Projeto Final

Disciplina Mineração de Dados em Biologia Molecular 2012

Vinicius Lucena

Objetivo

O projeto tem como foco utilizar ferramentas e conceitos de Mineração de Dados usados durante a disciplina numa forma prática combinando com conjuntos de dados biológicos retirados do conjunto de dados do UCI Machine Learning Repository. Retirado da área "Life Sciences" escolhemos os dados Ecoli, Yeast e Parkinsond para esse trabalho.

Introdução

Usamos 3 classificadores pertencentes a ferramenta WEKA a fim realizar o experimento, KNN, redes neurais MLP e SVM assim como o banco de dados selecionado foi Ecoli, Yeast e Parkinsons. Após tratarmos o arquivo de dados para o formato .arff de forma que obtemos os atributos.

No trabalho fizemos também, para exemplificar, representações gráficas scatter plot e boxplot a fim de facilitar a visualização do arranjo de dados. Assim como anotar as médias, desvios padrão, moda, mediana, cutose e obliquidade para cada um dos atributos usados.

Classificadores usados

KNN

O algoritmo k-Nearest Neighbor (kNN) é um algoritmo de aprendizado supervisionado do tipo lazy, introduzido por Aha et al. (1991). A ideia geral desse algoritmo consiste em encontrar os k exemplos rotulados mais próximos do exemplo não classificado e, com base no rótulo desses exemplos mais próximos, é tomada a decisão relativa à classe do exemplo não rotulado. Os algoritmos da família kNN requerem pouco esforço durante a etapa de treinamento. Em contrapartida, o custo computacional para rotular um novo exemplo é relativamente alto, pois, no pior dos casos, esse exemplo deverá ser comparado com todos os exemplos contidos no conjunto de exemplos de treinamento.

Redes neurais MLP

Nessas redes, cada camada tem uma função específica. A camada de saída recebe os estímulos da camada intermediária e constrói o padrão que será a resposta. As camadas intermediárias funcionam como extratoras de características, seus pesos são uma codificação de características apresentadas nos padrões de entrada e permitem que a rede crie sua própria representação, mais rica e complexa, do problema. Se existirem as conexões certas entre as unidades de entrada e um conjunto suficientemente grande de unidades intermediárias, pode-se sempre encontrar a representação que irá produzir o mapeamento correto da entrada para a saída através das unidades intermediária. Como provou Cybenko, a partir de extensões do Teorema de Kolmogoroff, são necessárias no máximo duas camadas intermediárias, com um número suficiente de unidades por camada, para se produzir quaisquer mapeamentos. Também foi provado que apenas uma camada intermediária é suficiente para aproximar qualquer função contínua.

SVM

As Máquinas de Vetores de Suporte (SVMs, do Inglês Support Vector Machines) cons- tituem uma técnica de aprendizado que vem recebendo crescente atenção da comunidade de Aprendizado de Máquina (AM). Os resultados da aplicação dessa técnica são comparáveis e muitas vezes superiores aos

obtidos por outros algoritmos de aprendizado, como as Redes Neurais Artificiais (RNAs). Exemplos de aplicações de sucesso podem ser encontrados em diversos domínios, como na categorização de textos, na análise de imagens e em Bioinformática (de nosso interesse).

Após tratarmos a base de dados transformando o arquivo .data em .arff, abrimos o programa weka e uma das bases de dados. À primeira vista temos o atributo nome como irrelevante, pois não traz informações quantitativas. Temos também outros atributos majoritariamente os dados agrupados no mesmo valor, assim como o atributo. Por isso removemos esses atributos por julgar irrelevante em na distribuição dos dados.

Tecnicas de amostragem

Utilizamos duas técnica de amostragem sendo elas StratifiedRemoveFolds
e SMOTE e comparamos as duas usadas nos três classificadores escolhidos.

StratifiedRemoveFolds
Esse filtro tem um conjunto de dados e dobra a saída, adequado para validação cruzada.

SMOTE
Remonta um conjunto de dados aplicando SMOTE (do inglês Synthetic Minority Oversampling TEchnique). O conjunto de dados original de caber inteiramente na memória. A quantidade de SMOTE e o número de vizinhos mais próximos podem ser especificados.

