Identificació de nombres escrits
Mineria de Dades

Miquel Perelló Nieto, Marc Bergés Garrido

15 de juny de 2012

Índex

Índex de figures

Capítol 1

Resum inicial

En aquest projecte ens trobem amb el problema de classificar dígits manuscrits del 0 al 9 provinents de l'escaneig automàtic de sobres del servei postal dels Estats Units. Cada un d'aquests dígits a classificar ha estat normalitzat resultant en una imatge en escala de grisos de 16 x 16 píxels.

Cada línia de les dades consisteix en el dígit identificat seguit per 256 valors d'escala de grisos (del -1 a l'1). Tot seguit es poden observar alguns exemples de la representació d'alguns dígits:

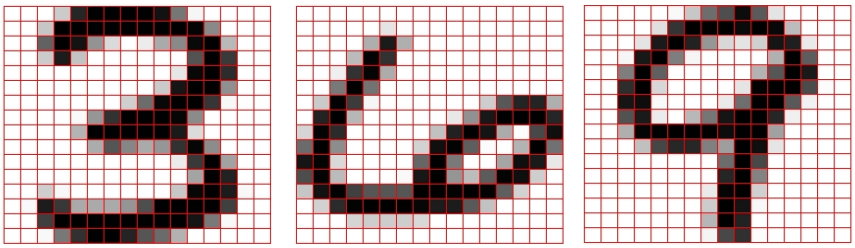


Figura 1.1: Variables explicatives inicials.

En total tenim 9298 observacions separades en dos grups; un d'entrenament amb 7291 d'aquestes observacions i un de prova amb 2007. Aquestes estan distribuïdes de la següent manera:

freq.	0	1	2	3	4	5	6	7	8	9	Total
Ent.	1194	1005	731	658	652	556	664	645	542	644	7291
Prov.	359	264	198	166	200	160	170	147	166	177	2007
%	0	1	2	3	4	5	6	7	8	9	
Entr.	0.16	0.14	0.1	0.09	0.09	0.08	0.09	0.09	0.07	0.09	
Prov.	0.18	0.13	0.1	0.08	0.10	0.08	0.08	0.07	0.08	0.09	

Segons se'ns informa, el conjunt de prova és complicat i arribar a assolir un rati d'error del 2,5% ja és considerat excel·lent.

Així doncs, aquest serà el nostre objectiu. Buscarem un model que ens pugui acostar a aquest error i l'intentarem afinar el màxim possible per obtenir els millors resultats. En aquest problema específic prioritzarem el fet d'obtenir el menor error possible davant el de reduïr la dimensionalitat ja que, pel servei postal dels Estats Units, un petit increment en l'error pot suposar un gran nombre de cartes que no arribin al seu destí correcte.

Primer començarem realitzant una extracció de característiques ja que el nombre de variables és molt elevat i la gran majoria d'elles no és molt significativa de cap de les categories de la variable de resposta. Posteriorment i depenent del nombre de variables extretes, considerarem la possibilitat de fer selecció de característiques per reduïr o no la dimensionalitat tenint en compte que aquest no és el nostre principal objectiu i si interfereix amb el percentatge d'error obtingut serà descartat immediatament. El següent pas ja serà l'elecció dels models a aplicar i la seva prova amb les dades de training. Finalment s'escollirà el millor model i s'aplicarà sobre les dades de test per obtenir l'error "real".

Per concloure la feina, explicarem els resultats i exposarem les conclusions que hem tret d'aquest projecte.

Capítol 2

Preprocesament de les dades

Per cada un dels individus de la mostra tenim 256 variables que composen una matriu de 16 x 16 i que van de -1 a 1 en l'escala de grisos éssent -1 el color blanc i l'1 el color negre.

Hem comprovat que no existeix cap valor mancant a cap dels dos conjunts que se'ns ha donat i tampoc tenim cap outlier ja que tots els valors de les variables van del -1 a l'1. No s'ha cregut necessari posar cap sumari ni gràfica ja que degut al gran nombre de variables de les que disposem no és possible veure res amb claredat i menys en un document com aquest. S'ha observat que hi ha una certa tendència general de certs píxels a ser casi sempre o blancs o negres (per exemple, els del centre solen ser negres i els dels extrems blancs) però tampoc és representatiu de cap categoria ja que normalment és cert per totes elles.

Tot i no detectar outliers a les variables, sí hem pogut veure alguns nombres realment mal escrits que no es poden arribar a identificar i poden suposar un problema a l'hora d'entrenar el model; al ser un nombre molt reduït s'ha optat per eliminar-los del conjunt d'entrenament. Es poden veure alguns exemples:

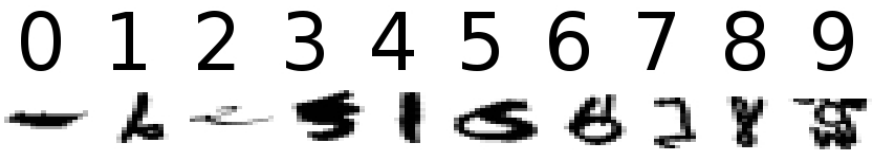


Figura 2.1: Mostra d'individus dificils de classificar .

