

Estudi i millores de Relief

Rafael Castillo
rafaelc@lsi.upc.edu

Gabriel Prat
gprat@lsi.upc.edu

19 de gener de 2004

1 Introducció

Pretenem fer un estudi d'aquest mètode de selecció d'atributs presentat per Kira i Rendell [1], partint de les millores aportades per Kononenko [2] i incorporant algunes noves idees.

L'estudi es centra bàsicament en dos punts: el primer seria provar de variar la mètrica usada per l'algorisme en les variables categòriques emprant la mètrica VDM introduïda a [4]; el segon, veure els problemes de l'algorisme amb els atributs redundants i intentar aportar alguna millora.

2 Estudi de la mètrica emprada

Necessitem una mètrica heterogènia per tal de poder treballar indistintament amb variables contínues i categòriques. La mètrica HEOM emprada en l'algorisme original, usa la distància euclídea com a mètrica per a les variables contínues, mentre que per les categòriques empra la coneguda amb el nom de "overlap". Aquesta darrera és tan simple com dir que les que són iguals estan a distància 0 i les que no són iguals a distància 1. Com ja apuntaven Wilson i Martinez [5], aquest tractament tan simplista de la mètrica als atributs nominals no està fent ús de informació addicional que ens aporten aquestes variables i que podria ésser útil en la generalització.

Per solucionar aquest problema emprarem la VDM (Value Difference Metric). Aquesta mètrica es basa en les freqüències d'aparició de cada valor d'un atribut per cada classe. Tal i com ho proposa a [4] la distància VDM entre dos individus i, j és:

$$D_{VDM}(i, j) = \sum_{a \in A} d_{VDM}(i_a, j_a, a) w(a, i_a) \quad (1)$$

On i_a és el valor de l'atribut a en l'individu i .

$$d_{VDM}(x, y, a) = \sum_{c \in C} \left(\frac{N_{a=x,c}}{N_{a=x}} - \frac{N_{a=y,c}}{N_{a=y}} \right)^2 \quad (2)$$

$$w(a, x) = \sqrt{\sum_{c \in C} \left(\frac{N_{a=x,c}}{N_{a=x}} \right)^2} \quad (3)$$

On $N_{a=x,c}$ és el número d'individus de la classe c que tenen x com a valor de l'atribut a i $N_{a=x}$ és el número d'individus que tenen x com a valor de l'atribut a .

També podem escriure aquestes fórmules d'una forma més simple veient-les com a probabilitats condicionades:

$$d_{VDM}(x, y, a) = \sum_{c \in C} [P(c|x_a) - P(c|y_a)]^2 \quad (4)$$

$$w(a, x) = \sqrt{\sum_{c \in C} [P(c|x_a)]^2} \quad (5)$$

On $P(c|x_a)$ és la probabilitat condicionada de que un individu sigui de la classe c sabent que el seu atribut a té un valor x .

El factor $w(a, x)$ és el pes d'un atribut i intenta aportar informació sobre la capacitat de discriminació d'aquest atribut. El valor mínim d'aquest atribut representa una distribució uniforme de les classes on cada valor de l'atribut apareix amb igual probabilitat per cada classe i podem veure que prendrà el valor:

$$w(a, x) = \sqrt{\sum_{c \in C} \frac{1}{|C|^2}} = \sqrt{\frac{|C|}{|C|^2}} = \sqrt{\frac{1}{|C|}} = |C|^{-1/2} \quad (6)$$

I que prendrà el seu valor màxim quan a sigui un discriminador perfecte i per tant aquest valor només

aparegui en una classe. A més podem veure fàcilment com aquest valor màxim és 1. Així doncs, aquest pes ens dóna una idea de fins a quin punt la distribució de probabilitats d'un atribut respecte la classe està esbiaixada o és uniforme.

