Privacidade de Dados – 2025

Trabalho 6 – Privacidade Diferencial – Mecanismo Exponencial

Iago Chaves, Javam Machado

1 Objetivo

O trabalho consiste em implementar um algoritmo aleatório, também chamado mecanismo, que introduz privacidade diferencial em consultas realizadas sobre bases de dados. Os alunos devem incluir esses mecanismos no modelo kNN (*k-Nearest Neighbors*), a fim de comparar as respostas geradas pelo classificador sem a introdução de ruído e as respostas geradas pelo kNN diferencialmente privado.

2 Especificação

Considere o conjunto de dados *Adult Income Dataset*. Você deve recuperá-lo por meio do link:

`https://www.kaggle.com/datasets/wenruliu/adult-income-dataset/`

Este dataset contém, originalmente, 48842 tuplas e 15 atributos. O pré-processamento que deve ser realizado sobre essa base está descrito abaixo.

- Remova os atributos *fnlwgt*, *education*, *capital-gain*, *capital-loss* e *hours-per-week*.
- Os valores faltantes estão representados por "?". Remova as tuplas que contêm este valor.
- Codifique cada coluna para que os valores assumidos sejam possíveis de serem utilizados pela função de distância. Exemplo: ao codificar a coluna *gender* para inteiro, o valor *Male* assumirá o valor 0 e *Female*, valor 1 (ou o contrário).

Após a etapa de pré-processamento, a base de dados deve ser dividida em 70% treino (X_{train}) e 30% teste (X_{test}). Em seguida, o aluno deverá treinar o algoritmo r-N tradicional do material suplementar, e usando a distância euclideana para computar as vizinhanças, **não podendo fazer uso de bibliotecas (e.g., scikit-learn)**. A partir do modelo treinado, a **predição deve ser feita sobre os dados de teste** e o resultado deverá ser armazenado em um arquivo. Em seguida, será necessário implementar o mecanismo Exponencial, que será acrescentado ao kNN para gerar uma predição privada. Portanto, o aluno deverá implementar o r-N tradicional e o r-N diferencialmente privado usando mecanismo Exponencial. Os algoritmos devem ser executados para os seguintes valores de $\varepsilon \in [0.1, 0.5, 1, 5, 10, 100]$. Todos os resultados devem ser armazenados em arquivos separados.

Por simplicidade, a inclusão dos algoritmos de privacidade sobre o classificador kNN pode ser feita baseando-se no Algoritmo 2 de "Gursoy, Mehmet Emre, et al. Differentially private nearest neighbor classification. Data Mining and Knowledge Discovery 31 (2017): 1544-1575.".

Uma vez construído o modelo, podemos classificar os dados de teste satisfazendo à privacidade diferencial. A Figura 1 mostra um exemplo de como funciona esse processo. No exemplo, os dados foram divididos em duas classes: preto e branco. Para determinar a classe de um dado x do conjunto de teste, contamos quantos pontos de cada classe estão dentro do raio do ponto x, o raio limita a distância máxima entre os pontos. Para este trabalho, **varie os valores do raio** $r \in \{1, 3, 6, 9\}$. Lembre-se do teorema da composição sequencial para divisão do orçamento de privacidade. **Leve em consideração o tamanho do conjunto de dados de teste. A utilização do mecanismo Exponencial muda o uso de orçamento em relação ao mecanismo de Laplace utilizado no trabalho anterior?**

3 Requisitos

- Linguagens: C++ ou Python
- Duplas: as mesmas do trabalho anterior

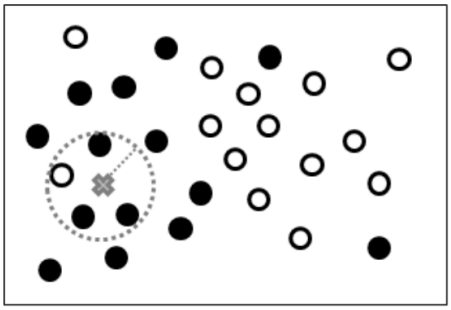

Figure 1: x é um ponto que desejamos classificar. Dado um raio r, contamos quantos itens de cada classe estão dentro desse raio. O ruído deverá ser acrescido a essas contagens para satisfazer a Privacidade Diferencial.

- **Preparar uma apresentação obrigatória que mostrará**:
  1. Qual função de utilidade utilizada;
  2. Qual sensibilidade global dessa função de utilidade;
  3. Como aplicou o mecanismo Exponencial;
  4. Qual valor de orçamento utilizado cada vez que utilizou o mecanismo Exponencial e o porquê;
  5. Qual a diferença do uso do orçamento do **trabalho 5 vs. trabalho 6** e porquê;
  6. Gráfico de acurácia por cada valor de ε (colocar, também, como referência, a linha representando a acurácia do r-N sem privacidade), adicionar linhas para cada valor de r.
- Preparar uma Demo para explicar, mostrar o seu programa e os resultados durante a aula de entrega. Escreva um Readme.txt descrevendo o projeto.
- Zipar o seu projeto (código fonte e executável), os resultados das predições, os gráficos e o Readme.txt em um único pacote e submeter via **Classroom**.
- O trabalho deverá ser entregue até as 10h da **quarta-feira, dia 07/01/2026**, e explicado durante a aula do mesmo dia.

Em caso de dúvidas, utilize o Google Classroom para solicitar esclarecimentos.

4 Avaliação

Na avaliação serão considerados os seguintes indicadores:

- **Corretude** do programa;
- **Geração dos arquivos** com as predições geradas pelo kNN original e ε-diferencialmente privado;
- **Acurácia** pela comparação da predição original e predições diferencialmente privadas;
- Clareza na **explicação** do programa durante a Demo;
- **Arguição:** qualidade da apresentação, bem como assertividade sobre os questionamentos, demonstrando real domínio do aluno sobre o trabalho realizado;
- **Pontualidade** e **documentação/qualidade** do código-fonte.