preto)

junto com seus 10 vizinhos mais próximos (verde preenchido). Na primeira figura, os dados não são normalizados, enquanto no segundo é.

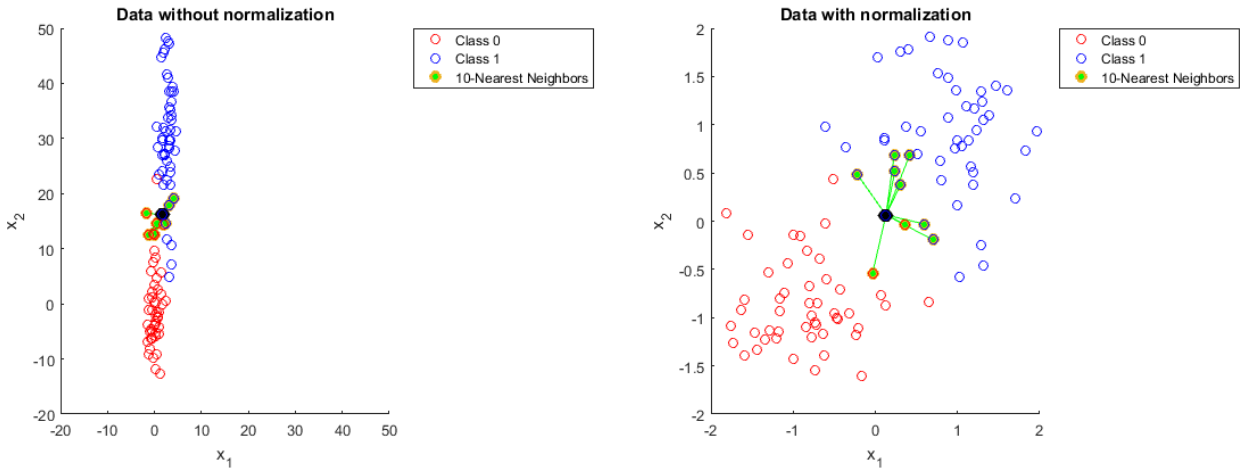

Figura 2: Dados normalizados são mais justos com os cálculos de distâncias

Observe que, sem normalização, todos os vizinhos mais próximos estão alinhados na direção do eixo com o menor intervalo, ou seja, x1. A normalização resolve este problema.

Devemos mencionar o fato de que o algoritmo KNN clássico baseia-se na votação por maioria com base na associação de classe de amostras k mais próximas para uma determinada amostra P_i do grupo de teste.

Na equação 3, temos o conjunto de classes C e c_i sendo a quantidade de vizinhos pertencentes a classe i próximos a amostra P_i. Podemos definir que os k vizinhos mais próximos é um subconjunto de C, após uma ordenação, tal que $\{c_0, c_1, \ldots, c_k\}$ são as k primeiras posições desse vetor.

$$ord(C) = c_1, c_2, c_3, \ldots, c_n \tag{3}$$

Outra estratégia que utilizada neste trabalho foi uma votação ponderada a partir do inverso de sua distância (Equação 4). Nesse caso, os vizinhos mais próximos de uma amostra de teste P_i terão uma influência maior do que os vizinhos mais distantes. Seja:

$$ord(S) = s_1, s_2, s_3, \ldots, s_n \tag{4}$$

S é um subconjunto ordenado de T, tal que o inverso da distância entre amostra P_i e um elemento do conjunto de treinamento T_j seja a medida de dissimilaridade entre eles.

4 Resultados

Nesta seção são analisados os resultados obtidos pelo algoritmo KNN. Os experimentos foram realizados em um MacBook Pro com 16gb de memória ram DDR3 a 1600mhz, processador Intel (R) Core (TM) CPU i5 @ 2.5GHz e sistema operacional macOS Mojave. Todos os testes foram realizado com uma implementação do algoritmo na linguagem *python* (conda-forge versão 2.7.15) e utilizando a biblioteca `scikit-learn`.