10-fold cross validation
Nesse treinamento cria-se 10 grupos independentes, cada fold é construído com 9/10 dos dados e testado com 1/10, é repetido 10 vezes e os resultados são expostos. Assim na fase de teste 10 modelos são construídos e avaliados se usado esse método de treinamento. Para os classificadores do explorer do WEKA nos mostra o todo, uma vez que não pode mostrar os 10 modelos.

Método Wrapper
Esse método usa um avaliador por subconjunto. Cria todos os possíveis subconjuntos do seu vetor característico, então usa um algoritmo de classificação para induzir os classificadores das características em cada subconjunto, o método irá considerar o subconjunto de características com cada um que o algoritmo de classificação com melhor perfomance. Para achar o vetor característico usa de técnicas de procura. Esse método nos mostra quais atributos são os melhores para se classificar os nossos dados.

1. Ecoli

1.1 Conjunto de Dados

Ecoli nome: nome do tipo Ecoli usada.

mcg: método McGeoch para reconhecimento de sequência sinal.
gvh: método de von Heijne para reconhecimento de sequência sinal.
lip: score von Heijne de sequência consenso de Sinal Peptidase II.
Atributo binário.
chg: Presença de carga no N-terminal de lipoproteína predita.
Atributo binário.
aac: score de análise discriminante do amino ácido contido fora da membrana e de proténas periplasmáticas.
alm1: score da membrana ALOM abrangendo programa de predição da região.
alm2: score do programa ALOM após excluir provável regiões sinal de clivagem da sequência.

Número de instâncias: 336

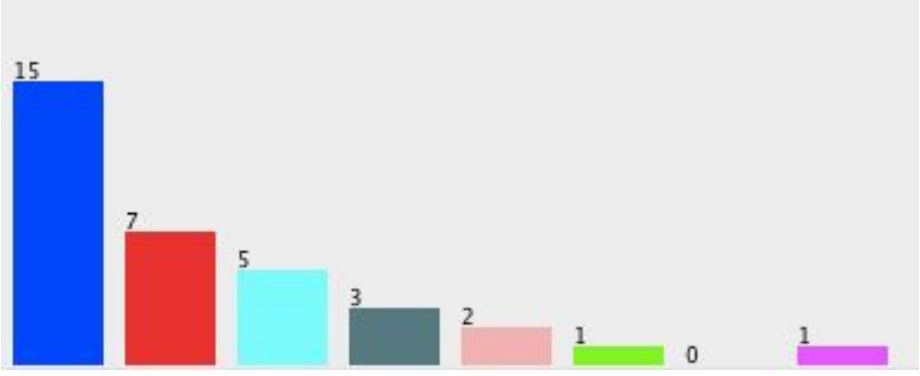
Número de atributos: 8 (7 preditivos, 1 nominal)

Valores de atributos faltando: 0

Distribuição de classe:
Cada classe representa um sítio de localização.

Classe	Instâncias
cp (citoplasma)	143
im (membrana interna sem sequência sinal)	77
pp (perisplasma)	52
imU (membrana interna sem sequência sinal de clivagem)	35
om (membrana externa)	20
omL (membrana externa da lipoproteína)	5
imL (membrana interna da lipoproteína)	2
imS (membrana interna com sequência sinal de clivagem)	2

Começamos aplicando a técnica de amostragem StratifiedRemoveFolds e comparamos os resultados para cada um dos classificadores. Nota-se que ao aplicá-lo possuímos o seguinte conjunto:

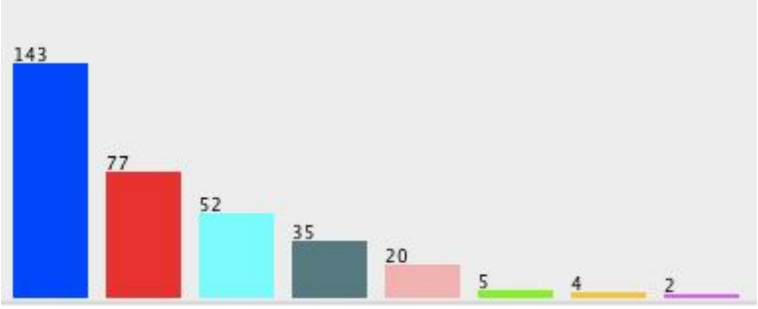


Restaram 34 instâncias de 5 atributos (mcg, gvh, aac, alm1e alm2) apenas, as classes apresentadas são cp, im, pp, imU, om, omL, imL e imS, respectivamente. Desse modo temos como resultados dos classificadores:

KNN-3 com 70% de percentage split	MLP- 10-fold-cross-validation	SVM-10-fold-cross-validation
=== Summary === Correctly Classified Instances 6 60 % Incorrectly Classified Instances 4 40 % Kappa statistic 0.5 Mean absolute error 0.1419 Root mean squared error 0.2759 Relative absolute error 70.3876 % Root relative squared error 84.4914 % Coverage of cases (0.95 level) 80 % Mean rel. region size (0.95 level) 50 % Total Number of Instances 10	=== Summary === Correctly Classified Instances 25 73.5294 % Incorrectly Classified Instances 9 26.4706 % Kappa statistic 0.6264 Mean absolute error 0.0976 Root mean squared error 0.2356 Relative absolute error 50.8748 % Root relative squared error 76.7956 % Coverage of cases (0.95 level) 85.2941 % Mean rel. region size (0.95 level) 36.3971 % Total Number of Instances 34	=== Summary === Correctly Classified Instances 22 64.7059 % Incorrectly Classified Instances 12 35.2941 % Kappa statistic 0.4531 Mean absolute error 0.1965 Root mean squared error 0.307 Relative absolute error 102.4043 % Root relative squared error 100.0671 % Coverage of cases (0.95 level) 94.1176 % Mean rel. region size (0.95 level) 75 % Total Number of Instances 34

Analisando lado a lado os resultados facilita nossa compreensão dos métodos, onde o classificador MLP possuiu a maior porcentagem de acerto e menor erro quadrado relativo e absoluto.

Agora fazemos a mesma comparação com os classificadores aplicando o filtro SMOTE no pré processo. Diferentemente do processo anterior, temos mais instâncias a serem avaliadas, agora com 338 (2 a mais que o original, resultante do novo conjunto de dados que o SMOTE realiza.



Aqui o gráfico nos mostra novamente como ficaram divididas as instâncias pelas classes, temos 338 instâncias, 5 atributos e as seis classes.

KNN-3 com 70% de percentage split	MLP- 10-fold-cross-validation	SVM-10-fold-cross-validation
=== Summary === Correctly Classified Instances 86 85.1485 % Incorrectly Classified Instances 15 14.8515 % Kappa statistic 0.7997 Mean absolute error 0.0563 Root mean squared error 0.1865 Relative absolute error 30.2604 % Root relative squared error 60.5583 % Coverage of cases (0.95 level) 91.0891 % Mean rel. region size (0.95 level) 17.203 % Total Number of Instances 101	=== Summary === Correctly Classified Instances 277 81.9527 % Incorrectly Classified Instances 61 18.0473 % Kappa statistic 0.7513 Mean absolute error 0.058 Root mean squared error 0.187 Relative absolute error 31.5681 % Root relative squared error 61.8383 % Coverage of cases (0.95 level) 92.8994 % Mean rel. region size (0.95 level) 22.0414 % Total Number of Instances 338	=== Summary === Correctly Classified Instances 277 81.9527 % Incorrectly Classified Instances 61 18.0473 % Kappa statistic 0.7458 Mean absolute error 0.1907 Root mean squared error 0.2967 Relative absolute error 103.858 % Root relative squared error 98.1538 % Coverage of cases (0.95 level) 98.2249 % Mean rel. region size (0.95 level) 75.2589 % Total Number of Instances 338

Aqui o algoritmo KNN classificou melhor as instâncias, um dos motivos seriam o maior número de instâncias do conjunto teste, comparado ao filtro anterior. E ainda possui o menor erro relativo de

todos os classificadores. Apesar do MLP e SVM classificarem de forma muito parecida os dados, notamos pela análise de erro relativo e erro quadrático que o SVM foi mais elevado.

Agora iremos comparar o desempenho dos dados feitos anteriormente por filtro com o método wrapper a fim de verificar a eficácia dos métodos e realizar uma análise crítica dos mesmos. Uma breve explicação sobre o método wrapper colocamos a seguir.