S'ha intentat veure quines de les variables originals eren més significants per la variable de resposta però ha estat impossible fer-ho utilitzant la F de Fisher ni Chi-quadrat (convertint-les en categòriques) ja que els p-valors obtinguts o bé donen 0 o bé NaN. Un cop vist aquest comportament s'ha procedit a fer la selecció usant filtres tals com el filtre de correlació o el de consistència així com el wrapper "random forest" però tot i d'aquesta manera, cap de les variables obtingudes resulta ser significativa. Això ens ha dut a fer una extracció de

variables que alhora que poden ser més significatives, poden tenir més sentit ja que permeten observar característiques de la imatge en conjunt; no de cada píxel per separat.

2.1 Extracció de característiques

L'extracció de característiques s'ha dut a terme utilitzant el software "MatLab" ja que aquest conté un toolbox específic per extreure informació d'imatges. Per a poder realitzar algunes de les extraccions ha estat necessari categoritzar les variables originals per obtenir una imatge binaritzada dels dígits. Aquesta binarització s'ha dut a terme segons els valors de cada un dels píxels. Com s'ha explicat anteriorment, a les variables originals, cada un dels píxels té un valor de l'escala de grisos que va de -1 a 1 (sent -1 el color blanc i l'1 el color negre). S'ha optat per assignar el color blanc al rang [-1, 0) i el color negre al rang [0, 1]. Tot seguit mostrem alguns exemples d'aquest procés de binarització:

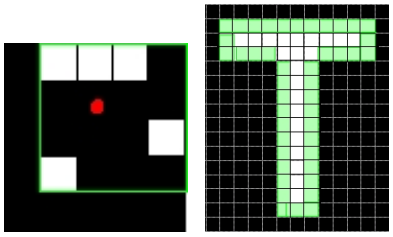
Figura 2.2: Exemple de nombres binaritzats en el nivell de grisos a zeros i uns.

Es pot observar com els resultats mantenen una gran similitud amb les imatges inicials i aquesta acció es pot justificar ja que el que realment ens importa dels dígits és el seu traç.

Com s'ha comentat, aquest procés s'ha dut a terme amb MatLab i s'han aconseguit extreure les següents característiques. Les gràfiques ens mostren la mitjana de la variable i les mitjanes de cada una de les categories. Amb això podem veure si les variables extretes són significatives (quan més a prop es troba una mitjana a la global, menys significativa és).

- **Area**: El nombre actual de píxels a la imatge.
- **Centroid_x**: Centre de gravetat de la imatge en l'eix de les x.
- **Centroid_y**: Centre de gravetat de la imatge en l'eix de les y.
- **ConvexArea**: Nombre de píxels dins del marge convex de la regió.
- **EquivDiameter**: Diàmetre d'un cercle amb la mateixa àrea que la imatge.

- **EulerNumber**: Nombre d'objectes a la imatge menys el nombre de forats d'aquests objectes
- **FilledArea**: Nombre de píxels dibuixats de la imatge.
- **MajorAxisLength**: Llargada en píxels de l'eix més gran de l'el·lipse que té el mateix moment central normalitzat que la imatge.
- **mean2**: Mitjana en la distribució de la matriu de punts.
- **MinorAxisLength**: Llargada en píxels de l'eix més petit de l'el·lipse que té el mateix moment central normalitzat que la imatge.
- **Orientation**: Angle entre l'eix de les x i l'eix més gran de l'el·lipse que té el mateix moment central normalitzat que la imatge.
- **Perimeter**: Perímetre de la imatge.
- **Phasesym**: Simetria de fase.
- **Solidity**: Percentatge de píxels dins el marge convex de la regió.
- **std2**: Desviació estandard de la matriu de punts.



(a) Centroide

(b) Perimetre

(c) Orientació

Figura 2.3: Exemples de l'extracció de característiques

A més, s'ha cregut oportú trobar la manera més comú d'escriure cada un dels dígits de les dades d'entrenament per comparar amb les observacions. Així doncs, s'ha trobat la mitjana de cada un d'ells i aquests en són els resultats:

Amb això, s'ha buscat, altre cop utilitzant MatLab, la distància de cada una de les observacions a la mitjana de la seva categoria i s'han obtingut les variables que expressen les seves similituds. Les gràfiques ens mostren la mitjana de la variable i les mitjanes de cada una de les categories. Amb això podem veure si les variables extretes són significatives (quan més a prop es troba una mitjana a la global, menys significativa és).

Figura 2.4: Mitjana de cada nombre dibuixat a ma en el training.

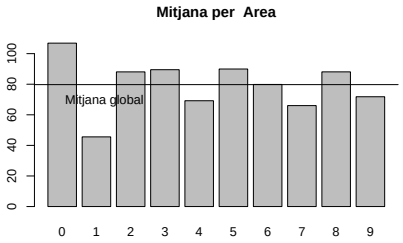
- **sim0**: Distància de l'observació a la manera mitjana d'escriure el zero.
- **sim1**: Distància de l'observació a la manera mitjana d'escriure l'u.
- **sim2**: Distància de l'observació a la manera mitjana d'escriure el dos.
- **sim3**: Distància de l'observació a la manera mitjana d'escriure el tres.
- **sim4**: Distància de l'observació a la manera mitjana d'escriure el quatre.
- **sim5**: Distància de l'observació a la manera mitjana d'escriure el cinc.
- **sim6**: Distància de l'observació a la manera mitjana d'escriure el sis.
- **sim7**: Distància de l'observació a la manera mitjana d'escriure el set.
- **sim8**: Distància de l'observació a la manera mitjana d'escriure el vuit.
- **sim9**: Distància de l'observació a la manera mitjana d'escriure el nou.

