Dataset 12

Introducción

Se presenta un conjunto de datos biológicos organizados en forma tabular que contienen varios atributos y un valor de la clase. Estos datos contienen información de la **estructura de las proteínas y sus propiedades**. Una **proteína** contiene una o varias secuencias de aminoácidos (cadenas de caracteres). Cada **secuencia** de aminoácidos es dividida en múltiples **subsecuencias** (todas las subsecuencias posibles de tamaño mínimo 2).

Mediante el sistema denominado **ASPFgen** se generan, a partir de un conjunto de secuencias de proteínas (tanto de training como de test), todas las subsecuencias posibles de aminoácidos y se genera una fila en una tabla por cada una de ellas. En cada fila se almacena un conjunto de atributos que se especifican en la siguiente sección de este documento. La **clase es discreta** y su valor indica en qué intervalo de distancia real se encuentran los aminoácidos primero y último de la subsecuencia que la fila representa.

Estos datos quedan almacenados en archivos en formato ARFF de Weka, pensados para la predicción de la estructura de nuevas proteínas o para la selección de atributos relevantes para la predicción. La predicción (clasificación) consistirá en determinar en qué intervalo de distancia se encuentran dos aminoácidos cualesquiera de una proteína de test. Cualquier clasificador debería poder ser aplicado: vecinos más cercanos, redes neuronales, algoritmos evolutivos, etc.

Atributos del dataset

Este dataset contiene varias tablas, pero todas tienen el conjunto de atributos que se describe a continuación. Se recuerda que cada fila representa una subsecuencia de aminoácidos.

Atributo	Tipo	Descripción
protein	String	Código de proteína
seq	String	Código de secuencia dentro de la proteína
naa1	int	Índice del primer aminoácido de la subsecuencia dentro de la secuencia
naa2	int	Índice del último aminoácido de la subsecuencia dentro de la secuencia (naa1 < naa2)
aa1	char	Símbolo del primer aminoácido de la subsecuencia (20 distintos)
aa2	char	Símbolo del último aminoácido de la subsecuencia (20 distintos)
(profile)	(varios tipos)	Conjunto de atributos del profile de la subsecuencia. Aquí se encuentran los **ÚNICOS** atributos que deben usarse para la selección de atributos o para la predicción.
class	int	Número de intervalo de distancia entre primer y último aminoácido de la subsecuencia. La clase puede tener **2 valores** diferentes: **0 y 1**. Los intervalos de distancia a los que hace referencia cada valor son: 0=(8,inf) 1=[0,8]. Ésta es una discretización usada en múltiples trabajos en la literatura [1].

Materiales

En la página web http://www.gualberto.es/datasets se encuentra el **dataset número 12**. Este dataset consta de varios archivos descargables que se describen a continuación.

- **dataset12_P62_R.zip (aún no disponible)**: Contiene el fichero *dataset12_R.arff*. Es la tabla con todos los ejemplos de training. Se obtuvo a partir de un conjunto de 400 tablas clasificadas por los atributos aa1 (20 distintos) y aa2 (20 distintos) (20x20=400). Se hizo un editado mediante muestreo aleatorio estratificado usando valores porcentuales lo suficientemente bajos para satisfacer las pruebas de rendimiento. Posteriormente, las 400 tablas editadas se fusionaron en este fichero.
- **dataset12_P62_T.zip (aún no disponible)**: Contiene el fichero *dataset12_T.arff*. Es la tabla con todos los ejemplos de test. Se obtuvo a partir de todas las proteínas de test y se eliminaron aquellas filas tales que naa2-naa1 < 24. Esto se hizo así debido a que dichas filas no deben evaluarse [1].
- **dataset12_P62_R400.zip**: Contiene 400 ficheros de training con nombres *dataset12_R_X-Y.arff*, donde X-Y son todos los pares de valores posibles de los atributos aa1 y aa2. En cada uno de estos archivos se encuentran únicamente los ejemplos con los valores de los atributos aa1 y aa2 especificado en su nombre de archivo (por ejemplo, el fichero *dataset12_R_L-W.arff* contiene sólo los ejemplos con atributo aa1="L" y aa2="W"). Por este motivo, en estas tablas se han eliminado los atributos aa1 y aa2, ya que tienen el mismo valor en cada tabla. Se hizo un editado mediante muestreo aleatorio estratificado usando valores porcentuales lo suficientemente bajos para satisfacer las pruebas de rendimiento.
- **dataset12_P62_T400.zip**: Contiene 400 ficheros de test con nombres *dataset12_T_X-Y.arff*, con el mismo reparto de datos que el anterior archivo. Se eliminaron aquellas filas tales que naa2-naa1 < 24, porque no se deben evaluar [1]. En estas tablas se han eliminado los atributos aa1 y aa2, ya que tienen el mismo valor en cada tabla.
- **dataset12_P62_R400_onlyprofile.zip**: Contiene lo mismo que el dataset12_R400, pero se han eliminado todos los atributos salvo los del profile (únicos necesarios para predecir) y la clase.
- **dataset12_P62_T400_onlyprofile.zip**: Contiene lo mismo que el dataset12_T400, pero se han eliminado todos los atributos salvo los del profile (únicos necesarios para predecir) y la clase.