No explorer do WEKA, carregamos o conjunto de dados original em seguida vamos até a aba *select attributes*. No campo attribute evaluator clicamos *choose* e selecionamos (a) *ClassifierSubsetEval* e clicando sobre o nome selecionado temos duas opções, na primeira selecionamos cada um dos classificadores usados anteriormente, *knn-3, mlp* e *SMO* como classificador e na segunda opção deixamos como *true* pois queremos um grupo de treinamento. No campo *Attribute selection mode* usamos *cross-validation* como sugerido e pressionamos o botão Start. A intenção desse procedimento é ter como resposta os atributos que melhor classificam nossas instâncias. Em seguida usamos também (b) *WrapperSubsetEval*, com todo o resto do procedimento idêntico, logo temos como respostas:

(a)

knn-3	MLP	SVM
=== Attribute selection 10 fold cross-validation (stratified), seed: 1 === number of folds (%) attribute 10(100 %) 1 mcg 8(80 %) 2 gvh 10(100 %) 3 aac 10(100 %) 4 alm1 6(60 %) 5 alm2	=== Attribute selection 10 fold cross-validation (stratified), seed: 1 === number of folds (%) attribute 10(100 %) 1 mcg 9(90 %) 2 gvh 10(100 %) 3 aac 10(100 %) 4 alm1 5(50 %) 5 alm2	Memória insuficiente para realizar operação.

(b)

KNN-3	MLP	SVM
=== Attribute selection 10 fold cross-validation (stratified), seed: 1 === number of folds (%) attribute 10(100 %) 1 mcg 9(90 %) 2 gvh 10(100 %) 3 aac 10(100 %) 4 alm1 9(90 %) 5 alm2	=== Attribute selection 10 fold cross-validation (stratified), seed: 1 === number of folds (%) attribute 10(100 %) 1 mcg 9(90 %) 2 gvh 10(100 %) 3 aac 10(100 %) 4 alm1 8(80 %) 5 alm2	Memória insuficiente para realizar a operação

Esse procedimento *Wrapper* avalia atributos do subconjunto dos dados de treinamento ou separa um conjunto test. Usa um classificador para estimar o mérito de cada atributo na seleção. Com isso, é de interesse os atributos que possuem 100% ou 'mérito' elevado. Comparamos os resultados sem filtro, porém usando apenas os atributos com 100% de mérito com os mesmos classificadores. Selecionamos então os atributos 1, 3 e 4.

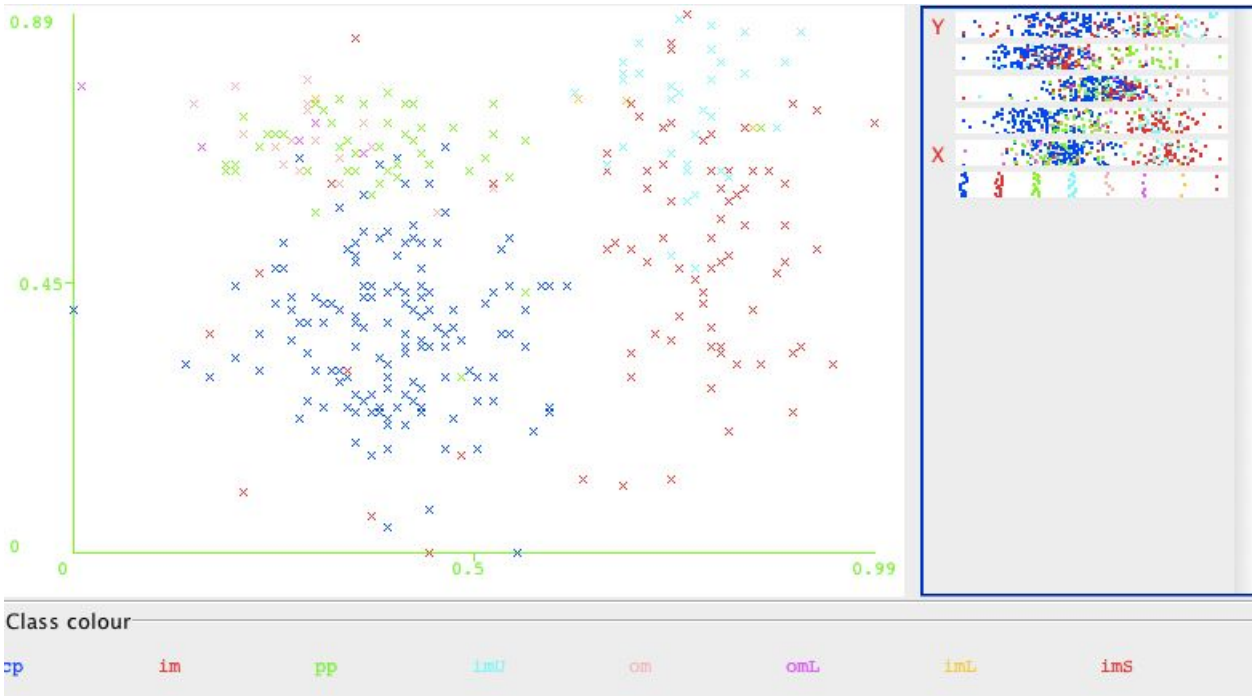
KNN-3	MLP	SVM
=== Summary === Correctly Classified Instances 275 81.8452 % Incorrectly Classified Instances 61 18.1548 % Kappa statistic 0.7461 Mean absolute error 0.0594 Root mean squared error 0.1893 Relative absolute error 32.4956 % Root relative squared error 62.7596 % Coverage of cases (0.95 level) 91.9643 % Mean rel. region size (0.95 level) 17.8943 % Total Number of Instances 336	=== Summary === Correctly Classified Instances 281 83.631 % Incorrectly Classified Instances 55 16.369 % Kappa statistic 0.7737 Mean absolute error 0.06 Root mean squared error 0.1808 Relative absolute error 32.8086 % Root relative squared error 59.9183 % Coverage of cases (0.95 level) 95.2381 % Mean rel. region size (0.95 level) 25.7813 % Total Number of Instances 336	Memória insuficiente para realizar o processo.