Per acabar, s'han extret manualment les següents variables a partir de la binarització:

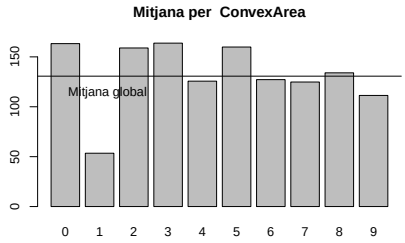
- **suma**: Nombre total de píxels negres.
- **hsymmetry**: Grau de simetria horitzontal.
- **vsymmetry**: Grau de simetria vertical.

2.2 Visualització de les característiques extretes

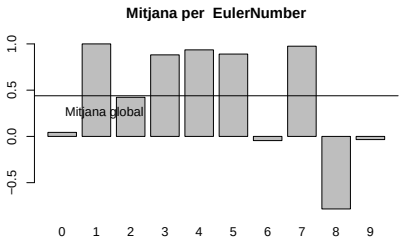
Les gràfiques ens mostren la mitjana global de la variable i les mitjanes de cada una de les categories. Amb això podem veure si les variables extretes són significatives (quan més a prop es troba una mitjana a la global, menys significativa és).



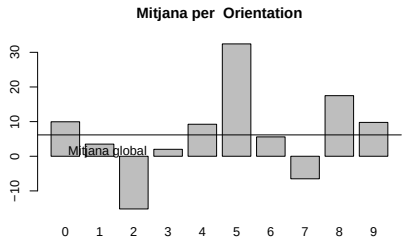
(a) Area



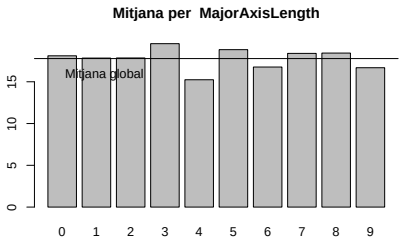
(b) Convex Area



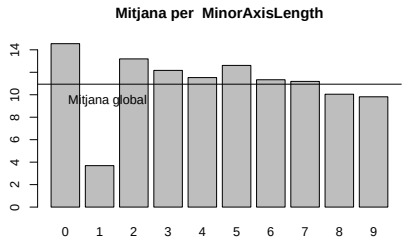
(c) Nombre d'Euler



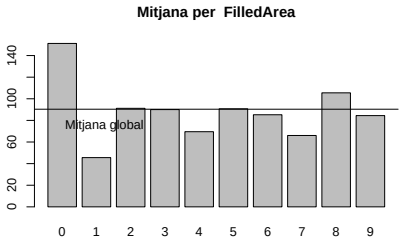
(d) Orientació



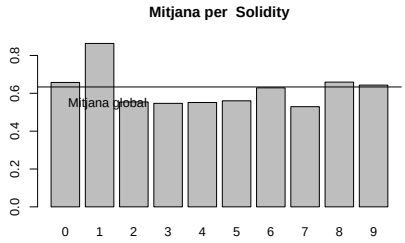
(e) Eix més llarg



(f) Eix més curt

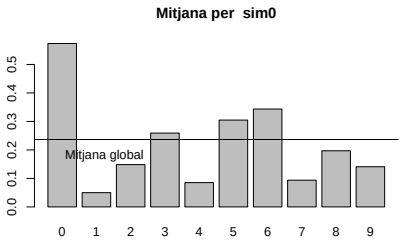


(g) Filled Area

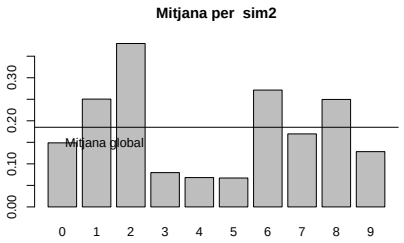


(h) Solidesa

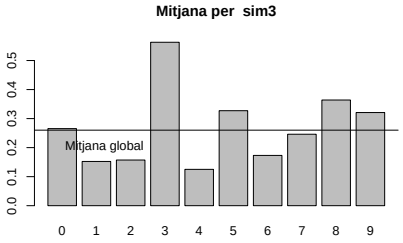
Figura 2.5: Exemples de l'extracció de característiques