Aquesta mètrica, però té també algun problema. Com hem vist, no té en compte la semblança dels atributs dels individus per calcular la distància sinó la distribució de probabilitats condicionades en funció de la classe. Això ens porta a que dos atributs amb igual distribució de probabilitats en funció de la classe estiguin a distància 0 segons aquesta mètrica, cosa que pot ser interessant en alguns casos però que en d'altres no ho és en absolut. En especial no és interessant en els problemes on la distribució d'alguns atributs respecte la classe és poc esbiaixada però la seva combinació sí que té un biaix gran.

Podem trobar un bon exemple en el problema de la paritat (parity-n) on tenim n atributs que poden prendre els valors 1 o 0 i la seva classe és 1 només quan el nombre d'atributs amb valor 1 és parell i 0 altrament. Cada valor de cada atribut té una distribució uniforme respecte la classe, però la combinació de tots ells ens permet predir exactament la classe. En aquest cas la distància calculada per la VDM entre dos individus qualsevol serà sempre 0 donat que totes les probabilitats condicionades prendran el mateix valor, de manera que no tindrem cap informació sobre quin individu escollir com a veí més proper.

A més, si afegim atributs irrelevantes distribuïts uniformement, aquests també tindran la mateixa distribució en les probabilitats i per tant la diferència entre ells també serà 0 i per tant el Relief els assignarà exactament el mateix pes que a als rellevants. Això doncs ens fa pensar que per alguns problemes on els atributs siguin molt dependents entre ells no serà bona idea utilitzar HVDM.

3 Estudi de la redundància

Relief ens dóna una mesura de rellevància de les variables, però què passa amb la redundància? Ens podem fer moltes preguntes interessants al respecte, però potser les primeres a respondre són les següents:

1. Donarà el mateix pes a dues variables si són redundants entre elles?
2. Si dues variables tenen el mateix pes, podem concloure que són redundants?

3. Les variables redundants es perjudiquen entre sí?

Intentarem respondre i justificar les preguntes fetes començant per l'algorisme original, i després intentarem tornar a respondre-les amb algunes modificacions sobre aquest (que explicarem després juntament amb la seva motivació).

Suposem doncs que tenim dos atributs A_1 i A_2 que són redundants, entenent que dos atributs són redundants si ens permeten determinar la classe de l'individu no només en el mateix nombre de casos sinó també en els mateixos casos (en els problemes reals això no succeirà completament entre cap parella d'atributs, pel que podem pensar més aviat en la norma donada com una mesura de redundància entre atributs).

Per respondre la pregunta 1 partirem d'un problema amb els dos atributs esmentats i un tercer (al que anomenarem A_3) que no és redundant als dos primers. Tan sols cal demostrar que la diferència entre el "nearest hit" i el "nearest miss" i ell és la mateixa per A_1 i A_2 , ja que si això és cert és evident que l'actualització a cada volta serà la mateixa per tots dos i per tant acabaran tenint el mateix pes.

Primerament farem aquesta demostració tan sols per dues classes, amb atributs binaris i suposant que no hi ha soroll i després parlarem de què passa en el cas més general. En el cas plantejat tenim els següents valors:

Cas 1: $A_1 = X_1 \quad A_2 = X_1 \quad A_3 = Y_1 \quad C = Z_1$

Cas 2: $A_1 = X_1 \quad A_2 = \neg X_1 \quad A_3 = Y_1 \quad C = Z_1$

Ja que altrament les variables no ens servirien per determinar la classe en els mateixos casos. El "nearest hit" i el "nearest miss" són de la forma:

Cas 1: $A_1 = X'_1 \quad A_2 = \neg X'_1 \quad A_3 = Y'_1 \quad C = Z'_1$

On Z'_1 serà igual a Z_1 pel "nearest hit" i igual a $\neg Z_1$ pel "nearest miss". Si X'_1 és diferent de X_1 llavors la diferència als dos atributs serà 1 i si és igual serà 0.

Cas 2: $A_1 = X'_1 \quad A_2 = \neg X'_1 \quad A_3 = Y'_1 \quad C = Z'_1$

Si X'_1 és diferent de X_1 llavors $\neg X'_1$ també serà diferent de $\neg X_1$ i la diferència serà 1, mentre que en el cas contrari ambdues valdran 0.