Comparando-se:

KNN-3 com 70% de percentage split	MLP- 10-fold-cross-validation	SVM-10-fold-cross-validation
=== Summary === Correctly Classified Instances 86 85.1485 % Incorrectly Classified Instances 15 14.8515 % Kappa statistic 0.7997 Mean absolute error 0.0563 Root mean squared error 0.1865 Relative absolute error 30.2604 % Root relative squared error 60.5583 % Coverage of cases (0.95 level) 91.0891 % Mean rel. region size (0.95 level) 17.203 % Total Number of Instances 101	=== Summary === Correctly Classified Instances 277 81.9527 % Incorrectly Classified Instances 61 18.0473 % Kappa statistic 0.7513 Mean absolute error 0.058 Root mean squared error 0.187 Relative absolute error 31.5681 % Root relative squared error 61.8383 % Coverage of cases (0.95 level) 92.8994 % Mean rel. region size (0.95 level) 22.0414 % Total Number of Instances 338	=== Summary === Correctly Classified Instances 277 81.9527 % Incorrectly Classified Instances 61 18.0473 % Kappa statistic 0.7458 Mean absolute error 0.1907 Root mean squared error 0.2967 Relative absolute error 103.858 % Root relative squared error 98.1538 % Coverage of cases (0.95 level) 98.2249 % Mean rel. region size (0.95 level) 75.2589 % Total Number of Instances 338

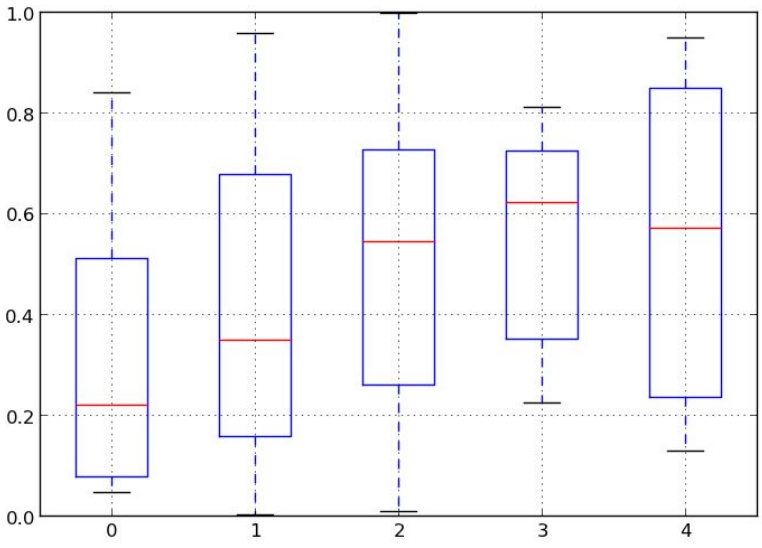
Vemos então que o método usado com o filtro SMOTE com o classificador KNN-3 é mais representativo para a nossa base de dados para uma possível seleção, porém os resultados obtidos são insuficientes por uma classificação correta de 85% apesar de alta não é eficaz.