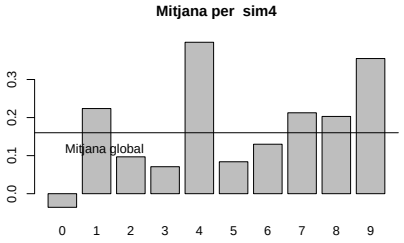
(a) Similitud al zero



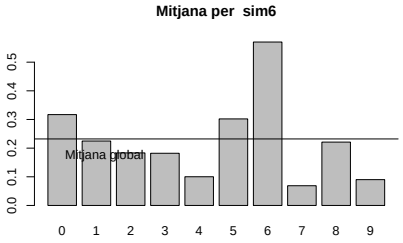
(b) Similitud al 2



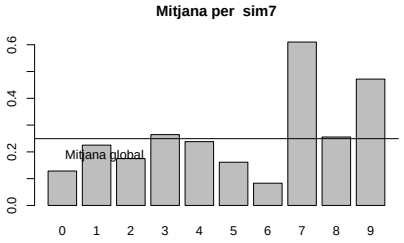
(c) Similitud al 3



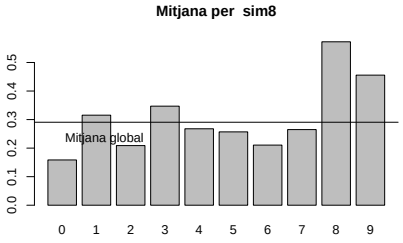
(d) Similitud al 4



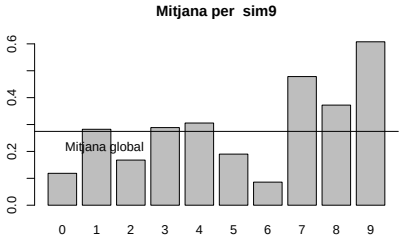
(e) Similitud al 6



(f) Similitud al 7



(g) Similitud al 8



(h) Similitud al 9

Figura 2.6: Exemples de similitud amb la mitjana de diferents nombres separada per classe.

Capítol 3

Selecció de característiques

S'ha provat de dur a terme una selecció d'aquestes característiques extretes mitjançant varis mètodes (filtre de correlació, filtre de consistència i random forest) però no hi ha hagut èxit ja que cada un d'ells ens oferia unes variables diferents i a l'hora de provar els models ens oferien un error més alt que utilitzant-les totes i, com ja s'ha comentat anteriorment, prioritzem el fet d'obtenir el menor error possible davant el de reduïr la dimensionalitat. Així, tot i que exposem els mètodes utilitzats i els resultats, no els tenim en compte a l'hora de crear els models.

En tots els casos s'ha utilitzat ten-fold cross-validation

3.1 Correlation filter

Els resultats obtinguts aplicant aquest filtre han estat:

- Centroid_x
- Centroid_y
- EulerNumber
- MajorAxisLength
- MinorAxisLength
- Perimeter

3.2 Consistency filter

Els resultats obtinguts aplicant aquest filtre han estat:

- Area
- Centroid_x
- Centroid_y
- EulerNumber

- hsymmetry
- MinorAxisLength
- MajorAxisLength
- Perimeter
- suma vsymmetry

3.3 Random Forest

Els 6 resultats més importants obtinguts aplicant aquest filtre han estat:

- Centroid_x
- Centroid_y
- EulerNumber
- MajorAxisLength
- MinorAxisLength
- Perimeter

3.4 Visualització de les més escollides

Tot seguit es mostren unes quantes figures a les quals es pot apreciar que existeixen certes classes que tenen una clara tendència a separar-se de la mitja.

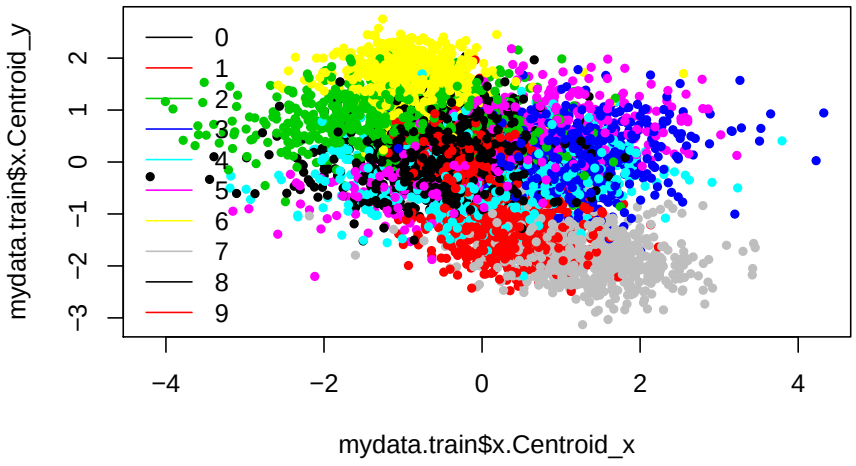


Figura 3.1: Exemple de la distribució per classes amb centroides.