La extensió a variables no binàries es bastant simple: els atributs tindran més possibles valors, però si

al “nearest hit” i al “nearest miss” A_1 té el mateix valor, forçosament A_2 també haurà de tenir-lo per complir amb la hipòtesi de la redundància. Passa el mateix en el cas que el valors siguin diferents. Si pensem en problemes amb més de dues classes (extensió del Relief proposada per Kononenko), tot el que hem dit fins ara es manté: a cadascun dels “nearest misses” succeeix que es manté la relació entre l’atribut original i l’afegit redundant a ell, fet que fa que l’increment també acabi essent el mateix.

Anem a parlar de què varia si hi ha una certa probabilitat (que anomenarem ps) de que les dues variables no siguin redundants en alguns valors per culpa del soroll. Més concretament anem a veure la probabilitat amb què aquest soroll farà variar la diferència d’una certa instància I_1 amb el seu “nearest hit” i el seu “nearest miss” d’un atribut A a una rèplica seva afectada amb una probabilitat ps que anomenarem A' (de fet, la fórmula següent serveix en general per veure el canvi de diferència amb qualsevol altre instància I_2):

$$pcd_{A-A'} = \frac{p(V_1 = V_2) \wedge p(V'_1 \neq V'_2) + p(V_1 \neq V_2) \wedge p(V'_1 = V'_2)}{p(V_1 \neq V_2) \wedge p(V'_1 = V'_2)} \quad (7)$$

On $pcd_{A-A'}$ és la probabilitat del canvi de diferència a les instàncies I_1 i I_2 entre els atribut A i A' , V_1 i V_2 són els valors de I_1 i I_2 respectivament per aquest atribut i V'_1 i V'_2 són els valors de l’atribut A' . Com que la relació entre els valors de l’atribut A i els valors de l’atribut A' no són independents entre sí, podem rescriure la fórmula:

$$pcd_{A-A'} = \frac{p(V'_1 = V'_2 | V_1 \neq V_2) p(V_1 \neq V_2) + p(V'_1 \neq V'_2 | V_1 = V_2) p(V_1 = V_2)}{p(V_1 \neq V_2) p(V_1 = V_2)} \quad (8)$$

Seguint amb el desenvolupament de la fórmula, quina és la probabilitat de que V'_1 sigui igual a V'_2 sabent que V_1 i V_2 eren diferents? Distingim varis casos: si V'_1 ha canviat però V'_2 no, tenim que la probabilitat de que siguin iguals sabent que abans no ho eren és d’una entre el conjunt de valors possibles que podria prendre menys un (el que tenia). Si qui ha canviat és V'_2 però V'_1 s’ha quedat igual passaria el mateix. Si tots dos canvien llavors la probabilitat de que tinguin el mateix valor és igual a la probabilitat que el primer en canviar (suposem que ha estat V'_1) no hagi pres el valor que tenia l’altre (en el cas explicat, V'_2), ja que de fer-ho seria impossible que el segon fos igual per la hipòtesi de que tots dos canvien, i de que l’altre passi a prendre el mateix valor,

cosa que farà novament en un d’entre els possibles $|V| - 1$ casos que pot triar (tots menys el valor que tenia). Com la probabilitat de que un valor hagi canviat és ps ja tindriem desglossada la primera part del sumatori. Ara la pregunta és, quina és la probabilitat de que V'_1 sigui diferent a V'_2 sabent que V_1 i V_2 eren iguals? aquest cas és més simple: si tan sols varia una d’elles llavors segur que la diferència varia. Si en canvi varien les dues el que hem de veure és la probabilitat de que les dues no hagin pres el mateix valor, que es pot comprovar que és igual a $|V| - 2$ dividit entre $|V| - 1$, doncs un cop a triat el valor per la primera aquesta serà la proporció de casos en què la segona serà diferent. Així doncs, podem rescriure la fórmula:

$$pcd_{A-A'} = \frac{2(ps(1-ps))}{|V|-1} + \frac{ps^2(|V|-2)}{(|V|-1)^2} p(V_1 \neq V_2) + \frac{2(ps(1-ps)) + ps^2(|V|-2)}{(|V|-1)} p(V_1 = V_2) \quad (9)$$

