Trabalho Prático 5

SISTEMAS NEURO-DIFUSOS PARA A MODELIZAÇÃO DE PROCESSOS

Computação Adaptativa (MEI)
CNSD (MIEB)

Marco Simões (MEI) - 2006125287
Sérgio Santos (MEI) - 2006125508

Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra
Dezembro, 2009

Introdução

Os sistemas neuro-difusos são um boa opção quando se pretende modelizar um determinado processo. Quando não existe conhecimento teórico ou empírico sobre um sistema em estudo, torna-se complexa a tarefa de aferir o seu comportamento. Os sistemas neuro-difusos possibilitam a geração de regras apenas com base em dados experimentais, recolhidos através da interacção com esse mesmo sistema. Adicionalmente, as regras geradas poderão ser usadas posteriormente para a derivação de novo conhecimento sobre o processo.

O objectivo deste trabalho consiste então em modelizar um processo, através de dados experimentais de entrada e saída fornecidos. Ao mesmo tempo, pretende-se comparar duas técnicas de agrupamento dos dados de entrada (*subtractive clustering* e *fuzzy c-means*), em relação à precisão dos modelos obtidos. Assumindo que os estamos a trabalhar sobre um sistema dinâmico com memória, é necessário adicionar como entradas do modelo valores antecedentes, segundo a equação:

$$y(k) = f(\, y(k-1), y(k-2), u(k-1), u(k-2)\,)$$

É sugerida a utilização do ANFIS (Adaptive Neuro-Fuzzy Inference Systems) para a geração das regras associadas aos agrupamentos calculados.

Desenvolvimento

O desenvolvimento passou por 3 fases distintas:

- Criação do controlador difuso manualmente e testá-lo com os dados fornecidos;
- Utilização de aprendizagem sobre o controlador manual, aplicando o ANFIS e os dados de treino fornecidos;
- Criação de um controlador automaticamente e treiná-lo e testá-lo com os dados fornecidos.

Estas fases fizeram-nos suplantar diferentes etapas, entre elas:

Carregamento dos dados

O carregamento dos dados foi feito a partir do ficheiro fornecido com o projecto. Para tal, basta-nos fazer um *xslread* do ficheiro e atribuímos os dados para variáveis no *workspace*.

Tratamento dos dados (normalização)

Para que pudéssemos utilizar de forma mais controlada os dados na geração manual do controlador, precisámos normalizar os dados para que os centróides não fugissem

da gama [-1 1]. Assim, garantimos que os nossos centróides estão na gama que queremos e podemos definir os limites do nosso controlador em [-1 1].

Geração de matriz de dados

A nossa matriz de dados consiste em 5 colunas, sendo a primeira a saída com um atraso de uma unidade, a segunda equivalente mas com um atraso de duas unidade, a terceira igual à entrada mas com um atraso de uma unidade, a quarta equivalente mas com um atraso de duas unidade e, finalmente, a última coluna corresponde à saída no instante corrente.

Divisão dos dados (60/30/10)

Os dados de entrada têm que ser divididos para que possamos usar diferentes valores para treino e teste. Assim, sorteamos a matriz para que as divisões não sejam sequenciais e cortamo-la em 3 partes, correspondendo a 60% para dados de treino, 30% para dados de teste e 10% para *check*, necessários para garantir a inexistência de *overfitting*.

Aplicação de técnicas de clustering

Para fazer *clustering* dos dados (operação necessária para a criação do controlador neuro-difuso) utilizámos dois métodos independentes – *Subtractive Clustering* e *Fuzzy C-Means*.

Criação de um controlador manualmente

Feito o *clustering* pudemos usar os centróides resultantes para gerar um controlador e as respectivas regras. Utilizámos a função de pertença *gbellmf* por ser utilizada nos métodos automáticos descritos mais à frente, garantindo-nos assim uma análise mais justa aquando da comparação de este controlador com os restantes.

Geração de Controladores por Clustering

Através dos métodos genfis2 e genfis3 gerámos os controladores automáticos que usam métodos de *clustering subtractive* e *fuzzy c-means*, respectivamente.