1.2 Scatter plot



No eixo X temos a classe alm2 e no eixo Y a classe mcg, a imagem ilustra como estão divididos as instâncias de acordo com o que cada uma dessas duas classes representam, e as instâncias são representadas por cores que indicam a que classe pertencem.

1.3 Box plot



2. Yeast

2.1 Conjunto de Dados

Número de instâncias: 1484.
Número de atributos : 9 (8 preditivo, 1 nominal)

Informaçoes de atributos.
Nome da sequência: Número de acesso para o banco de dados SWISS-PROT
mcg: método McGeoch para reconhecimento de sequência sinal.
gvh: método de von Heijne para reconhecimento de sequência sinal.
alm: score de ALOM.
mit: score de análise de discriminante do conteudo do aminoácido N-terminal
erl: Presença do "HDEL". Atributo binário.
pox: Peroxissomal sinal alvo no C-terminal.
vac: Score de analise discriminante do amino ácido contendo vacuolo e proteina extracelular.
nuc: Score de análise discriminante de sinal de localização nuclear de proteínas nucleares e não nucleares.

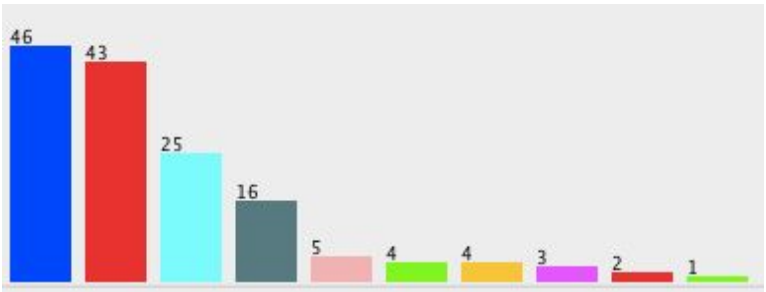
Atributos com valores faltando: nenhum.

Distribuição de classe.

CYT (cytosolic or cytoskeletal)	463
NUC (nuclear)	429
MIT (mitochondrial)	244
ME3 (membrane protein, no N-terminal signal)	163
ME2 (membrane protein, uncleaved signal)	51
ME1 (membrane protein, cleaved signal)	44
EXC (extracellular)	37
VAC (vacuolar)	30
POX (peroxisomal)	20
ERL (endoplasmic reticulum lumen)	5

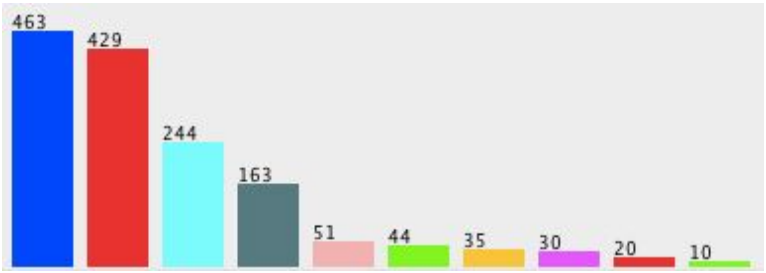
Realizamos os mesmos procedimento feitos com o conjunto de dados Ecoli. Aplicamos os filtros e vemos os resultados com cada classificador e em seguida o método *Wrapper* e por fim, comparamos resultados.

StratifiedRemoveFolds, ficamos com 149 instâncias divididas da seguinte maneira:



KNN-3	MLP	SVM
=== Summary === Correctly Classified Instances 71 47.651 % Incorrectly Classified Instances 78 52.349 % Kappa statistic 0.3044 Mean absolute error 0.1196 Root mean squared error 0.2725 Relative absolute error 75.908 % Root relative squared error 97.5073 % Coverage of cases (0.95 level) 75.1678 % Mean rel. region size (0.95 level) 20.8725 % Total Number of Instances 149	=== Summary === Correctly Classified Instances 76 51.0067 % Incorrectly Classified Instances 73 48.9933 % Kappa statistic 0.3628 Mean absolute error 0.114 Root mean squared error 0.2683 Relative absolute error 72.348 % Root relative squared error 95.99 % Coverage of cases (0.95 level) 83.2215 % Mean rel. region size (0.95 level) 32.7517 % Total Number of Instances 149	Máquina com memória insuficiente.

Aplicando o filtro SMOTE temos 1489 instâncias.