Un cas particular d’aquesta fórmula el tenim quan $|V|$ és igual a 2 (atributs binaris). En aquest cas, els dos termes que multipliquen per una banda a la probabilitat de que V_1 i V_2 siguin diferents i per l’altre a la probabilitat de que siguin iguals passen a ser el mateix, pel que podem treure factor comú i com les dues probabilitats sumen 1 (donats dos valors V_1 i V_2 o bé són iguals o bé són diferents), ens queda quelcom tan simple com:

$$pcd_{A-A'} = \frac{2(ps(1-ps))}{(2ps - 2ps^2)} = \quad (10)$$

Anem a veure ara que passa amb l’increment dels pesos. Per simplificar, direm X al factor que està multiplicant la probabilitat de que V_1 i V_2 siguin diferents i Y al factor que multiplica a la de que siguin iguals. Tenim:

$$pcw_{A-A'} = X \cdot p(V_1 \neq V_2)_{NM} + Y \cdot p(V_1 = V_2)_{NM} - (X \cdot p(V_1 \neq V_2)_{NH} + Y \cdot p(V_1 = V_2)_{NH}) \quad (11)$$

En aquesta fórmula simplement hem aplicat les idees presentades a les anteriors a la fórmula que ens serveix per incrementar els pesos: el “nearest miss” té una influència amb el mateix signe de la diferència (interessa que els veïns d’altres classes estiguin lluny)

mentre que al “nearest hit” succeeix el contrari. Simplificant (la probabilitat de que dues variables siguin iguals és 1 menys la probabilitat de que siguin diferents) i traient factor comú:

$$pcw_{A-A'} = X(p(V_1 \neq V_2)_{NM} - p(V_1 \neq V_2)_{NH}) + Y(p(V_1 \neq V_2)_{NH} - p(V_1 \neq V_2)_{NM}) \quad (12)$$

Una primera observació és que aquesta fórmula, pel cas particular presentat abans on els atributs que tractem són binaris ($|V| = 2$), com $X=Y$ i els altres termes es contraresten, ens diu que el soroll afegit no afectarà al pes. Per altres valors de $|V|$ ens recolzarem en una gràfica que ens permetrà esbrinar la importància de cada terme:

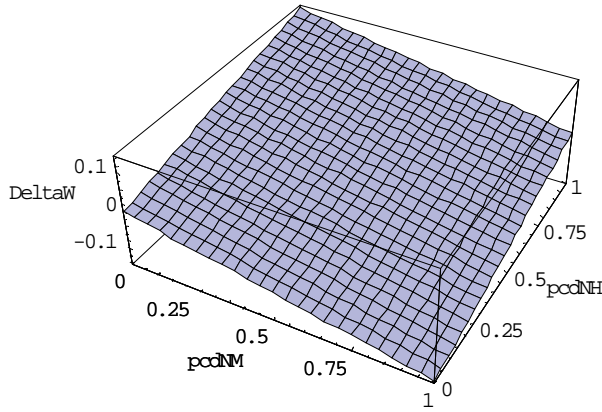


Figura 1: $ps = 0.1$, $|V| = 3$

Veiem que a mesura que s’incrementa la relació entre les probabilitats de canvi entre els valors de l’atribut original i la seva rèplica en el “nearest hit” i el “nearest miss”, també s’incrementa la probabilitat de canvi del pes. Això farà que el soroll afecti més a les variables on aquesta relació és gran, és a dir, per un cantó a aquelles on la probabilitat de que el valor dels dos atributs al “nearest hit” sigui diferent sigui alta mentre que al “nearest miss” sigui baixa (per tant a les variables més rellevants), i per l’altre a les variables on succeeixi el contrari (per tant a les variables més irrelevantes, on tan sols hi havia contribucions negatives per culpa dels “nearest hits”).