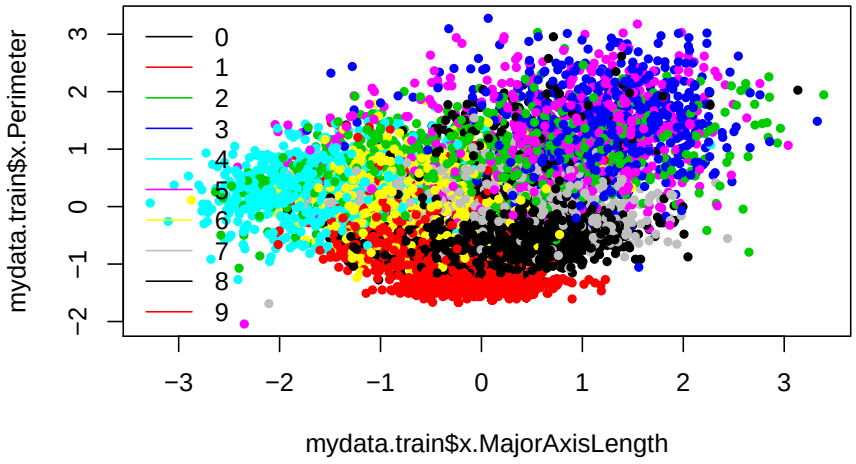


Figura 3.2: Exemple de la distribució per classes amb perimetre i eix més llarg.

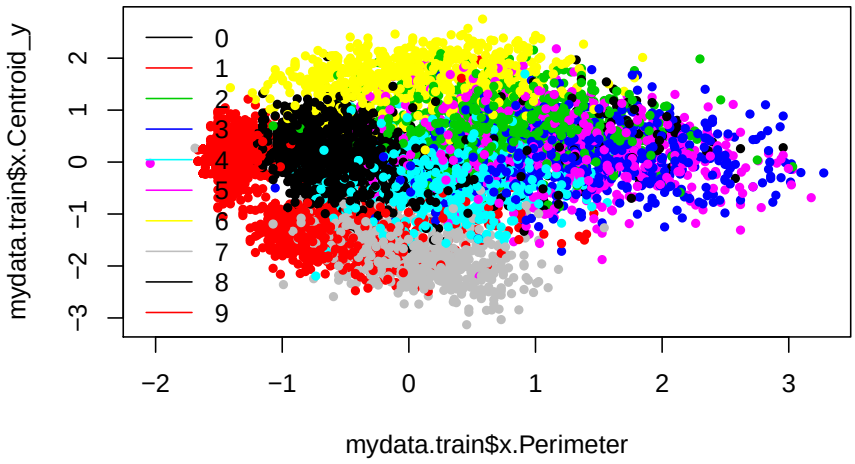


Figura 3.3: Exemple de la distribució per classes amb perimetre i coordenada y del centroide.

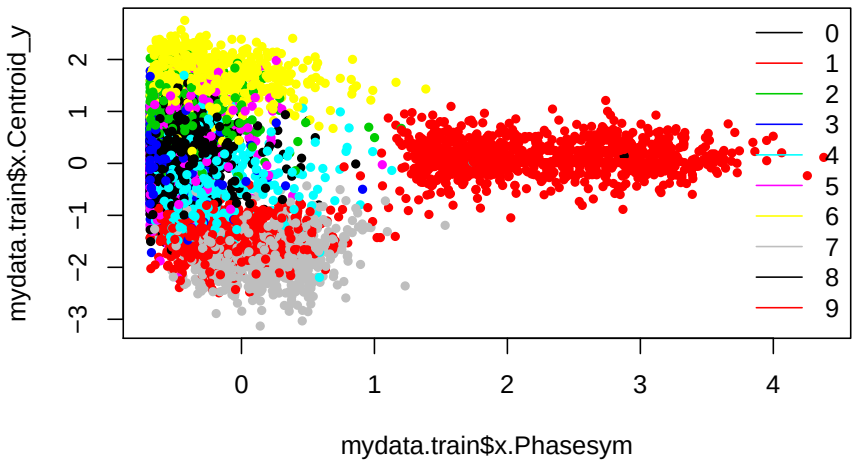


Figura 3.4: Exemple de la distribució per classes amb similitud de fase i coordenada y del centroide.

Capítol 4

Protocol de validació

El protocol de validació que s'ha escollit és ten-fold crossvalidation amb estratificació. Això és degut a que tenim un gran nombre d'observacions d'entrenament (7291) però com s'ha observat al primer apartat, tenim 10 categories i un cop dividit el nombre d'individus en els 10 grups per fer la cros-validació, ens hem d'assegurar que n'hi hagi un nombre proporcional de cada una d'elles.

	0	1	2	3	4	5	6	7	8	9	Total
Ent.	1194	1005	731	658	652	556	664	645	542	644	7291
E.E.	1075	905	658	592	587	500	598	581	488	580	6562
Val.	119	101	73	66	65	56	66	65	54	64	729

Taula 4.1: Nombre d'individus de Entrenament, i el nombre d'individus en les divisions del ten-fold crossvalidation.

La 10 cros-validació implica que del total de la mostra d'entrenament se'n faran 10 subconjunts dels quans 9 serviran d'entrenament i el desè servirà per dur a terme la validació. Això es realitza un total de 10 cops per tal que un d'aquests subconjunts hagi servit de validació dels altres 9 un cop.

Amb l'estratificació volem aconseguir que les mitjanes dels valors de resposta siguin iguals a tots els subconjunts.

En cap cas s'han fet servir les dades de prova per la creació i validació dels models.

