Universidade Federal de Rio Grande do Norte
Departamento de Informática e Matemática Aplicada
Disciplina: Aprendizado de Máquina
Professora: Anne Magály de Paula Canuto (anne@dimap.ufrn.br)

Trabalho Prático do Final da Disciplina – Parte 2

1. Objetivos:

O principal objetivo desta parte do trabalho prático é analisar como os métodos e técnicas vistos em sala de aula (aprendizado supervisionado e não-supervisionado) se comportam em uma aplicação prática. Para tal, o aluno terá os seguintes objetivos

I. **Aprendizado supervisionado:** Comparar os resultados obtidos com os algoritmos supervisionados vistos em sala de aula (redes neurais, *k*-nn, árvores de decisão e *naive bayesian learning*) quando aplicados as duas bases de dados escolhidas no trabalho I (pré-processamento dos dados). Para a realização deste objetivo, alguns objetivos específicos são desejados, que são o seguintes:
   - Verificar o impacto do parâmetro *k* no k-NN além de analisar o impacto do uso ou não do processo de escalonamento dos valores dos atributos numéricos para [0,1] e do uso de *distanceweight* no erro de classificação do *k*-nn.
   - Verificar como o erro de classificação e o tamanho da árvore de decisão dependem do parâmetro de poda (poda ou sem poda).
   - Verificar o impacto da segunda suposição do naive bayes (os dados obedecem a uma distribuição normal) no erro de classificação deste classificador.
   - Verificar como o erro de classificação depende do tamanho da rede neural, como também da taxa de aprendizado e do número de iterações.

II. **Aprendizado não supervisionado:** Analisar os dois principais métodos não supervisionados vistos em sala de aula (k-means e hierárquico) e tentar definir o melhor número de grupos para o problema em questão, baseado no resultados desses algoritmos.

2. Metodologia do trabalho

Como já sabemos, as três principais etapas do processo de aprendizado de máquina são as seguintes: análise dos dados, métodos utilizado e pós-processamento. Na etapa de métodos utilizados, que é o objeto de estudo deste trabalho, serão analisados os algoritmos supervisionados e não-supervisionados vistos em sala de aula.

Este trabalho prático poderá ser feito utilizando qualquer linguagem de programação. Além disso, os resultados obtidos devem ser interpretados e apresentados em forma de relatório (conforme Anexo A). Ao final, o aluno deve preparar um relatório, dedicando bastante tempo à interpretação dos resultados obtidos. Cada aluno deve decidir como serão ilustrados/discutidos os resultados obtidos.

2.1. Métodos de Aprendizado Supervisionados

Como já mencionado, quatro métodos de aprendizado de máquina devem ser utilizados neste trabalho, que são: redes neurais (Multi-Layer Perceptron), k-nn, árvores de decisão e um método de aprendizagem bayesiana (Naive Bayesian). Para cada método supervisionado, é preciso responder detalhadamente a seguinte pergunta:

- Qual foi o melhor conjunto de parâmetros para a base de dados escolhida, por que? Em outras palavras, existe alguma análise, a nível de parâmetros e/ou dados utilizados, que possa justificar e/ou explicar o comportamento do método em questão?

Aqui é extremamente importante não se deter **APENAS** nos resultados e dizer que, por ter dado o menor erro, o conjunto de parâmetro X foi o melhor. Mas sim, o aluno deve tentar entender e justificar o comportamento do método para a base de dados escolhida. Nas próximas subseções, descreve-se melhor os métodos individualmente

2.1.1. ***k*-NN:** Serão feitos experimentos em dois cenários: uso ou não do escalonamento dos valores dos atributos numéricos para [0,1]. Em cada um desses contextos, o procedimento deve ser o seguinte. Serão feitos treinamentos usando 10-*fold cross validation* para diferentes valores de *k* (devem ser escolhidos, pelo menos, três valores diferentes de *k*). Aqui, o aluno deve buscar o valor de k que produza o melhor resultado para esta base de dados.

Fazer os experimentos com e sem peso (utilizado para desempate, caso haja empate). Após realizar os experimentos com este método, o aluno deve buscar respostas para as seguintes perguntas:

- Qual foi o melhor valor de k?
- Foi importante escalonar os valores?
- E o peso teve algum impacto no desempenho do método?

2.1.2. **Árvores de decisão:** Serão feitos treinamentos usando 10-*fold cross validation*, variando-se apenas o parâmetro poda (ou seja, construir árvore com poda e sem poda). Aqui, as perguntas a serem respondidas são as seguintes:

- A AD tinha overfitting antes da poda?
- Com o uso da poda é possível afirmar se a AD estava ou não com overfitting?