Teste e aprendizagem de todas as soluções

Construídos todos os controladores, utilizamos a função de aprendizagem ANFIS para evoluir os controladores, dando-lhes os dados de treino acima referidos. Para teste da solução, fazemos *evalfis* para validar os controladores gerados com os dados de treino gerados na divisão dos dados. No final, comparamos o resultado obtido com o esperado através da função *sum square error* (SSE).

Execução

Para poder experimentar o projecto desenvolvido, desenvolvemos uma interface gráfica(Figura 1), **interface.m**, que permite configurar:

- Tipo de clustering: subtractive clustering ou fuzzy c-means;
- Parâmetros específicos de um determinado tipo de clustering;
- Parâmetros de treino do modelo neuro-difuso;

Figura 1 - Interface gráfica do projecto

O botão "Run Clustering & Evaluate" despoleta o teste da configuração escolhida, criando, treinando e testando 20 modelos. Entre cada modelo variam os casos de treino, validação (para evitar *overfitting*) e teste, que são escolhidos aleatoriamente segundo a proporção 60/30/10 já mencionada. O erro médio desses 20 modelos é apresentado na caixa respectiva. Também é possível visualizar o último dos modelos gerados, através da interface *fuzzy*, pelo botão "Open Fuzzy". Para melhorar a experiência do utilizador, é ainda desenhado um gráfico que mostra o output atingido e o desejado, a azul e verde respectivamente. É de notar que com erros muito baixos a diferença não é perceptível, vizualizando-se muitas vezes apenas uma das cores, que sobrepõe a outra.

Incluímos o modelo desenvolvido manualmente, com o nome de **manualFIS.fis**.

Finalmente, desenvolvemos também um script para automatizar a execução de testes de modelos, **batchTesting.m**, embora só deva ser executado para a realização de testes detalhados como os que apresentaremos de seguida neste relatório.

Resultados

Foram realizados diversos testes por forma a avaliar a performance dos modelos gerados, especialmente comparando as técnicas de *clustering* utilizadas. Os parâmetros que variámos, por técnica de *clustering* foram:

- Subtractive clustering
  - Raio de influência;

- Squash;
- Taxa de aceitação;
- Taxa de rejeição.

- Fuzzy c-means
  - Número de clusters;
  - Número máximo de iterações;
  - Valor mínimo de melhoramento;
  - Expoente da matriz de partição.

As tabelas contendo todos os resultados obtidos ordenados por performance, encontram-se na pasta do projecto, sob o nome de **resultados.xlsx**.

Conclusões

A Figura 2 apresenta uma visualização do clustering efectuado manualmente sobre os dados de entrada. São estes clusters que são usados para a criação manual do controlador lógico.

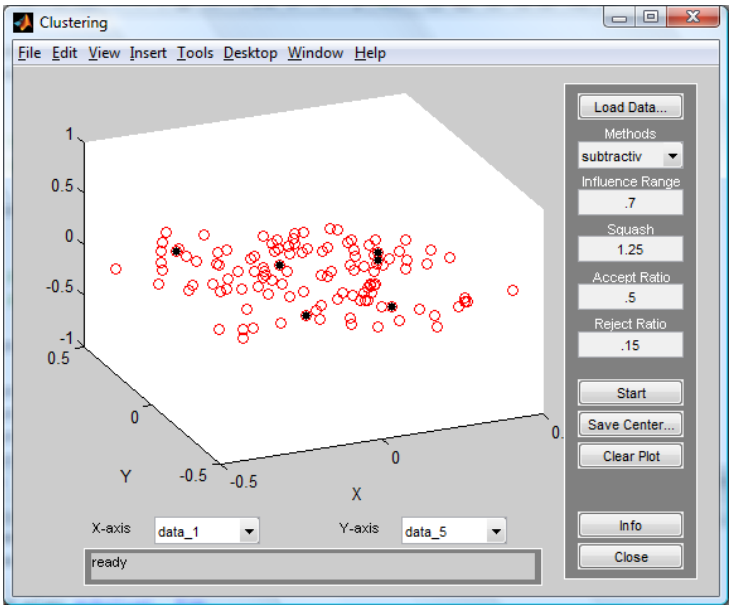


Figura 2 - Visualização dos Clusters dos dados de entrada

A execução do controlador criado manualmente obteve resultados não muito bons, como podemos visualizar na figura 3, que mostra o resultado de um teste com o controlador manual, que obteve um erro de 2.3228 (Sum Squared Error).

