Segunda Lista de Exercícios da Disciplina "Aprendizagem de Máquina"

André da Motta Salles Barreto
amsb@lncc.br

Data de entrega: 14/12/2009

Nessa segunda lista de exercícios o aluno irá lidar com dois problemas reais, sendo um de regressão e outro de classificação (aqui o termo "real" indica que os dados foram de fato extraídos do domínio da aplicação, e não criados artificialmente). Como na primeira lista de exercícios, o objetivo é prever o valor da variável de saída em um conjunto de teste. Segue uma breve descrição de cada problema:

Corretagem de imóveis: O objetivo nesse problema de regressão é estabelecer uma relação entre as características dos bairros da cidade de Boston e o preço médio de suas casas. Os atributos de entrada representam diversas características do bairro, como por exemplo: taxa de criminalidade, nível de poluição e facilidade de acesso a centros comerciais. A variável de saída é o preço médio de uma casa em milhares de dólares americanos. Os dados referentes a esse problema estão contidos nos arquivos cujos nomes começam com a palavra "`casa`" (os nomes dos arquivos seguem a mesma lei de formação adotada na primeira lista de exercícios).

Diagnóstico de câncer de mama: Nesse problema o objetivo é classificar um tumor de mama como sendo maligno ou benigno a partir de informações colhidas a respeito de suas células. Os atributos de entrada representam informações sobre as células, como uniformidade de tamanho e forma, espessura de membrana e quantidade de adesão marginal, entre outras. A variável de saída representa uma das duas classes do problema: $y = -1$ indica que a célula é saudável e $y = 1$ acusa uma célula cancerosa. Os dados referentes a esse problema estão contidos nos arquivos com nomes iniciados com a palavra "`diagnostico`".

O aluno deverá usar uma rede neural para resolver o primeiro problema e uma máquina de vetor de suporte (SVM) para resolver o segundo. A rede neural deve conter pelo menos uma camada oculta e deve ser treinada pelo algoritmo de retropropagação do erro (*backpropagation*). A versão do SVM adotada deve ser aquela em que são permitidas classificações incorretas (através do uso das variáveis ξ_i). Além disso, o SVM deve usar um *kernel* não-linear. O método de treinamento do SVM deve ser preferencialmente o algoritmo SMO, embora um pacote de programação quadrática também possa ser usado para resolver o problema. O aluno é responsável por implementar os algoritmos em ambos os casos (ou seja, o uso de código de terceiros *não* é permitido). No caso de o aluno optar por treinar o SVM usando um pacote de programação quadrática, este último deve ser adotado *apenas* para o cálculo das variáveis α_i; todo o resto deve ser programado pelo aluno. Para cada um dos dois problemas, os seguintes passos devem ser executados:

1. **Descrição do modelo:** a primeira providência é definir as características básicas do modelo, bem como do algoritmo de treinamento adotado. No caso da rede neural, isso corresponde a determinar a função de ativação das unidades ocultas e um valor adequado para a taxa

de aprendizagem. No caso do SVM essa primeira etapa se reduz à seleção de um *kernel* não-linear e de um método para solucionar o problema de programação quadrática.

2. **Seleção de modelo:** nessa etapa o aluno deve usar a validação cruzada (simples ou em k blocos) para escolher o modelo que melhor resolva o problema. No caso da rede neural, isso significa determinar o número de unidades na(s) camada(s) oculta(s). No caso do SVM, a seleção de modelo corresponde à definição de um valor adequado para a variável "C" e para os parâmetros do *kernel* escolhido (por exemplo: caso o *kernel* gaussiano seja adotado, deve-se determinar um valor para o desvio padrão σ; caso o aluno opte por um *kernel* polinomial, é preciso definir o grau do polinômio). O aluno deve reportar a estimativa do erro de generalização para *cada* configuração testada e apontar aquela escolhida como sendo a melhor.

3. **Seleção de atributos**: uma vez selecionado o modelo, o próximo passo é selecionar os atributos mais relevantes para a solução do problema. O aluno deve usar a seleção à frente ou a seleção baseada em filtro para determinar o subconjunto dos atributos que resulte na melhor estimativa de generalização. Essa etapa deve ser realizada com o modelo selecionado no passo anterior. O aluno deve reportar a estimativa do erro de generalização para *cada* subconjunto de atributos testado e apontar aquele escolhido para ser usado no modelo final.

4. **Combinando modelos:** nessa etapa do exercício o aluno já deve contar com a definição completa do modelo. O objetivo agora é formar um comitê de modelos a fim de aumentar o poder final de generalização. Para tal, pode-se adotar tanto o *bagging* quanto o *boosting* (em ambos os casos, o comitê deve ser formado por pelo menos 5 modelos diferentes). Evidentemente, os membros do comitê devem ser configurados de acordo com as decisões tomadas nos passos anteriores.

O aluno deve escrever um relatório descrevendo em detalhes a execução dos passos acima. Embora o relatório possa ser conciso, é preciso deixar claras as decisões tomadas em cada passo do exercício (qual o algoritmo utilizado, como ele foi configurado, *etc.*). Além disso, o aluno deve enviar por *e-mail* os resultados obtidos nos conjuntos de teste. Isso deve ser feito tanto para o modelo encontrado ao final da etapa 3 quanto para o comitê configurado no passo 4. Os dados devem estar organizados da mesma maneira que os arquivos da lista, com um valor de y em cada linha (e *nada* além disso). No caso do problema de classificação, o valor de y deve ser -1 ou 1. Os nomes dos arquivos devem seguir a seguinte lei de formação:

Sobrenome do aluno	+	**Problema**	+	**Modelo**
		`casas` / `diagnostico`		`simples` / `comite`

Por exemplo, as minhas respostas para o problema de regressão utilizando um comitê de redes neurais deveriam estar contidas em um arquivo denominado `barreto_casas_simples.dat` (como o cálculo do erro é feito de maneira automática por um programa de computador, os nomes dos arquivos devem ser *exatamente* como mostrado acima). Finalmente, o aluno deve enviar o código-fonte dos algoritmos utilizados em um *único* arquivo. O código deve estar comentado e conter apenas o estritamente necessário para a resolução do exercício acima.