2.1.3. ***Naive bayesian learning*:** Para este método, eu gostaria que vcs respondam as seguintes perguntas:

- Os meus dados numéricos estavam obedecendo uma distribuição normal?
- Os meus dados possuem alguma relação?.

2.1.4. **Redes Neurais:** Serão feitos treinamentos de redes MLP usando o *backpropagation* padrão (com o t*ermo momentum* fixo de 0.8), variando-se os seguintes parâmetros:

- Quantidade máxima de iterações (ou ciclos).
- Quantidade de neurônios intermediários (ou escondidos) da rede --- serão usadas redes com apenas uma camada intermediária.
- Taxa de aprendizado.

Para cada um destes três parâmetros, devem ser usados três valores. Sendo assim, o aluno deve escolher três valores diferentes para: quantidades máxima de iterações,

quantidades de neurônios escondidos e taxa de aprendizado. Exemplo: poderiam ser usados 100, 1.000 e 10.000 iterações; 4, 8 e 12 neurônios escondidos; e taxas de aprendizado de 0.1, 0.01 e 0.001. Isto é apenas um exemplo.

O próprio aluno é quem vai escolher os valores a serem usados, dependendo do problema de classificação a ser resolvido. Vale a pena lembrar que não existem valores ideais que podem ser usados para qualquer problema, de modo que uma taxa de aprendizado de, por exemplo, 0.01 pode ser "pequena demais" para um determinado problema, sendo "grande demais" para outro. O mesmo é verdade para a quantidade de iterações e de nodos escondidos. Porém, para a taxa de aprendizado, há uma particularidade que o aluno deve obedecer. O aluno não precisa escolher exatamente os valores supracitados, mas é necessário que o aluno escolha valores com granularidades diferentes. Portanto, o aluno deve escolher os seguintes valores 0,V; 0,0V e 0,00V, onde V pode ser qualquer valor inteiro no intervalo [1,9]. Para os outros dois parâmetros, é importantes que o aluno não escolha valores muitos próximos (por exemplo: 5, 7 e 9 neurônios na camada escondida), para que tenhamos um quadro mais genérico do comportamento da rede quando variando estes parâmetros.

Uma sugestão para o número de neurônios seria: usar o valor (num. Att + num. classes)/2 como ponto de partida. Para o numero de iterações, a minha sugestão seria 100, 1.000 e 10.000, como sugerida acima. Porém, se o simulador weka demorar muito com 10.000 iterações, o aluno pode diminuir para 5.000 iterações ou algum valor menor.

Para a execução do treinamento, para cada combinação dos três parâmetros supracitados, deve-se usar um método *2-fold-cross-validation*, sendo anotados os respectivos resultados. Portanto, devem ser executados 3 x 3 x 3 = 27 treinamentos preliminares.

Uma vez feita as 27 execuções, deve ser escolhida a melhor rede obtida. Em outras palavras, o aluno irá escolher a melhor combinação de valores para estes três parâmetros. Define-se como "melhor rede" a rede que proporcionou os melhores resultados, sendo o mais compacta possível. Uma boa escolha pode ser baseada no erro (ou precisão) do conjunto de treinamento e no tamanho da rede. Exemplo: se o menor erro de treinamento foi obtido usando 1.000 iterações, oito nodos escondidos e taxa de aprendizado de 0.001, então este pode ser considerado o melhor conjunto de parâmetros (ou seja, a melhor rede). Caso haja um empate nas precisões de mais de uma rede, dê prioridade as redes mais simples.

IMPORTANTE: para um bom projeto de redes neurais, não é suficiente fazer apenas uma execução para cada conjunto de parâmetros, pois sabemos que, variando-se a inicialização dos valores dos pesos, podemos obter resultados finais diferentes. Dessa forma, seriam necessárias várias inicializações de valores dos pesos para cada conjunto de parâmetros para podermos escolher de forma mais adequada a melhor rede. Entretanto, o objetivo nesta prática é de apenas propiciar uma noção de como experimentos com redes neurais são feitos. É por isso que só exigido um treinamento para cada um dos 27 configurações de parâmetros.

Para o conjunto de parâmetros escolhido (melhor rede), deve ser aplicado o *10-fold cross validation*. Neste caso, fazer cinco execuções com diferentes inicializações de pesos (ou seja, fazer cinco treinamentos, cada um partindo de uma inicialização de pesos distinta). É conveniente organizar estes resultados em uma tabela e verificar os valores de média e desvio padrão para os resultados.