KNN-3	MLP	SVM
=== Summary === Correctly Classified Instances 807 54.1974 % Incorrectly Classified Instances 682 45.8026 % Kappa statistic 0.396 Mean absolute error 0.1027 Root mean squared error 0.26 Relative absolute error 65.8975 % Root relative squared error 93.1931 % Coverage of cases (0.95 level) 76.0242 % Mean rel. region size (0.95 level) 18.6367 % Total Number of Instances 1489	=== Summary === Correctly Classified Instances 866 58.1598 % Incorrectly Classified Instances 623 41.8402 % Kappa statistic 0.456 Mean absolute error 0.1047 Root mean squared error 0.2378 Relative absolute error 67.2064 % Root relative squared error 85.2548 % Coverage of cases (0.95 level) 92.7468 % Mean rel. region size (0.95 level) 33.6199 % Total Number of Instances 1489	Memória insuficiente para realizar a operação

A acurácia aumentou significativamente usando o filtro SMOTE. Onde o classificador MLP teve aumento de 7% assim como o KNN-3.

Agora aplicamos a técnica de Wrapper já citada e rodamos os classificadores com os atributos mais relevante.

(a)

KNN-3	MLP	SVM
=== Attribute selection 10 fold cross-validation (stratified), seed: 1 === number of folds (%) attribute 10(100 %) 1 mcg 10(100 %) 2 gvh 10(100 %) 3 alm 10(100 %) 4 mit 2(20 %) 5 vac 10(100 %) 6 nuc	=== Attribute selection 10 fold cross-validation (stratified), seed: 1 === number of folds (%) attribute 10(100 %) 1 mcg 10(100 %) 2 gvh 10(100 %) 3 alm 10(100 %) 4 mit 10(100 %) 5 vac 10(100 %) 6 nuc	Memória insuficiente para realizar a operação

(b)

KNN-3	MLP	SVM
=== Attribute selection 10 fold cross-validation (stratified), seed: 1 === number of folds (%) attribute 10(100 %) 1 mcg 10(100 %) 2 gvh 10(100 %) 3 alm 10(100 %) 4 mit 3(30 %) 5 vac 10(100 %) 6 nuc	=== Attribute selection 2 fold cross-validation (stratified), seed: 1 === number of folds (%) attribute 1(50 %) 1 mcg 2(100 %) 2 gvh 2(100 %) 3 alm 2(100 %) 4 mit 2(100 %) 5 vac 2(100 %) 6 nuc	Memória insuficiente para realizar a operação

Como o knn-3 nos dá um atributo com baixo 'mérito', retiramos o atributo 5 e realizamos novamente as classificações e o atributo 1 para realizarmos o MLP.