Capítol 5

Models de predicció escollits

Els models escollits han estat:

5.1 K veïns més propers

És un mètode de classificació no paramètric que estima el valor de la funció de densitat de probabilitat o directament la probabilitat a posteriori que un elemen pertanyi a la classe a partir de la informació proporcionada pel conjunt de prototipus. En el procés d'aprenentatge no es fa cap suposició sobre la distribució de les variables predictores.

5.2 Naive Bayes

És un classificador probabilístic basat en el teorema de Bayes i algunes hipòtesis simplificadores que es solen resumir en la indeèndència entre les variables predictores.

5.3 Xarxa neuronal

És un model computacional inspirat en la estructura i/o aspectes funcionals de les xarxes neuronals biològiques. Una xarxa neuronal està formada per grups interconnectats de neurones artificials i processa informació utilitzant un enfocament connexionista a la computació. És un sistema adaptatiu que canvia la seva estructura basant-se en informació externa o interna que flueix a través de la xarxa durant la seva fase d'aprenentatge. Són eines per modelar dades estadístiques no lineals i es fan servir per modelar relacions complexes entre entrades i sortides o per trobar patrons a les dades.

5.4 Màquina de vector de suport

És un concepte a estadística i computació per un conjunt de mètodes d'aprenentatge supervisat per tal d'analitzar dades i reconèixer patrons, utilitzat tant en classificació com en regressió. Una SVM agafa les dades d'entrada i prediu, per cada una d'elles, quina de les dues possibles classes forma l'entrada, fent de la SVM un classificador lineal binari no probabilístic.

Capítol 6

Resultats dels models de predicció

A l'hora de crear els models amb les dades s'han utilitzat les funcions tunePareto per KNN i Naive Bayes, tune.nnet per la xarxa neuronal i tune.svm per la màquina de vector de suport. Aquestes funcions ens han permès dur a terme la 10 cros-validació i l'afinament dels paràmetres de cada un dels models. Posteriorment, quan s'hagi triat el millor model depenent dels valors obtinguts de l'error d'entrenament, es passarà a realitzar un afinament d'aquests paràmetres. Es presenten els resultats de la utilització d'aquestes funcions.

6.1 K veïns més propers

Fent la prova amb varis valors del paràmetre k obtenim els següents errors de training:

K	CV.Error
1	0.05481746
2	0.06375240
3	0.05454296
5	0.05326654
7	**0.05292341**
9	0.05498216
11	0.05661543

Veiem que el millor valor pel paràmetre k és 7 i ens dóna un error del 5,3%.

6.2 Naive Bayes

Per Naive Bayes l'única opció disponible per tocar era el kernel així que hem provat tots els que teníem a la nostra disposició obtenint els següents resultats:

En aquest cas el resultat és força dolent, un 15,7% d'error de training indiferentment del kernel utilitzat.

kernel	CV.Error
gaussian	0.1569174
epanechnikov	0.1569174
rectangular	0.1569174
triangular	0.1569174
biweight	0.1569174
cosine	0.1569174
optcosine	0.1569174

6.3 Xarxa neuronal

Com que el nombre de neurones a la capa oculta ha d'anar del nombre d'inputs (25 variables) al d'outputs (10 categories), provarem els valors 15, 20 i 25 amb varis valors de decay. Si al final resulta ser el model escollit intentarem afinar en aquest aspecte.

	size	decay	error	dispersion
1	15	0.0	0.04694348	0.005051069
2	20	0.0	0.04570795	0.006233480
3	25	0.0	0.04762974	0.004761343
4	15	0.1	0.03926429	0.005157083
5	20	0.1	0.03637166	0.002837489
6	25	0.1	0.03498854	0.004911924
7	15	0.5	0.03664961	0.005089838
8	20	0.5	0.03198014	0.004935766
9	25	0.5	0.03019402	0.003706689

6.4 Màquina de vector de suport

Per veure els resultats que ens podia donar el model amb màquina de vector suport primerament hem fet una selecció del millor kernel per aquest problema. Un cop hem trobat quins kernels semblaven ser més adequats hem buscat quins eren els paràmetres més òptims per cada un d'ells. A més també s'han provat els resultats amb totes les variables extretes en els passos anteriors, i unes altres proves amb les seleccionades per els algorismes utilitzats en la fase de selecció de característiques 3.

En tots aquests casos s'ha utilitzat 10-fold cross validation amb les classes estratificades per no provocar mal ajustos per falta de dades d'una classe.

6.4.1 Buscant el kernel òptim

	CV.Error	
Cost	Kernel	CV.Error
0.01	polynomial	0.40
0.01	radial	0.21
0.01	sigmoid	0.23
0.1	polynomial	0.14
0.1	radial	0.09
0.1	sigmoid	0.15
1	polynomial	**0.06**
1	**radial**	**0.05**
1	sigmoid	0.21
10	polynomial	**0.05**
10	**radial**	**0.04**
10	sigmoid	0.24