Aqui, o aluno deve tentar responder as seguintes perguntas:

- Qual intervalo de valores a rede estava com underfitting ou com overfitting?
- O que acontece quando eu fixo os dois parâmetros e vario a taxa de aprendizado?
- E se o aluno fixar dois parâmetros e variar o numero de neurônios na camada escondida, o que acontece? A mesma pergunta para o numero de iterações?

2.2. Comparação entre os métodos supervisionados

Pegue o melhor resultado de cada método utilizado e faça uma análise comparativa do desempenho de tais métodos. Nesta análise comparativa, o aluno deve tentar responder de forma detalhada as seguintes perguntas:

- Qual foi o melhor método para o problema que você escolheu?
- Para esta análise comparativa, é preciso olhar a precisão, o desvio padrão e, em alguns casos, a matriz de confusão.
- Tem alguma explicação para a escolha deste método? As vezes, é necessário analisar os dados para se chegar a essa explicação. Portanto, analisem direito o comportamento dos métodos investigados e os dados para que se tente chegar a essa explicação.

2.3. Métodos de aprendizado não supervisionados

Como já mencionado, dois métodos de aprendizado não supervisionados de máquina devem ser utilizados neste trabalho, que são: k-médias e hierárquico *aglomerativo.* Para a utilização destes algoritmos na base de dados que foi escolhida, é preciso retirar o atributo classe da base de dados e, então, o mesmo está apto para utilização.

2.3.1. **k-means:** Serão feitos experimentos com k variando no intervalo definido para cada base de dados, sempre deixando o número de classes da sua base no meio do intervalo. Para uma base com 6 classes, por exemplo, variamos o k de 2 até 12. Para cada valor de k, serão feitas 5 execuções. Após os experimentos, serão calculados os índices DB de todos os agrupamentos construídos. Uma vez calculado os índices DBs, a média por tamanho é calculada e coloque os resultados em um gráfico, onde o eixo x representa o valor de k e o eixo y representa o índice DB. Defina como o número de grupos mais adequado o que tiver o MENOR índice DB.

2.3.2. *5.2. Hierárquico aglomerativo*: Assim como no k-médias, serão feitos experimentos com o numero de grupos variando e a variação vai depender do número de classes do seu problema. Como este é um algoritmo determinístico, é necessário apenas uma execução por tamanho. Para cada tamanho, calcule o índice, crie o mesmo gráfico da subseção anterior e defina qual o melhor numero de grupos

Nesta fase, devemos responder as seguintes perguntas

- Qual foi o número de grupos definido pelo k-means?
- E pelo hierárquico?
- Foi o mesmo resultado, se não, Por que será?
- Qual o melhor resultado, k-means ou hierárquico?

Para responder esta ultima pergunta, é preciso selecionar a melhor partição de cada algoritmo (k-means e hierárquico). Uma vez escolhida a melhor partição, aplicar o índice CR entre a melhor partição e a partição original (base original com a inclusão do atributo classe). O algoritmo que possuir um CR mais próximo de 1, em relação a base original, é considerado aquele que conseguiu uma partição mais parecida com a original e é então considerado o que possui o melhor resultado. Novamente, a implementação do índice CR será de total e completa responsabilidade do aluno.

Anexo A
-
Modelo de Monografia para o trabalho

1. **Seção 1--- Introdução**

Qual é o problema que você está trabalhando? Por que ele é relevante? O que poderia ser feito se o problema fosse resolvido? Em que contexto e quão freqüentemente o problema aparece? Que dados estão disponíveis? Quais outras informações (conhecimento de domínio, exemplos resolvidos etc.) estão disponíveis?

2. **Seção 2 ---- Descrição do problema e Base de Dados**
    - Descrição breve da base de dados, só para lembrar o problema

3. **Seção 3 ---- Experimentos** (Explicar o comportamento dos parâmetros estudados, dificuldades encontradas, limitações do modelo).

Primeiro, a equipe deve explicar que métodos computacionais você usará para abordar o problema? Seja específico. Por que você acredita que os métodos propostos resolverão o problema? Você está comparando métodos múltiplos? Se sim, por que estes métodos (e não outros)? Há algum limite à aplicabilidade do método? É viável, do ponto de vista prático, implementar o método e aplicá-lo aos dados disponíveis? Os dados disponíveis são adequados?

4. **Seção 5 ---- Sugestões para melhoria dos experimentos**

5. **Seção 6 ---- Referências bibliográficas**