Evaluación de resultados

Todas las condiciones que se resumen en esta sección del documento están formalmente descritas en el artículo de referencia de evaluación en CASP9 [1].

Para realizar predicciones (o evaluar atributos en un proceso de selección a atributos) debe utilizarse como **test** el fichero ***dataset12_T.arff*** o algún ***dataset12_T_X-Y.arff*** y, por cada ejemplo de test, utilizar como **training** SÓLO aquellas filas de ***dataset12_R.arff*** cuyos atributos aa1 y aa2 coincidan con los del ejemplo de test a predecir, o bien usar la tabla ***dataset12_R_X-Y.arff*** correspondiente al par de valores de aa1 y aa2 (usando todas sus filas sin filtro ninguno).

En concreto, las medidas utilizadas para evaluar predicciones serán **accuracy**, **coverage** y **F-measure** cuyos valores serán obtenidos mediante las siguientes fórmulas:

accuracy (precision) = TP / (TP + FP)

coverage (recall) = TP / (TP + FN)

F-measure = 2 * accuracy * coverage / (accuracy + coverage)

- **TP** es true positives o número de predicciones cuyo valor, **tanto real como predicho**, de la clase es **1**.
- **FP** es false positives o número de predicciones cuyo valor **real** de la clase es **0** y el valor **predicho** para la misma es **1**.
- **FN** es false negatives o número de predicciones cuyo valor **real** de la clase es **1** y el valor **predicho** para la misma es **0**.

La obtención de estas medidas debe hacerse computando el número de TP, FP y FN por cada código de proteína (cada valor del atributo 'protein'), obteniéndose un valor de accuracy y coverage **por cada proteína distinta de test**. Finalmente, se obtendrá una media para todas las proteínas, resultando **un único valor de accuracy y coverage** para todos los ejemplos de test.

Para disponer de **valores de referencia**, se puede considerar un buen resultado si se obtiene **accuracy ≈ 0.09** y **coverage ≈ 0.31**. Si se obtienen estos valores o mejores, la comparación con otros métodos actuales aún no es inmediata. Para la **comparación final** con artículos actuales de referencia será necesario en última instancia hacer un **ranking** de las predicciones que hayan devuelto **clase predicha 1.**

Para elaborar dicho ranking es necesario disponer de valores de bondad o confianza devueltos por el predictor. Por ello, sería interesante, **SÓLO** en el caso de aplicar predictores, que éstos asignen un **valor de bondad para sus predicciones de clase 1**. **El ranking situará en primer lugar las predicciones con mayor valor de bondad**. No obstante, para la comparación con los valores anteriormente mencionados (accuracy ≈ 0.09 y coverage ≈ 0.31) no se necesita realizar dicho ranking. No se propone realizar ranking ni se suministran valores de referencia usando ranking por dos razones:

- La primera es porque la diferencia entre hacer ranking o no en términos de accuracy o coverage suele tener un comportamiento similar, independientemente del método de predicción o los atributos que se usen. En concreto, cuando se hace el ranking final, accuracy aumenta desde 0.09 aproximadamente hasta 0.26 (con un ranking L/5) mientras que coverage decrece desde 0.31 aproximadamente hasta 0.06, siendo estos valores ya comparables.
- La segunda razón es porque el uso del ranking requiere, como se ha comentado, generar valores de bondad por cada predicción de clase 1 y añadir el paso extra de generar dicho ranking. Por simplicidad, se propone relegar el ranking a la fase final de comparación de resultados.