La conclusió de tot aquest desenvolupament és doncs que el soroll tendirà a fer que els pesos

s’apropin a zero, incrementant més aquells que estaven per sota i decremantant els que estaven per sobre. Anem a veure ara quin és doncs el paper de la probabilitat de que hi hagi soroll (a les fórmules presentades, ps).

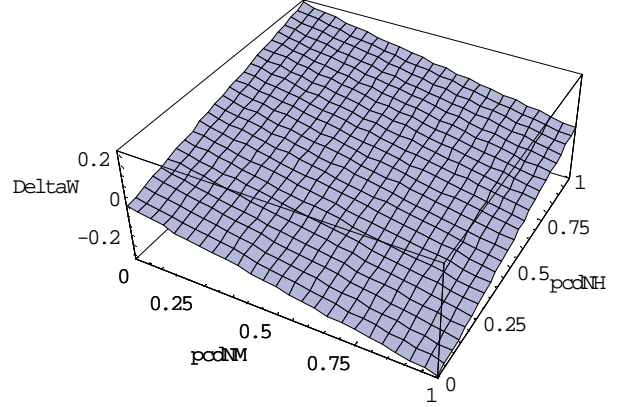


Figura 2: $ps = 0.2$, $|V| = 3$

Pel que fa a l’efecte de ps , veiem com sembla ser que simplement ens està controlant el rang de la probabilitat de que l’increment de pes variï.

Un cop acabat l’estudi sobre la relació entre els pesos d’atributs redundants, anem a seguir amb la implicació en sentit contrari que ens plantejavem a la pregunta 2: quan dues variables tenen el mateix pes, podem pensar que el tenen perquè són redundants?

Doncs bé, sembla bastant clar que el suggerit no serà cert. Com hem dit abans el Relief ens dona una mesura de la rellevància de l’atribut, per la qual cosa si acceptem que funciona bé també hem d’acceptar que dos atributs igualment rellevants i no redundants (serveixen per determinar la classe de l’individu en el mateix nombre de casos però en casos completament disjunts) rebin el mateix pes. En aquest cas amb un contraexemple que mostri aquest fet ens valdrà, i un problema molt simple en el que es dona aquest fet és XOR (o la seva extensió parity- n): en aquest problema els “nearest hits” sempre restaran (els individus de la mateixa classe tenen valor diferent en tots els atributs), i els “nearest misses” sumaran o restaran en funció de com fem la tria en cas d’empat. Si ho fem aleatòriament al límit és fàcil veure que l’aportació dels “nearest misses” serà també la mateixa per tots dos atributs, amb el que tots dos acabaran amb el mateix pes.

Anem finalment a donar resposta a la pregunta 3, on se'ns plantejava si la redundància estava afectant al rendiment de l'algorisme. Podríem veure primer què passa amb els pesos que dona Relief als atributs abans i després d'afegir-ne de redundants a un d'ells, i després explicar el motiu del canvi. Podríem partir per tant del cas emprat a 1, primer només amb A_2 i A_3 i després afegint A_1 . El que observem empíricament és que el pes de A_2 ha baixat (responent a la pregunta 1 ja hem vist que A_1 serà igual a A_2). El motiu és que per l'efecte d'afegir l'atribut redundant pot haver-hi un canvi al “nearest miss”, i si aquest canvi afecta a la variació de la diferència llavors ho fa negativament:

Imaginem que tenim un individu R_1 i que el seu “nearest miss” està a distància de hamming d d'ell. Si a l'atribut A_1 ambdós tenen el mateix valor, llavors no pot ser que afegint A_2 canviem de veí (ja que la distància entre els dos es manté i com era el veí més proper de la mateixa classe ho ha de seguir sent). En canvi si els dos tenien diferent valor a l'atribut A_1 , com a molt a l'afegir A_2 ens pot passar que un altre individu R_2 passi a estar a la mateixa distància ($d+1$) que R_1 (en el cas que abans el “nearest miss” fos l'únic a distància d) o bé a distància menor que R_1 (en el cas que abans el “nearest miss” no fos l'únic a distància d), però el que és segur és que el nou “nearest miss” tindrà el mateix valor a l'atribut A_1 i que per tant aquest fet tindrà una influència negativa en el pes.