O *dataset* foi dividido de tal forma que 70% das amostras fosse separadas para treinamento e o restante para teste. O tempo não foi avaliado no escopo dos teste e portanto não será levado em conta. No entanto, a sua média de execução para um determinado k é de poucos segundos.

Após um teste unitário com $k = 5$, e variando os parâmetros de distância e classificadores, obtemos a seguinte tabela:

A tabela 2 mostra que o resultado utilizando os parâmetros `distance`, que é o inverso da distância como critério de voto e distância euclidiana temos uma acurácia de 63% neste dataset.

Portanto, a partir do resultado mais promissor, obtemos a seguinte matriz e confusão:

weight	distance	accuracy
uniform	L_1	53.4%
uniform	L_2	54%
distance	L_2	63.4%
distance	L_2	62.5%

Tabela 2: Variações do método KNN com $k = 5$

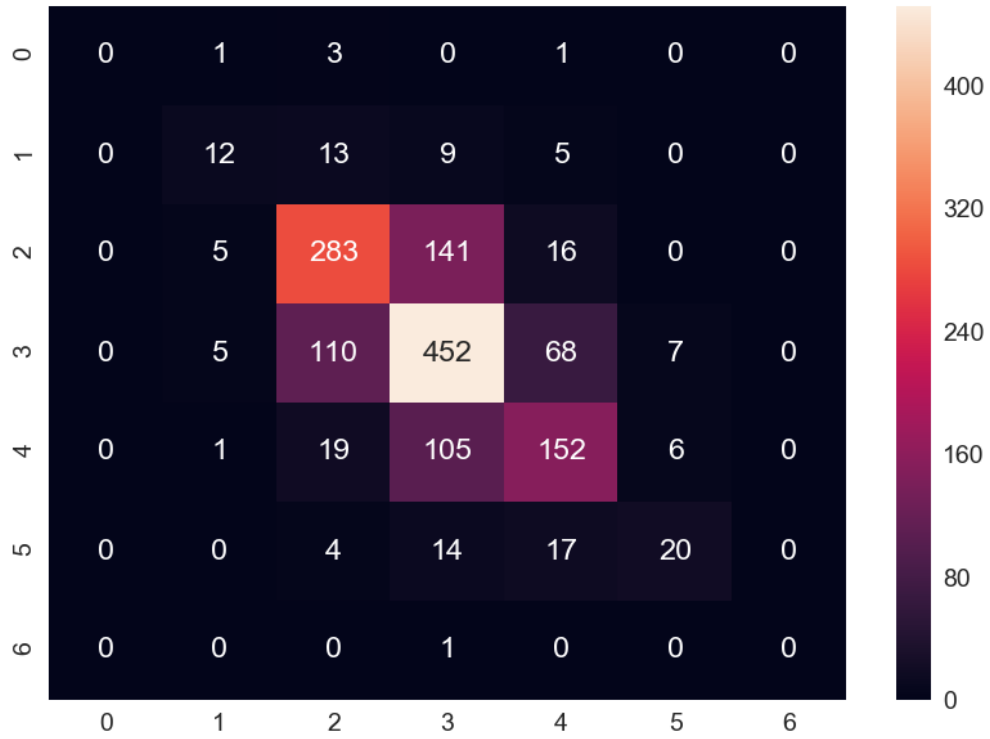

Figura 3: Matriz de confusão referente as classes preditas e classe verdadeiras

É um fato constatável que, o *dataset* possui grande parte dos seus vinhos concentrados entre as notas 2 e 3. Sendo a classe 3 mais comum e o acerto do algoritmo é preciso para esta classe.

Na figura 4, podemos ainda observar a evolução do algoritmo conforme a aumentamos a vizinhança de análise da amostra desconhecida. A medida em que o parâmetro k cresce, a acurácia cai.