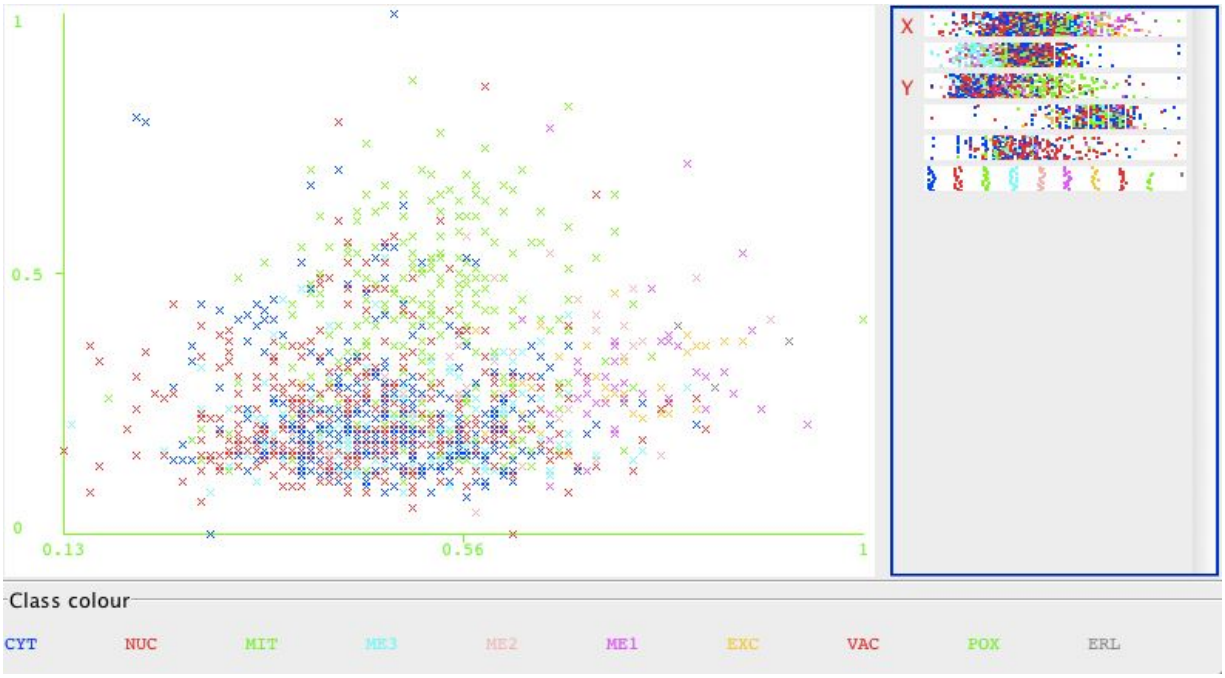
KNN-3	MLP	SVM
=== Summary === Correctly Classified Instances 809 54.5148 % Incorrectly Classified Instances 675 45.4852 % Kappa statistic 0.3995 Mean absolute error 0.1016 Root mean squared error 0.2573 Relative absolute error 65.3202 % Root relative squared error 92.3048 % Coverage of cases (0.95 level) 77.6954 % Mean rel. region size (0.95 level) 18.6927 % Total Number of Instances 1484	=== Summary === Correctly Classified Instances 841 56.6712 % Incorrectly Classified Instances 643 43.3288 % Kappa statistic 0.4322 Mean absolute error 0.1094 Root mean squared error 0.24 Relative absolute error 70.3482 % Root relative squared error 86.1162 % Coverage of cases (0.95 level) 92.9919 % Mean rel. region size (0.95 level) 36.7992 % Total Number of Instances 1484	Memória insuficiente para realizar a operação

Comparando-se :

KNN-3	MLP	SVM
=== Summary === Correctly Classified Instances 807 54.1974 % Incorrectly Classified Instances 682 45.8026 % Kappa statistic 0.396 Mean absolute error 0.1027 Root mean squared error 0.26 Relative absolute error 65.8975 % Root relative squared error 93.1931 % Coverage of cases (0.95 level) 76.0242 % Mean rel. region size (0.95 level) 18.6367 % Total Number of Instances 1489	=== Summary === Correctly Classified Instances 866 58.1598 % Incorrectly Classified Instances 623 41.8402 % Kappa statistic 0.456 Mean absolute error 0.1047 Root mean squared error 0.2378 Relative absolute error 67.2064 % Root relative squared error 85.2548 % Coverage of cases (0.95 level) 92.7468 % Mean rel. region size (0.95 level) 33.6199 % Total Number of Instances 1489	Memória insuficiente para realizar a operação

Notamos que o melhor desempenho acontece quando usamos o filtro SMOTE para as classificações.

2.2 Scatter plot



No eixo X temos a classe gvh e no Y a classe mit. Notamos que comparando com o conjunto de dados da Ecoli assim como a dificuldade dos classificadores para os dados de Yeast.arff vemos uma mistura maior das classes representados no scatter plot.

Conclusão

Fica claro que o processo de mineração de dados biológicos depende diretamente e de forma completamente intrínseca da maneira que avaliamos e consideramos os atributos relevantes àquilo que procuramos. Não obstante há vários métodos de classificação, e que em cada exemplo devemos considerar melhor os algoritmos usados e considerar sempre a qualidade dos dados coletados. Esse trabalho nos mostrou que para cada conjunto é interessante um tipo de abordagem mas não utilizá-lo de forma isolada dos demais métodos. Como mostrado no primeiro grupo de dados, o método Wrapper foi mais eficiente para seleção, porém o mesmo não ocorre para o segundo conjunto de dados. Assim como há influência dos classificadores no resultado há parcela importante na distribuição dos dados e quantidade utilizada. Infelizmente o uso de máquina elevado desfavorece o uso da função SMO, que inicialmente foi eficaz, porém com muitos dados não foi possível utilizar.

Denovo, ressalta-se que os resultados de melhor algoritmo e pior algoritmo devem ser interpretados com atenção: o melhor algoritmo para resolver um certo problema ou tipo de problema pode não ser o ideal no caso de um problema de outra natureza.