De igual forma és cert que al “nearest hit”, pas-sarà quelcom contrari:

Imaginem novament que tenim un individu R_1 i que el seu “nearest hit” està a distància de hamming d d'ell. Si a l'atribut A_1 els dos tenen el mateix valor, llavors no pot ser que afegint A_2 canviem de veí (pel mateix motiu que abans). En canvi, si els dos tenien diferent valor a l'atribut A_1 , com a molt l'afegir A_2 ens pot passar que un altre individu R_2 passi a estar a la mateixa distància ($d+1$) que R_1 (en el cas que abans el “nearest hit” fos l'únic a distància d) o bé a distància menor que R_1 (en el cas que abans el “nearest hit” no fos l'únic a distància d), però el que és segur és que el nou “nearest hit” tindrà el mateix valor a l'atribut A_1 i que per tant aquest fet tindrà una influència positiva en el pes.

Vist això es podria pensar que ambdós influències es contraresten, però és evident que a mesura que un atribut és rellevant les diferències amb els veïns de la seva classe haurien de tendir a 0 i les diferències amb els veïns de les altres classes a 1, i aquest fet

provoca el que podríem anomenar un “efecte sostre” a la influència dels “nearest hits”: per molt que anem apropant les instàncies dels veïns més propers de la meua classe en la direcció de l'atribut que hem “replacat” no hi ha variació en la diferència, ja que els veïns no em penalitzaven abans d'afegir-lo. En canvi apropar els veïns de les altres classes en aquesta direcció, que abans no em penalitzaven per tenir valors diferents, farà que ara si que em penalitzin. Igualment aquest “efecte sostre” passarà a la influència dels “nearest misses” pels atributs irrelevants, ara però amb un efecte contrari. El resultat és que a mesura que afegim variables redundants a una de les existents, el seu pes s'anirà apropant a 0, tal i com ja avançaven M. Robnik-Šikonja, I. Kononenko a [3].

4 Competència de pesos

En aquesta secció introduïm una normalització de l'increment dels pesos per tal de fer-los competir, o sigui fer que l'increment d'un cert pes impliqui la disminució dels altres i veure si aquesta modificació té algun impacte sobre les qüestions plantejades a la secció anterior i en el cas que el tingui si aplicar aquesta modificació ens ajuda a detectar les variables redundants.

Seguint el mateix esquema que abans, primer comprovarem si ara l'algorisme assigna el mateix pes a les variables redundants o si la competició les afecta i els seus pesos varien. Com abans, estudiarem l'efecte que pugui tenir l'actualització del pes que fa Relief per cada individu que ara podem pensar que s'assemblarà a:

$$w_a(t+1) = w_a(t) + \frac{\Delta w_a(t)}{\sum_{a \in A} \Delta w_a(t)} \quad (13)$$

On A representa el conjunt de tots els atributs i $w_a(t)$ és el pes de l'atribut a en l'instant de temps t i $\Delta w_a(t)$ el seu increment en el mateix instant. Aquesta fórmula, però ens presenta un problema que és que com que $\Delta w_a(t)$ està entre -1 i 1 el sumatori podria tenir un valor negatiu i fer canviar el signe de l'increment. Per tornar a un increment entre 0 i 1 fem:

$$w_a(t+1) = w_a(t) + \Delta' w_a(t) \quad (14)$$

$$\Delta' w_a(t) = 2 \left(\frac{\Delta w_a(t) + 1}{\sum_{a \in A} \Delta w_a(t) + |A|} \right) - 1 \quad (15)$$

Així doncs si $\Delta w_a(t)$ estava entre 0 i 1 $\Delta'w_a(t)$ també ho estarà i continuarem mantenint les propietats de l'algorisme.