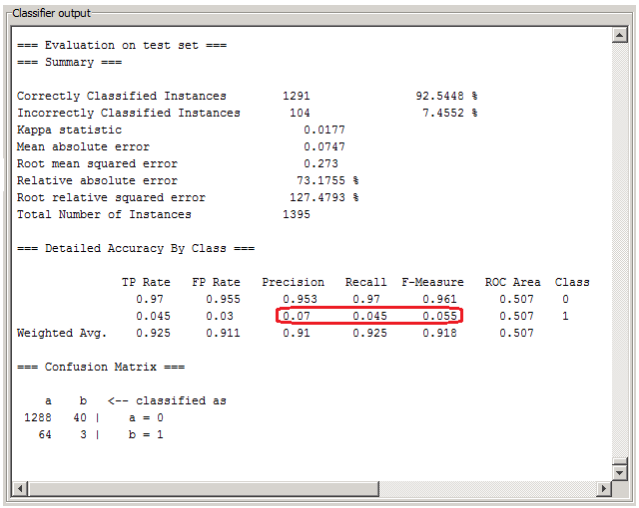


Figura 1: Medidas de Weka a tener en cuenta

En la figura 1 se muestran, rodeadas en color rojo, las medidas de Weka correspondientes a accuracy (precision), coverage (recall) y F-measure a tener en cuenta en la evaluación de predicciones. Si se pueden utilizar dos medidas para optimizar simultáneamente, usar accuracy y coverage; en caso contrario, usar F-measure.

La figura 1 muestra los resultados obtenidos usando IBk (k=1), training *dataset12_R_L-L.arff* y test *dataset12_T_L-L.arff*. Se usó un 10% en el muestreo aleatorio estratificado y la ejecución tardó 5 segundos en la máquina anteriormente mencionada. Se probaron también otros porcentajes de reducción y se analizó el tiempo de ejecución y los resultados obtenidos. Por ejemplo, se utilizó un porcentaje del **33.33%**, que para training *dataset12_R_L-L.arff* y test *dataset12_T_L-L.arff* obtuvo un **accuracy medio de 0.0596** y **coverage de 0.0599** en un tiempo medio de **9 segundos**. Ver la tabla de resultados según porcentaje de reducción de ejemplos.

Requisito de rendimiento

Se ha impuesto un **límite en el tiempo de ejecución** de una predicción completa mediante **1-NN** (IBk de Weka, K=1) de todos los ejemplos de test de cada tabla *dataset12_T_X-Y.arff*. Este límite temporal es de **1 minuto**. Es decir, se ha usado como training la tabla de training más extensa (*dataset12_R_L-L.arff*) y como test su tabla de test correspondiente (*dataset12_T_L-L.arff*), que también era la tabla de test más extensa. Con estas tablas se utilizó el algoritmo IBk de Weka (K=1) y tardó menos de 1 minuto. Estas pruebas se han realizado en una máquina con 2 procesadores Intel Xeon X5482 a 3.2GHz con 32GB de RAM.

Para conseguir este requisito, se probaron distintas técnicas de editado sobre los ficheros de training. No se deben reducir los ejemplos de test porque se dejarían de realizar predicciones necesarias para la evaluación y comparación de resultados. Se probó **muestreo estratificado aleatorio** con diferentes porcentajes de reducción, dando lugar a resultados demasiado diversos (ver página 5). Se probaron, por tanto, **diferentes algoritmos de editado** (ver página 6). Se optó por emplear el **algoritmo VSM (con K=3)** de la herramienta Keel, por su buen comportamiento en términos de **eficacia y eficiencia**. Se probó VSM (K=3) **con validación cruzada** (ver página 7). Finalmente, **se redujo el conjunto de atributos** en el profile inicial (62 atributos), obteniendo un **subconjunto resultante de la unión de 4 subconjuntos**: ORIG+CFS, ORIG+SymFCBF, VSM+CFS y VSM+SymFCBF (ver página 8). Este subconjunto resultante, para el **dataset L-L**, obtuvo **16 atributos**. Se analizó también el efecto del orden de aplicación de FS y editado (ver página 10).

