Comparação de Métricas em KNN para Classificação de Dígitos Manuscritos

Heloisa Benedet Mendes GRR20221248
João Pedro Vicente Ramalho GRR20224169
Luan Marko Kujavski GRR20221236
Departamento de Informática
Universidade Federal do Paraná – UFPR
Curitiba, Brasil
{hbm22, jpvr22, lmk22}@inf.ufpr.br

I. Introdução

A classificação de dígitos manuscritos é um problema clássico em aprendizado de máquina, frequentemente utilizado como *benchmark* para avaliar algoritmos de classificação. O *dataset Digits* consiste em imagens de dígitos de 0 a 9, cada uma com 8×8 pixels, totalizando 64 características por amostra. Neste trabalho, investigamos o comportamento do classificador KNN quando submetido a diferentes métricas de distância e distintas proporções de divisão entre conjuntos de treinamento e teste. Em paralelo, um classificador linear baseado em SGD é treinado para comparar sua performance com a abordagem KNN.

II. Metodologia

A. Descrição do Dataset

O *dataset* utilizado é o *Digits*, disponível na biblioteca `scikit-learn`. Ele contém 1.797 amostras de dígitos manuscritos, cada uma representada por um vetor de 64 dimensões (imagem 8×8 achatada). As classes variam de 0 a 9, totalizando 10 categorias. A distribuição das classes é aproximadamente balanceada.

B. Pré-processamento

Antes de treinar qualquer classificador, aplicamos `StandardScaler` para normalizar cada característica em média zero e variância unitária.

C. Divisão Treino & Teste

Para avaliar o impacto do tamanho do conjunto de teste, realizamos experimentos com cinco diferentes proporções: 30%, 40%, 50%, 60% e 70% de dados reservados para teste, e o complemento (1 - test_size) para treino. A divisão foi feita via `train_test_split` com *random_state* padrão (aleatório). Para cada proporção, os dados são divididos, escalonados e então utilizados nos classificadores.

D. Classificador KNN

Para o classificador KNN foram considerados os seguintes parâmetros:

- Número de vizinhos (K): $\{1, 3, 5, 7, 9\}$.
- Métricas de distância:
  - **Euclidiana**
  - **Manhattan**
  - **Cosseno**

O procedimento geral para cada experimento KNN é:

1) Dividir o conjunto de dados em treino e teste conforme a proporção especificada.
2) Ajustar o `StandardScaler` nos dados de treino e transformar treino e teste.
3) Instanciar `KNeighborsClassifier(n_neighbors=K, metric=...)` para o valor de K e métrica em questão.
4) Treinar com os dados de treino escalonados e predizer as classes do conjunto de teste.
5) Calcular a acurácia (accuracy $= \frac{\text{número de previsões corretas}}{\text{número de amostras de teste}}$).
6) Armazenar o resultado em um objeto JSON para posteriores análises.

E. Classificador Linear (SGD)

Além do KNN, treinamos um classificador linear simples baseado em `SGDClassifier` (Sem Kernel), sem ajuste de hiperparâmetros — ou seja, usando os valores padrão da implementação do `scikit-learn`. O fluxo é similar:

1) Dividir os dados (para cada proporção de teste).
2) Escalonar as características.
3) Instanciar `SGDClassifier()`.
4) Ajustar nos dados de treino e predizer o conjunto de teste.
5) Calcular a acurácia final.

Os resultados de cada experimento também são armazenados no mesmo objeto JSON que contém os resultados do KNN, utilizando chaves identificadoras como `"Linear 0.3"` para indicar que a acurácia corresponde ao classificador linear com 30% de teste.

III. Resultados e Discussão

Para avaliar o desempenho, coletamos as acurácias geradas pelos experimentos KNN e pelo classificador linear. As Figuras 1, 2 e 3 apresentam comparações gráficas (curvas de acurácia) para diferentes valores de K sobre as divisões de treino de 30%, 40%, 50%, 60% e 70%.

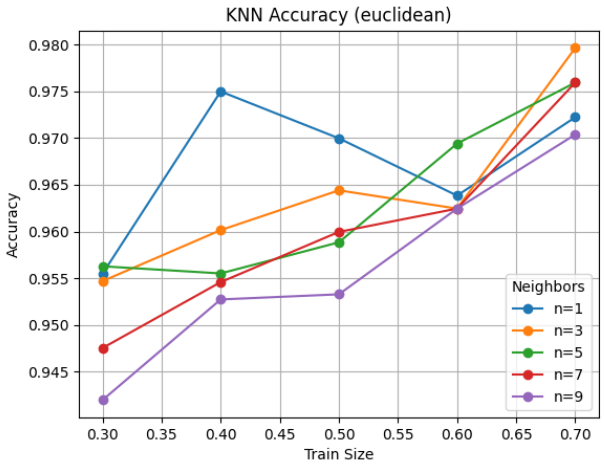

Figura 1: Desempenho KNN usando métrica Euclidiana.