Estudiem doncs què passa ara amb els $\Delta'w_a(t)$ de dues variables redundants. Podem veure que si $\Delta'w_{a1}(t) = \Delta'w_{a2}(t)$, $\forall t$ llavors com que els pesos inicials són iguals, el pes final dels dos atributs serà el mateix, com hem vist que passava si no normalitzàvem. També és fàcil adonar-nos que aquesta condició es complirà mirant els termes de $\Delta'w_a(t)$. Està clar que els factors constants que multipliquen i resten no canviaran, a part d'això, el denominador de la divisió és comú per tots els atributs i al numerador hi trobem $\Delta w_a(t)$ que hem demostrat igual per dues variables redundants en la secció anterior.

Per tant amb aquesta competició d'increments es continua complint que si dos atributs són redundants llavors l'algorisme els assigna el mateix pes.

Per la pregunta 2, continuem tenint el mateix problema que si no estiguéssim fent competir els pesos. Si dues variables són igual de rellevants però en casos diferents (com en el cas de XOR o parity-n) llavors es faran les actualitzacions de manera anàloga al cas sense competició i la única cosa que variarà serà el valor final del pes, però també acabaran tots els atributs igual de rellevants amb el mateix pes.

Pel què fa a la pregunta 3, també podem veure com la resposta no canviarà donat que la relació entre les actualitzacions de pesos de les variables no canvia donat que només hem afegit termes que són constants per tots els atributs i per tant no n'estem perjudicant ni potenciant cap respecte els altres en cap actualització de pesos. Així doncs les mateixes premisses que hem utilitzat abans ens són vàlides per dir que amb la introducció d'una variable redundante a una altra provoquem que els pesos de les dues variables s'acostin més al 0 que el pes de la variable inicial. Així doncs, la competició de pesos tampoc ha aconseguit que la introducció de variables redundants no modifiqués el pes de la variable original abans de introduir-les.

5 Conclusions

Pel què fa a l'estudi de la mètrica, caldria estudiar millor les propietats de la HVDM per intentar predir en quins problemes ens interessarà aplicar-la i en quins no. Sembla clar que en problemes amb atributs dicotòmics aquesta mesura no és tan interessant donat que intenta donar distàncies diferents entre els diferents valors d'un atribut, però està clar

que si només en tenim dos la distància serà sempre 0 o un altre valor entre 0 i 1 però no ens està aportant molta més informació.

A més les proves realitzades han estat amb conjunts de dades sintètics, tots amb atributs binaris i molts amb distribucions de probabilitat de cada valor de cada atribut uniformes en respecte a la classe. En definitiva, sembla ser que els pitjors problemes per la VDM. Així doncs només podem dir que caldria un estudi més intensiu i més exhaustiu de la seva utilitat.

Les preguntes a les respostes plantejades pel que fa referència a la redundància, ens porten a la conclusió que caldria trobar alguna forma per tal de fer més robust aquest algorisme enfront aquest problema : a la resposta a 1 ja hem vist la forma amb què el soroll afecta als pesos, negativa pels nostres interessos; a 2 hem vist com el fet de tenir pesos iguals (malgrat no haver soroll) no significa que les variables fossin redundants, fet que juntament amb l'anterior ens fa veure que sense cap modificació no tenim forma de detectar aquestes relacions; a 3 hem vist com, a més a més, aquesta redundància també afecta negativament al rendiment de l'algorisme (com passava amb el soroll a 1, a mesura que afegim redundància anem fent caure els pesos a 0, potenciant els atributs més irrelevantes i perjudicant als rellevants i en definitiva empobrint el rendiment de l'algorisme.

Per solucionar això hem intentat fer competir els pesos per intentar que algunes, idealment totes excepte una, de les variables redundants prenguin valors baixos mentre que d'altres prenguin els valors reals. De totes maneres, la proposta d'aquest document ha demostrat ser bastant innòcua al funcionament de l'algorisme. De fet a [3] ja apuntaven que els pesos ja s'estaven normalitzant implícitament en certa manera en el funcionament normal de l'algorisme cosa que ja fa pensar que la normalització dels increments no aporta massa res.