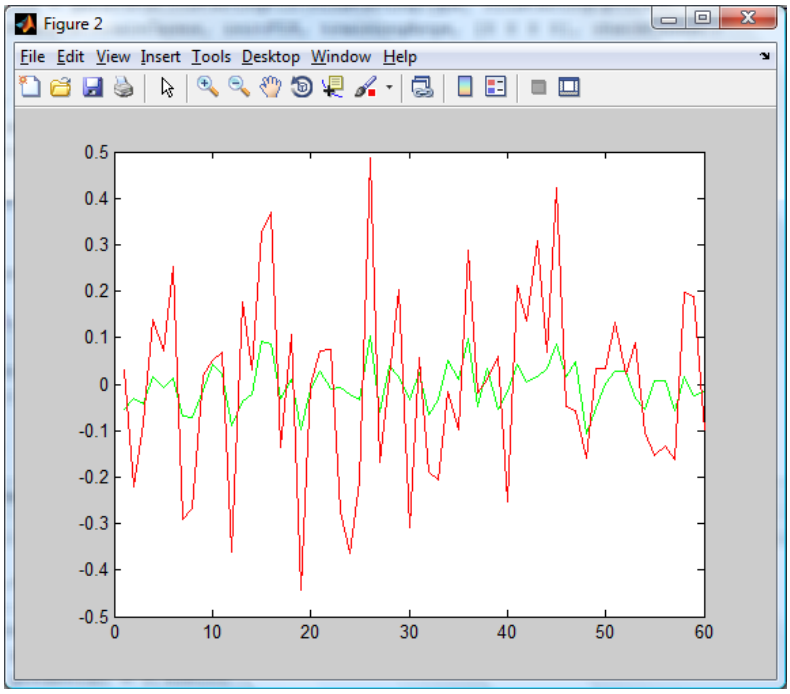


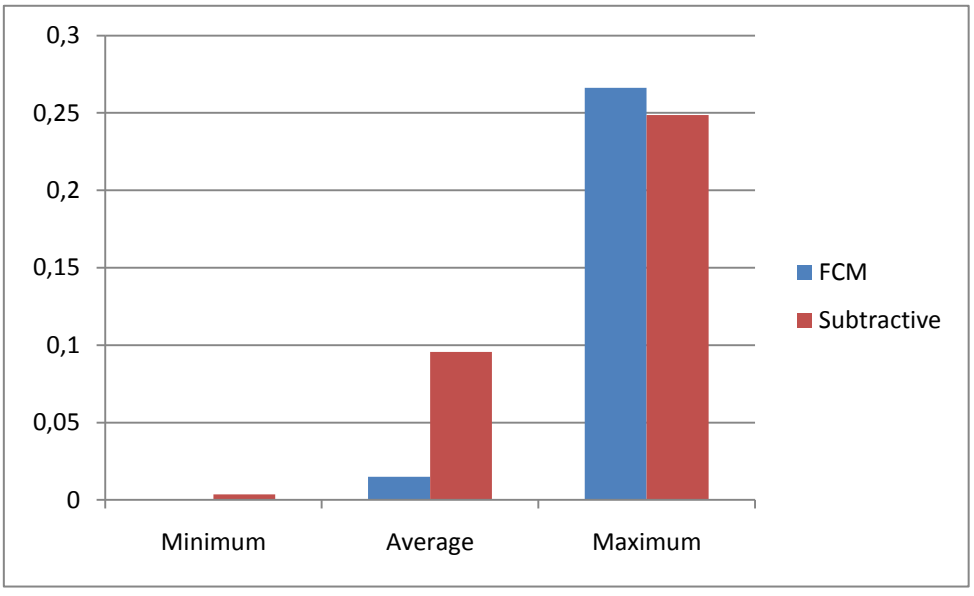
Figura 3 - Resultado obtido (verde) em contraste com o resultado esperado (vermelho) da execução do controlador manual

Nos testes restantes, onde são contrapostos os dois tipos de clustering usados, podemos ver na seguinte análise de performance que a técnica Fuzzy C-Means obtém, excepto no seu pior caso, melhores resultados que a técnica subtractiva.