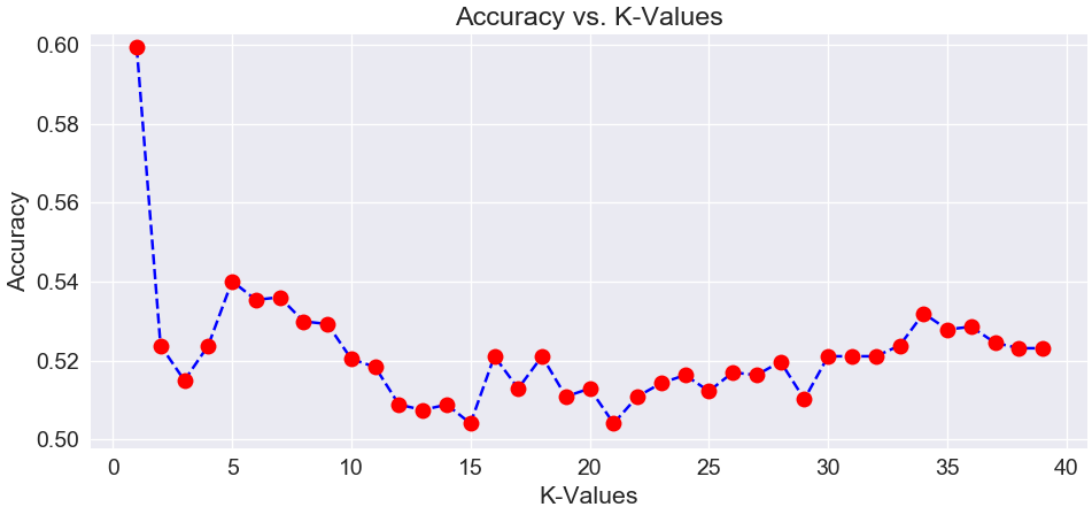

Figura 4: Matriz de confusão referente as classes preditas e classe verdadeiras

Como parte do relatório, devemos aplicar métricas de avaliação no método afim de qualificar sua análise, seja:

classe	`precision`	`recall`	`f-score`	`support`
3	0.00	0.00	0.00	5
4	0.50	0.31	0.38	39
5	0.66	0.64	0.6	445
6	0.64	0.71	0.67	642
7	0.61	0.55	0.58	283
8	0.54	0.38	0.45	55
9	0.00	0.00	0.00	1

Tabela 3: Resultados para cada classe dos parâmetros: `precision`, `recall`, `f-score`, `support`

A tabela 3 mostra as métricas utilizadas para avaliar o algoritmo, portanto, `precision` é a relação $\frac{\texttt{TP}}{\texttt{TP+FP}}$ onde `TP` é o número de positivos verdadeiros e `FP` o número de falsos positivos. A precisão é intuitivamente a capacidade do classificador de não rotular como positiva uma amostra que é negativa.

O `recall` é a relação $\frac{\texttt{TP}}{\texttt{TP+FN}}$ onde `TP` é o número de positivos verdadeiros e `FN` o número de falsos negativos. O `recall` é intuitivamente a capacidade do classificador de encontrar todas as amostras positivas.

O `f-score` pode ser interpretado como uma média harmônica ponderada de `precision` e `recall`, em que `f-score` atinge seu melhor valor em 1 e o pior escore em 0.

E finalmente, `suporte` é o número de ocorrências de cada classe no conjunto de teste.

4.1 Conclusão

Como visto na Seção anterior, o melhor resultado foi obtido quando k=5 e utilizando a distância de manhattan com ponderação, resultando em 64.3% de acerto.

Deste modo, podemos afirmar que o KNN não é eficiente em sua forma clássica para este problema e necessita de incrementos no seu procedimento para que ele se torne mais "acurado".

Referências

[Asuncion and Newman 2007] Asuncion, A. and Newman, D. (2007). Uci machine learning repository.

[Cortez et al. 2009] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553.

[Dakhlaoui et al. 2012] Dakhlaoui, H., Bargaoui, Z., and Bárdossy, A. (2012). Toward a more efficient calibration schema for hbv rainfall–runoff model. *Journal of hydrology*, 444:161–179.

[Dasarathy 1991] Dasarathy, B. V. (1991). Nearest neighbor ({NN}) norms:{NN} pattern classification techniques.

[Webb 2003] Webb, A. R. (2003). *Statistical pattern recognition*. John Wiley & Sons.