Tabla de resultados según porcentaje de reducción de ejemplos con muestreo aleatorio estratificado

Se ha probado a reducir el número de filas del archivo *dataset12_R_L-L.arff* (60.360 filas) con diferentes porcentajes de reducción. El dataset resultante de cada experimento (numerado en la columna Nº exp) fue obtenido aleatoriamente con distintas semillas (se usó System.currentTimeMillis() para generar las semillas). Los resultados de precisión y recall de la siguiente tabla fueron obtenidos usando IBk (k=1), training el *dataset12_R_L-L.arff* reducido y test *dataset12_T_L-L.arff*. El profile elegido en este estudio contiene 62 atributos.

% red.	Nº exp.	T. ejec.	Nº filas	Precision	Recall
10%	1	5 seg.	6036	0.07	0.045
	2	3 seg.		0.066	0.06
	3	3 seg.		0.055	0.045
	4	3 seg.		0.052	0.06
	5	3 seg.		0.034	0.03
	6	"		0.023	0.03
	7	"		0.035	0.045
	8	"		0.079	0.075
	9	"		0.056	0.06
	10	"		0.021	0.015
20%	1	9 seg.	12072	0.00	0.00
	2	5 seg.		0.051	0.06
	3	5 seg.		0.068	0.06
	4	5 seg.		0.015	0.015
	5	5 seg.		0.068	0.075
	6	"		0.045	0.06
	7	"		0.091	0.09
	8	"		0.033	0.03
	9	"		0.066	0.06
	10	"		0.054	0.045
33.33%	1	9 seg.	20120	0.013	0.015
	2	9 seg.		0.03	0.03
	3	9 seg.		0.058	0.06
	4	9 seg.		0.058	0.06
	5	8 seg.		0.129	0.119
	6	9 seg.		0.055	0.045
	7	"		0.063	0.075
	8	"		0.064	0.075
	9	"		0.074	0.06
	10	"		0.052	0.06
100%	1	28 seg.	60360	0.048	0.06

Tabla de resultados según algoritmo de editado (same training and test)

Se ha probado a reducir el número de filas del archivo *dataset12_R_L-L.arff* (60.360 filas) con diferentes algoritmos de editado. Los resultados de precisión y recall de la siguiente tabla fueron obtenidos usando IBk (k=1), training el *dataset12_R_L-L.arff* reducido y test *dataset12_T_L-L.arff*. El profile elegido en este estudio contiene 62 atributos. La reducción de *dataset12_R_L-L.arff* fue realizada con la herramienta Keel usando **el mismo conjunto de training y test (el fichero *dataset12_R_L-L.arff*)**.

Algoritmo	Parámetros	T. ejec.	Nº filas	Precision	Recall
ENN	K=1		54660	0	0
CNN	K=1		5700	0.048	0.627
CNN	K=3		4117	0.048	0.925
CHC	PSize=50,NEval=10000,Alfa=0.5,K=1	48h 37’	23	0	0
VSM	K=10	1h 23’	6114	0.059	0.567
VSM	K=5	1h 18’	5527	0.06	0.597
VSM	K=3	1h 20’	5989	0.063	0.582
VSM	K=1	1h 21’	7531	0.046	0.269
ENRBF	Por defecto en Keel		56768	0	0
ModelCS	Por defecto en Keel		57788	0	0
ENN-Th	K=1		51193	0	0
POP	Por defecto en Keel		43112	0.051	0.075
PSR-CG	Por defecto en Keel		9948	0.046	0.209
RNG	No terminan				
GGA	No terminan				
PBIL	No terminan				
SGA	No terminan				

Tabla de resultados según algoritmo de editado (cross validation)

Se ha probado a reducir el número de filas del archivo *dataset12_R_L-L.arff* (60.360 filas) con diferentes algoritmos de editado. Los resultados de precisión y recall de la siguiente tabla fueron obtenidos usando IBk (k=1), training el *dataset12_R_L-L.arff* reducido y test *dataset12_T_L-L.arff*. El profile elegido en este estudio contiene 62 atributos. La reducción de *dataset12_R_L-L.arff* fue realizada con la herramienta Keel usando una **validación cruzada de 10 bolsas sobre el fichero *dataset12_R_L-L.arff***.