Performance Comparison

	Minimum	Average	Maximum
FCM	*0,0004	0,014906141	0,266151219
Subtractive	*0,0036	0,095667183	0,248553487

*x10^-8



Verificamos no relatório acima apresentado que quer no melhor caso, quer em média, FCM demonstra-se mais eficiente na construção do controlador difuso.

Vejamos agora internamente cada técnica.

FCM:

O seu melhor caso apresenta um erro (SSE) de 4E-12, sendo a sua configuração:

clusterNum	maxIterations	minImprovement	exponent
2	300	0,01	1,2

A nível de **nº de clusters**, nota-se que o aumento de número de clusters corresponde a um agravamento do erro, tendo os melhores resultados sido obtidos com 2 clusters, e os piores com 100;

A nível do **número de iterações**, não se nota uma diferença significativa. Concluímos assim que 20 iterações parecem ser suficientes para uma aprendizagem com um sucesso significativo.

A nível do **valor mínimo de melhoramento**, a tabela abaixo mostra que o afastamento da origem causa um leve melhoramento nos valores de erro, quer mínimos, médios ou máximos.

minImprov.	Minimum	Average	Maximum
0	5E-12	0,0229531	0,266151219
0.00001	5E-12	0,0179699	0,250937832
0.1	4E-12	0,0037954	0,125435822

Tabela 1- Valores mínimos de melhoramento

Finalmente, relativamente aos valores de **expoente da matriz de partição**, verificamos que o aumento do seu valor é prejudicial ao desempenho do controlador, em que os valores de 1.2 e 2 apresentam muito melhores resultados que valores como 3 e 5.

Subtractive Clustering:

O seu melhor resultado apresenta um erro (SSE) de 3,6E-11 tendo como configurações:

influenceRange	squash	acceptRatio	rejectRatio
0,5	1,5	0,75	0,5

Relativamente ao **raio de influência**, o aumento deste é benéfico ao desempenho do controlador, tendo 0.5 muito melhores resultados que 0.15 ou mesmo 0.3.

O mesmo se verifica para o valor de **Squash**, em que os resultados para valores como 1.5 são melhores que 1.25 e, consequentemente, 1.0.

Para a **taxa de aceitação**, a tabela abaixo mostra que a nível de valor mínimo, a melhor taxa assume o valor de 0.75, mas em valor médio e máximo o valor 0.5 apresenta

melhores resultados. Conclui-se que os valores mais extremistas (0.25 e 1, neste caso) são más escolhas para esta taxa.

AcceptRatio	Minimum	Average	Maximum
0,25	1,162E-09	0,1031323	0,234774856
0,5	5,1E-11	0,0910953	0,226911508
0,75	3,6E-11	0,0933238	0,248553487
1	6E-11	0,0951173	0,239181927

Tabela 2 - Valores para Taxa de Aceitação

No que toca à **taxa de rejeição**, à excepção dos valores máximos, que pioram ligeiramente, o aumento do valor desta taxa resulta em melhoria de desempenho nos respectivos controladores difusos. A tabela abaixo mostra essas melhorias, a nível mínimo e médio, e o ligeiro agravamento a nível máximo.

RejectRatio	Minimum	Average	Maximum
0.1	8,4747E-08	0,107106	0,235787837
0,25	1,162E-09	0,0981321	0,246303862
0,5	3,6E-11	0,0817634	0,248553487

Tabela 3 - Valores para Taxa de Rejeição