Fundamentos de Data Stream Mining
Prof. Fabrício Enembreck

Desafio 1 – Comparando estratégias de Detecção de Mudanças
Equipe: até 4 membros

O objetivo desse desafio consiste em familiarizar os alunos com o framework MOA e os conceitos de detectores de mudança.

Parte I – Geração de streams sintéticas

Essa primeira parte da atividade consiste em produzir os datasets utilizados nos experimentos da próxima parte da atividade. Como os elementos avaliados são detectores, é necessário gerar streams sintéticas onde o momento, o tipo e a quantidade de drifts são conhecidos. Para isso o seguinte processo deve ser realizado:

1) No menu "Other Tasks" escolha a opção WriteStreamToARFFFile
2) Configure o parâmetro arfffile escolhendo um diretório e nomeando o arquivo como streamBinaryAbruptDrift
3) Maxinstances deve ser 3000
4) Configure o parâmetro stream para ConceptDriftStream e insira nessa stream segmentos onde os drifts ocorrem aproximadamente a cada 500 instâncias e são do tipo cd.AbruptChangeGenerator com diferentes seeds
5) Execute a task. Isso deve criar o arquivo arff de stream com drifts abruptos.
6) Repita os processos de 1 a 5 alterando o nome da stream para streamNotBinaryAbruptDrift. Antes de rodar a task, marque o campo notBinaryStream para a classe cd.AbruptChangeGenerator em todos os seguimentos de stream que foi utilizado.

Parte II – Geração de resultados e comparação entre detectores

Na parte I do projeto foram geradas duas streams com drifts abruptos. Na segunda parte da atividade a performance dos detectores deverá ser comparada. Para isso, faça o seguinte procedimento:

1) No menu "Concept Drift", configure um experimento da seguinte forma:
2) Selecione a task "EvaluateConceptDrift"
3) Selecione um learner. Vc deverá repetir os experimentos para pelo menos 3 learners. Sugestão: use os learners ADWIN, DDM e CUSUM, um de cada vez
4) stream: selecione um dos arquivos que vc gerou na etapa I
5) evaluator: selecione "BasicConceptDriftPerformanceEvaluator"
6) instancelimit: coloque o tamanho da stream
7) sampleFrequency: esse parâmetro não altera os resultados finais, mas auxilia a visualizar o comportamento dos detectores ao longo da stream. Coloque o valor 1, para vc visualizar o resultado a cada instância da stream. Depois de rodar os experimentos ajuste o eixo X do gráfico para facilitar a visualização.
8) Rode o experimento e extraia as seguintes métricas:

   Precisão = detectedchanges / truechanges

Delay = delaydetection(average)

FalsosPositivos = max(0, detectedchangesObservedInStream - truechanges)

Os valores das métricas *truechanges*, *detectedchanges* e *delaydetection* podem ser visualizados na última linha do grid. A métrica detectedchangesObservedInStream deve ser obtida com a observação da quantidade de drifts identificados ao longo da stream representados pelas linhas verticais no gráfico.

9) Execute os itens 1 a 8 alterando o detector (ADWIN, CUSUM, DDM, outros) e coloque os resultados em uma tabela
10) Analise os resultados comparando as métricas obtidas.
11) Tente "tunar" os detectores utilizados, ajustando os parâmetros deles e avaliando se isso melhora as métricas.
12) Repita os passos 1 a 12 para as 2 streams que vc gerou na Parte I. Resultados e análises devem ser feitas para cada stream individualmente. Conclua as análises verificando se os resultados foram diferentes para as duas streams.
13) Gere um relatório em formato pdf de até 4 páginas onde vc apresenta o protocolo usado para geração das streams, métricas, resultados obtidos e análises.