Algoritmo	Fold	Filas.O	Tej.O	Prec.O	Rec.O	Filas.R	Tej.R	Prec.R	Rec.R
VSM k=3	1	54324	37''	0.337	0.309	5349	9''	0.078	0.54
	2	54324	38''	0.314	0.248	5346	9''	0.071	0.479
	3	54324	1' 35"	0.346	0.312	5378	9''	0.077	0.552
	4	…	1' 28"	0.292	0.265	5382	…	0.07	0.487
	5		…	0.318	0.287	5343		0.076	0.524
	6			0.329	0.304	5456		0.075	0.513
	7			0.27	0.226	5424		0.072	0.474
	8			0.306	0.273	5326		0.066	0.448
	9			0.306	0.278	5360		0.072	0.506
	10			0.331	0.281	5403		0.079	0.553
	AVG±STD			**0.314±0.021**	**0.278±0.025**			**0.0736±0.003**	**0.5076±0.033**

Nota.- Las columnas de color verde (con nombres que terminan en ".O"), se refieren al conjunto original (sin reducir). Las columnas de color rojo (con nombres que terminan en ".R"), se refieren al conjunto una vez reducido.

Tabla de resultados según algoritmo de editado (independent training and test) y algoritmo de selección de atributos

Se ha probado a reducir el número de filas del archivo *dataset12_R_L-L.arff* (60.360 filas) con diferentes algoritmos de editado. Los resultados de precisión y recall de la siguiente tabla fueron obtenidos usando IBk (k=1), training el *dataset12_R_L-L.arff* reducido y test *dataset12_T_L-L.arff*. El profile elegido en este estudio contiene 62 atributos. La reducción de *dataset12_R_L-L.arff* fue realizada con la herramienta Keel usando **como training el fichero *dataset12_R_L-L.arff* y como test el fichero *dataset12_T_L-L.arff***. Se han aplicado varios algoritmos de **selección de atributos** de Weka sobre el conjunto original y los conjuntos reducidos.

Num	Dataset	#Filas	#Atributos: {Lista Atributos}	T.ejec.	Precision	Recall
1	ORIG	60360	62	28”	0.048	0.06
2	VSM_stt	5989	62	10”	0.063	0.582
3	VSM_cv10f	≈5400	62	9”	0.073±0.00	0.507±0.03
4	VSM_itt	5989	62	10”	0.063	0.582
5	ORIG+CFS	60360	5: {1,15,16,19,20}		0.046	0.045
6	VSM_stt+CFS_1	5989	2: {1,9}		0.063	0.403
7	VSM_stt+CFS_2	5989	1: {1}		0.047	0.209
8	VSM_stt+CFS_3	5989	3: {10,18,19} (sin 1)		0.048	0.358
9	VSM_stt+CFS_4	5989	4: {10,14,18,19} (sin 1)		0.048	0.418
10	ORIG{1,9,10,14,15,16,18,19,20}	60360	9: {1,9,10,14,15,16,18,19,20}		0.081	0.09
11	VSM_stt{1,9,10,14,15,16,18,19,20}	5989	9: {1,9,10,14,15,16,18,19,20}		0.054	0.478
12	ORIG+SymFCBF(fulltrain)	60360	2: {1,5}		0	0
13	ORIG+SymFCBF(cv10f)	60360	7: {5,1,8,57,17,46,61}		0.057	0.06
14	VSM_stt+SymFCBF(fulltrain)	5989	1: {1}		0.047	0.209
15	VSM_stt+SymFCBF(cv10f)	5989	2: {1,32}		0.055	0.358
16	ORIG{1,5,8,9,10,14,15,16,17,18,19,20,32,46,57,61}	60360	16:{1,5,8,9,10,14,15,16,17,18,19,20,32,46,57,61}		**0.18**	0.134
17	VSM_stt{1,5,8,9,10,14,15,16,17,18,19,20,32,46,57,61}	5989	16:{1,5,8,9,10,14,15,16,17,18,19,20,32,46,57,61}		0.052	0.463
18	ORIG{1,5,8,15,16,17,19,20,46,57,61}	60360	11: {1,5,8,15,16,17,19,20,46,57,61}		0.076	0.075
19	VSM_stt{1,9,10,14,18,19,32}	5989	7: {1,9,10,14,18,19,32}		0.057	0.478
20	ORIG{1,9,10,14,18,19,32}	60360	7: {1,9,10,14,18,19,32}		0.075	0.075

Notas:

- El sufijo “**_stt**” sobre el nombre de un algoritmo de editado indica que se ha utilizado el mismo conjunto de training y test en el mismo (stt = same training and test).
- El sufijo “**_cv10f**” sobre el nombre de un algoritmo de editado indica que se ha utilizado una validación cruzada de 10 bolsas sobre el conjunto de training.
- El sufijo “**_itt**” sobre el nombre de un algoritmo de editado indica que se ha utilizado como conjunto de training *dataset12_R_L-L.arff* y como test *dataset12_T_L-L.arff* (itt = independent training and test).
- El algoritmo VSM fue aplicado con K=3. El algoritmo CFS de Weka fue utilizado junto a BestFirst con la configuración por defecto de Weka 3.6.6, usando tanto “Use full training set” como “Cross-validation”.

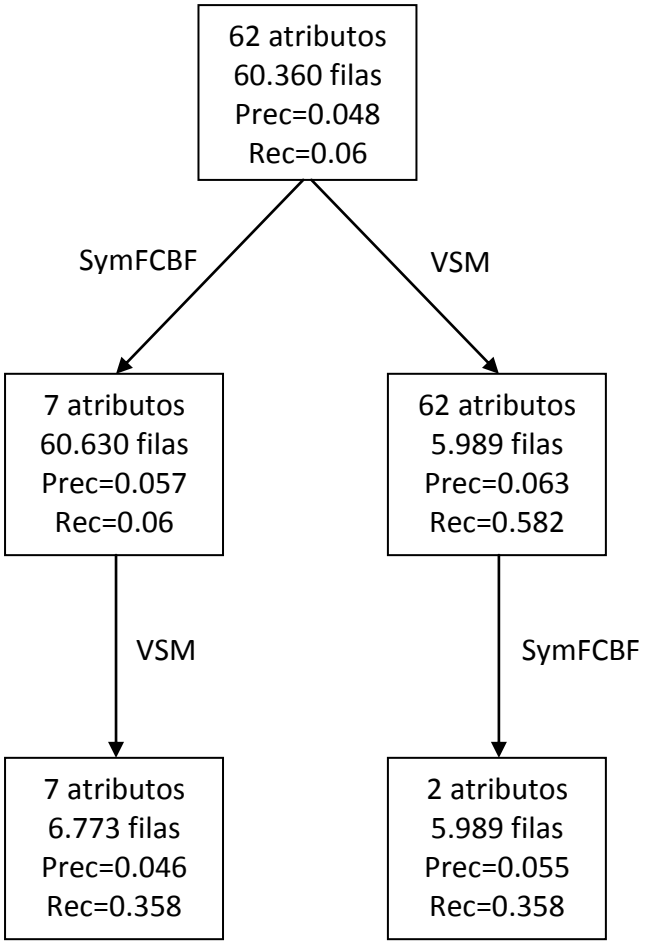
Conclusiones:

1. Si se aplica CFS tanto sobre VSM_xxx, el subconjunto de atributos que devuelve es el atributo 1 (length). Ya conozco bien ese atributo y sé que sólo es muy útil cuando dicho