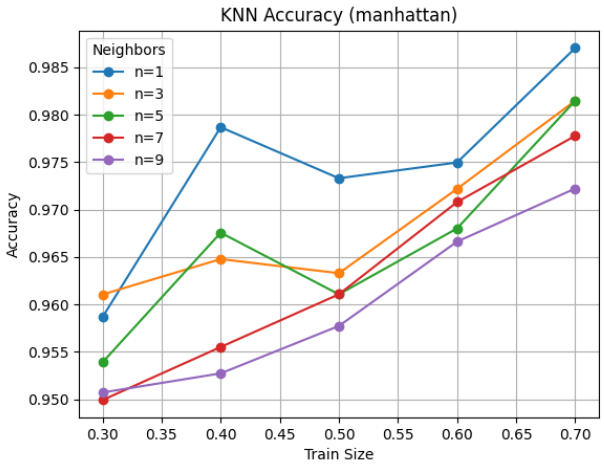

Figura 2: Desempenho KNN usando métrica Manhattan.

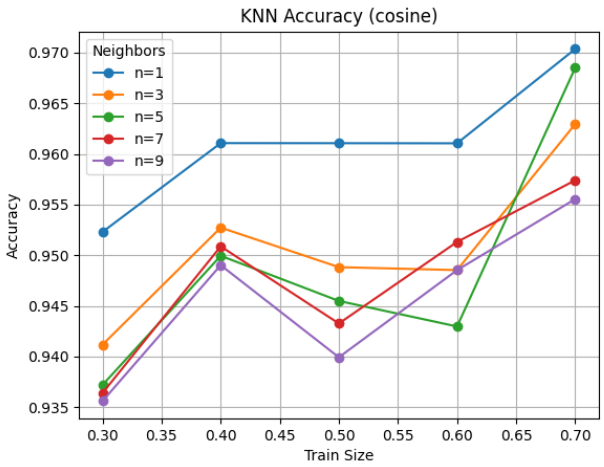

Figura 3: Desempenho KNN usando métrica Cosseno.

A. *Análise das Figuras*

- Na métrica **Euclidiana** (Figura 1), observa-se que, no geral, quanto maior o *train_size*, maior a acurácia. A maior acurácia (98%) foi obtida com $n = 3$
- Na métrica **Manhattan** (Figura 2), as curvas são muito semelhantes às da métrica Euclidiana, porém, o resultado usando $n = 1$ ultrapassou 98,5% de acurácia.
- Na métrica **Cosseno** (Figura 3), nota-se que as acurácias são levemente inferiores às obtidas com as outras métricas. A melhor acurácia (cerca de 97%) foi obtida com $n = 1$.

B. *Desempenho do Classificador Linear*

A Figura 4 mostra a acurácia do classificador linear (SGD) em função do tamanho do conjunto de teste.

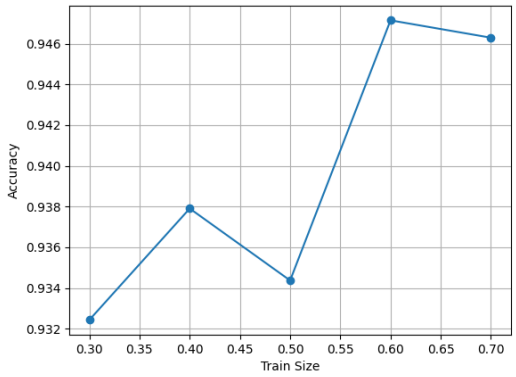

Figura 4: Acurácia do classificador linear (SGD) para diferentes proporções de teste.

Observa-se que conforme o *train_size* cresce, a acurácia cresce. Esse é um comportamento esperado, já que o modelo recebe mais dados para atualizar seus pesos.

Quando comparamos o melhor resultado de KNN (métrica Manhattan, $K = 1$, 70% de treino: acurácia $0,987$) com o do classificador linear (60% de treino: acurácia $0,948$), nota-se que KNN apresenta desempenho significativamente superior.

IV. Conclusão

Neste trabalho, implementamos e avaliamos o classificador KNN variando métricas de distância (*euclidiana*, *manhattan* e *cosseno*), número de vizinhos e proporções de divisão treino/teste. Observou-se que:

- **Euclidiana** e **Manhattan** são as métricas que proporcionam melhor acurácia;
- **Cosseno** apresentou desempenho inferior.
- O classificador linear (SGD) ficou atrás do KNN em todos os cenários avaliados.