6 Treball futur

Vistes les idees aplicades en altres algorismes com en algorismes de "lazy learning" o com en la mateixa VDM de ponderació de diferències entre atributs a l'hora de calcular les distàncies en funció d'un pes proporcional a l'estimació de la rellevància d'aquell atribut, i partint de la base que el nostre algorisme intenta predir justament aquesta rellevància, podríem utilitzar el càlcul parcial d'aquesta magnitud que obtenim al tractar cada individu per ponderar el càlcul de distàncies en els individus restants.

Podríem pensar també en algun factor de temperatura que faci que en les primeres iteracions, assumint que el càlcul parcial dels pesos és poc fiable, el tingui menys en compte a l'hora de ponderar les distàncies.

Un aspecte clar a treballar seria millorar la robustesa de l'algorisme a la redundància. Ja hem vist que l'algorisme té dos problemes bàsics amb la redundància. Es fa difícil de detectar quan hi ha soroll entre els atributs redundants donat que l'algorisme els assigna pesos semblants però no prou, i a més assigna pesos iguals a atributs no redundants. I per altra banda, l'existència d'atributs redundants a un altre que és rellevant fa que tots ells obtinguin una ponderació no tan elevada com els pertocaria si només en tinguéssim un. Així doncs, seria molt interessant trobar alguna modificació de l'algorisme per tal de tractar millor la redundància.

Inicialment es podria intentar solucionar pels casos de dos atributs redundants entre sí, cosa que a priori sembla més senzilla. A més, hem de tenir en compte que Relief tracta els atributs individualment i, per tant, sembla complicat modificar-lo de manera que acabi detectant redundàncies complexes d'un atribut amb un conjunt d'atributs a no ser que es faci alguna modificació important a l'algorisme per tal de treballar ja no amb un atribut individual sinó en algun cas amb un conjunt d'atributs.

També seria interessant continuar estudiant l'efecte que té sobre l'algorisme la utilització de mètriques diferents a l'hora de calcular les distàncies entre individus o les diferències entre els valors dels atributs. Per començar, acabar d'estudiar la incorporació de la VDM i més enllà d'això buscar altres mètriques que ens puguin ser útils o fins i tot veure si pot ser interessant utilitzar-ne una per calcular la distància i una altra per la diferència que tinguin en compte aspectes diferents com és el cas de HEOM i

HVDM.

Una altra línia de treball podria ser donar més suport empíric a les aportacions teòriques d'aquest document. Les proves realitzades com hem esmentat anteriorment eren totes amb conjunts de dades sintètics que, tot i aportar molta informació donat el nostre extens coneixement sobre el domini i dels resultats esperats, poden tenir algun aspecte molt exagerat o allunyat dels valors habituals en problemes reals i donar-nos una visió excessivament optimista (o pessimista) dels resultats obtinguts. Això sembla clar en el cas de la VDM, que amb les proves realitzades obtenia sempre resultats pitjors però sobre el paper i també en la bibliografia sembla ser prou útil.

Referències

- [1] Kenji Kira and Larry A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, pages 129–134, 1992.
- [2] Igor Kononenko. Estimating attributes: Analysis and extensions of relief. In Francesco Bergadano and Luc De Raedt, editors, *ECML*, volume 784 of *Lecture Notes in Computer Science*, pages 171–182. Springer, 1994.
- [3] Marko Robnik-Sikonja and Igor Kononenko. Theoretical and empirical analysis of relief and rrelief. *Machine Learning*, 53(1-2):23–69, 2003.
- [4] Craig Stanfill and David L. Waltz. Toward memory-based reasoning. *Commun. ACM*, 29(12):1213–1228, 1986.
- [5] D. Randall Wilson and Tony R. Martinez. Improved heterogeneous distance functions. *J. Artif. Intell. Res. (JAIR)*, 6:1–34, 1997.