6.4.2 Paràmetres òptims amb kernel polinòmic

- metode : 10-fold cross validation
- kernel : polinomic
- variables : x.Area , x.EulerNumber , x.Perimeter , x.MajorAxisLength , x.MinorAxisLength , x.Centroid_x , x.Centroid_y , x.Orientation , x.Solidity , x.ConvexArea , x.FilledArea , x.EquivDiameter , x.Phasesym , x.std2 , x.mean2 , x.sim0 , x.sim1 , x.sim2 , x.sim3 , x.sim4 , x.sim5 , x.sim6 , x.sim7 , x.sim8 , x.sim9
- millor resultat : 0.03992109
  - gamma : 0.5
  - cost : 4

	gamma	cost	error	dispersion
1	0.50	4.00	0.04	0.01
2	1.00	4.00	0.04	0.01
3	2.00	4.00	0.04	0.01
4	0.50	8.00	0.04	0.01
5	1.00	8.00	0.04	0.01
6	2.00	8.00	0.04	0.01
7	0.50	16.00	0.04	0.01
8	1.00	16.00	0.04	0.01
9	2.00	16.00	0.04	0.01
10	0.50	32.00	0.04	0.01
11	1.00	32.00	0.04	0.01
12	2.00	32.00	0.04	0.01

- metode : 10-fold cross validation

- kernel : polinomic
- variables : x.Perimeter , x.EulerNumber , x.Phasesym , x.MajorAxisLength , x.Centroid_x , x.Centroid_y , x.FilledArea , x.Orientation , x.Solidity , x.sim0 , x.sim1 , x.sim2 , x.sim3 , x.sim5 , x.sim6 , x.sim7
- millor resultat : 0.04829033
  - gamma : 0.5
  - cost : 4

	gamma	cost	error	dispersion
1	0.50	4.00	0.05	0.01
2	1.00	4.00	0.05	0.01
3	2.00	4.00	0.05	0.01
4	0.50	8.00	0.05	0.01
5	1.00	8.00	0.05	0.01
6	2.00	8.00	0.05	0.01
7	0.50	16.00	0.05	0.01
8	1.00	16.00	0.05	0.01
9	2.00	16.00	0.05	0.01
10	0.50	32.00	0.05	0.01
11	1.00	32.00	0.05	0.01
12	2.00	32.00	0.05	0.01

6.4.3 Paràmetres òptims amb kernel radial

- metode : 10-fold cross validation
- kernel : radial
- variables : x.Area , x.EulerNumber , x.Perimeter , x.MajorAxisLength , x.MinorAxisLength , x.Centroid_x , x.Centroid_y , x.Orientation , x.Solidity , x.ConvexArea , x.FilledArea , x.EquivDiameter , x.Phasesym , x.std2 , x.mean2 , x.sim0 , x.sim1 , x.sim2 , x.sim3 , x.sim4 , x.sim5 , x.sim6 , x.sim7 , x.sim8 , x.sim9
- millor resultat : 0.07488397
  - gamma : 0.5
  - cost : 4

	gamma	cost	error	dispersion
1	0.50	4.00	0.07	0.01
2	1.00	4.00	0.22	0.02
3	2.00	4.00	0.59	0.02
4	0.50	8.00	0.07	0.01
5	1.00	8.00	0.22	0.02
6	2.00	8.00	0.59	0.02
7	0.50	16.00	0.08	0.01
8	1.00	16.00	0.22	0.02
9	2.00	16.00	0.59	0.02
10	0.50	32.00	0.08	0.01
11	1.00	32.00	0.22	0.02
12	2.00	32.00	0.59	0.02

- metode : 10-fold cross validation
- kernel : radial
- variables : x.Perimeter , x.EulerNumber , x.Phasesym , x.MajorAxisLength , x.Centroid_x , x.Centroid_y , x.FilledArea , x.Orientation , x.Solidity , x.sim0 , x.sim1 , x.sim2 , x.sim3 , x.sim5 , x.sim6 , x.sim7
- millor resultat : 0.04897338
  - gamma : 0.5
  - cost : 8

	gamma	cost	error	dispersion
1	0.50	4.00	0.05	0.01
2	1.00	4.00	0.10	0.01
3	2.00	4.00	0.27	0.02
4	0.50	8.00	0.05	0.01
5	1.00	8.00	0.10	0.01
6	2.00	8.00	0.27	0.02
7	0.50	16.00	0.05	0.01
8	1.00	16.00	0.10	0.01
9	2.00	16.00	0.27	0.02
10	0.50	32.00	0.05	0.01
11	1.00	32.00	0.10	0.01
12	2.00	32.00	0.27	0.02

6.4.4 Elecció final dels paràmetres

Aquí es pot veure un resum de les quatre execucions de 10-fold cross validation, i en el qual podem veure que amb el kernel polinòmic, amb gamma 0.5, cost 4 i amb totes les variables el resultat es l'optim.