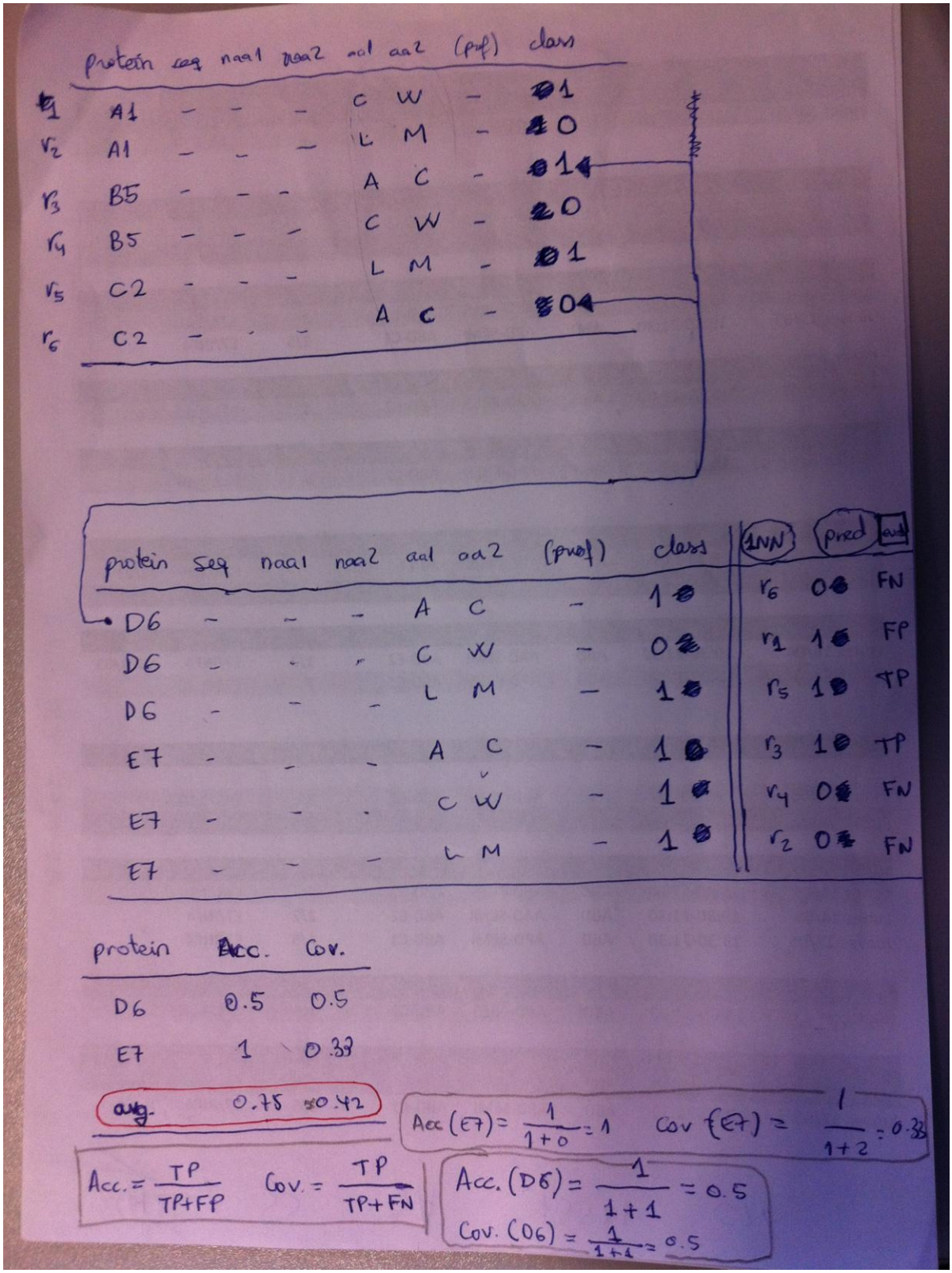
atributo toma valores en el intervalo [0,0.025). Y en este intervalo las predicciones son "fáciles" y no sirven para competir.

2. Por el motivo expresado en la conclusión 1, he probado CFS sobre el dataset eliminando los ejemplos con length en el intervalo [0,0.025). De esta forma, podría comprobar cómo de útil es dicho atributo en las predicciones "difíciles" y competitivas. El resultado se expresa en la tabla como CFS_1 y produjo los atributos 1 (otra vez length) y 9. No obstante, también se probó a predecir únicamente con el atributo 1 y el resultado se expresa con CFS_2.
3. Se ha probado a eliminar el atributo 1 de la entrada para el algoritmo de selección de atributos, para conocer la bondad del resto de atributos. El resultado se expresa con CFS_3 y CFS_4. CFS_3 usando "Use full training set" y CFS_4 usando "Cross-validation".
4. Keel genera el mismo dataset reducido con "_stt" y con "_itt". ¿? Curioso.
5. Un experimento de editado de Keel utiliza un dataset de training y otro de test en la carpeta "datasets" y produce otro de training reducido y el mismo de test (aunque tenga diferente tamaño, es por los decimales) en la carpeta "<nombre alg>.dataset<xxx>".

Árbol de resultados según algoritmo de editado y de selección de atributos: SymFCBF (cv10f) y VSM (k=3)



Ejemplo de predicción y evaluación



Referencias

[1] Monastyrskyy, B.; Fidelis, K.; Tramontano, A. & Kryshtafovych, A. Evaluation of residue-residue contact predictions in CASP9. *Proteins, Genome Center, University of California-Davis, 451 Health Sciences Drive, Davis, CA 95616, USA,* **2011**, *79 Suppl 10*, 119-125