I en la figura 6.1 es posa un exemple de la classificació del model triat per separar les classes per Centroide.

kernel	gamma	cost	variables	CV.Error
polinomic	**0.5**	**4**	**25**	**0.03992109**
polinomic	0.5	4	16	0.04829033
radial	0.5	8	25	0.07488397
radial	0.5	4	16	0.04897338

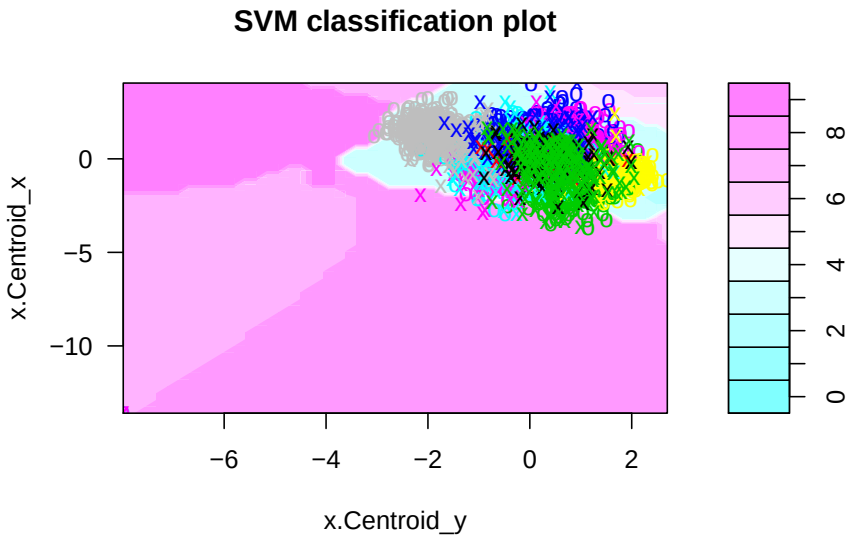


Figura 6.1: Exemple de classificació de totes les dades de training per centroides.

6.5 Comparació dels millors resultats de cada model

A continuació es presenten els millors resultats de cada model:

model	cv.error
KNN	0.05292341
Naive bayes	0.1569174
Xarxa neuronal	0.03019402
SVM	0.03992109

Capítol 7

Resultats model seleccionat

Així doncs, observem que el model amb un menor error de training és la xarxa neuronal. Un cop triat el nostre model final procedim a afinar encara més els seus paràmetres abans d'aplicar-lo a les dades de prova per obtenir el seu error real. Fins ara els resultats òptims s'han trobat amb els valors de size = 25 i decay = 0.5. S'ha repetit el procés de crida a la funció tune.nnet amb valors propers a aquests dos per veure si aconseguíem un error més petit al que tenim actualment però s'ha pogut comprovar com aquests són els millors.

Així doncs, l'error d'entrenament del nostre model ha quedat en un 3,02

Per fer això, realitzem una predicció amb el model i les dades de test i observem el següent:

	0	1	2	3	4	5	6	7	8	9
0	349	0	2	1	1	3	0	0	1	0
1	0	254	0	0	2	0	0	0	1	1
2	1	0	178	3	7	1	3	1	2	0
3	0	2	4	148	0	7	0	1	0	2
4	1	4	1	0	181	0	2	5	0	2
5	1	0	6	10	2	143	1	1	2	0
6	2	3	4	0	1	0	164	0	1	0
7	1	1	1	1	1	2	0	139	0	4
8	4	0	2	2	0	2	0	0	159	0
9	0	0	0	1	5	2	0	0	0	168

Les files ens diuen la classe predita pel nostre model i les columnes la classe real del dígit. Fent els càlculs pertinents veiem que el model ha tingut 1883 encerts i 124 fallades (d'un total de 2007 observacions).

Podem veure com l'error de prova és del 93.82162%, cosa que significa que de cada 100 observacions a les que apliquem el nostre model, encertarem el dígit real 94 cops.

Capítol 8

Conclusions

Durant el treball ens hem vist obligats a investigar diferents mètodes per fer l'extracció de dades ja que el nostre problema presentava un gran nombre de variables però cap d'elles era prou significativa. Un cop feta l'extracció s'ha intentat fer una selecció d'aquestes però no s'han obtingut uns resultats satisfactoris i hem optat per deixar de banda aquest procés. Un cop hem tingut les dades pre-processades ens hem decantat per un protocol de validació i hem provat diferents models per veure quin era el més adient en el nostre cas, duent a terme un seguit de proves i acabant escollint el que ens donava el millor error de training per posteriorment veure quin era el seu error de test.

Aquest procés ens ha dut a investigar sobre el nostre problema concret buscant informació a la xarxa i buscant mètodes per extreure variables vàlides així com a buscar informació sobre diferents models i la seva aplicació amb el software estadístic R. Durant el projecte ens hem vist obligats a prendre decisions que no sabíem si eren les encertades i que hem hagut de verificar mitjançant prova i error.

Finalment, hem obtingut un model que prediu els dígits amb un alt percentatge d'encert tot i les maneres molt diferents d'escriure'ls.

Bibliografia

[1] "The R Project for Statistical Computing", http://www.r-project.org/, Institute for Statistics and Mathematics of the WU Wien

[2] "Mineria de Dades", http://www.lsi.upc.edu/~belanche/Docencia/mineria/mineria.html, Facultat d'Informàtica de Barcelona, UPC

[3] "R for beginners", http://www.lsi.upc.edu/~belanche/Docencia/mineria/Practiques/R/begin_R.pdf, E. Paradis

[4] "The R Guide", http://www.lsi.upc.edu/~belanche/Docencia/mineria/Practiques/R/Owen-TheRGuide.pdf, W. J. Owen