dbstats: una llibreria d'R que implementa els mètodes estadístics basats en distàncies

Adrià Caballé Mestres

Eva Boj del Val, Pedro Delicado Useros i Josep Fortiana Gregori

Màster en Estadística i Investigació Operativa
Universitat Politècnica de Catalunya

13 de juny de 2012

Índex

Agraïments

M'agradaria donar les gràcies a les persones que m'han ajudat en la realització d'aquest projecte. Sobretot agrair als meus codirectors, els professors Pedro Delicado Useros, Josep Fortiana Gregori i Eva Boj del Val. Poder treballar conjuntament amb ells durant més de sis mesos ha estat realment gratificant. He après molt des d'un punt de vista acadèmic i també personal. La confiança dipositada en mi ha estat total, m'he sentit un més del grup, no, únicament, un alumne. M'han ajudat sempre que ho he necessitat, en el desenvolupament del paquet estadístic que presento, en la realització de la memòria escrita, així com per a resoldre qualsevol problema o dubte de caràcter més personal. L'únic que he intentat fer per agrair tots aquests detalls que m'han mostrat ha estat treballar intensa i profundament. He intentat retornar amb fets i responsabilitat tot el suport que m'han proporcionat. Moltes gràcies als tres.

Adrià

Abstract

Resum

El concepte de distància s'ha utilitzat en diferents camps i és imprescindible en molts del mètodes estadístics en la actualitat. Els anys 1989 i 1990, el professor Carles Cuadras, juntament amb Arenas, tots dos professors de la Universitat de Barcelona, realitzen dos publicacions on neix la idea de mètodes estadístics basats en distancies, desenvolupant entre d'altres la regressió basada en distàncies. Al llarg dels últims anys, s'ha treballat en la recerca de noves tècniques estadístiques basades en distàncies, amb múltiples articles publicats, així com comunicacions i cursos. En aquest treball s'intenta estudiar i il·lustrar d'una forma extensa els fonaments d'aquests mètodes, així com la reproducció dels resultats teòrics en el programari lliure R.

S'han treballat les tècniques d'escalatge multidimensional, regressió basada en distàncies, la versió local de la regressió basada en distàncies, el model lineal generalitzat basat en distàncies i la seva versió local. Per últim, també s'ha tractat els mínims quadrats parcials basats en distàncies.

Tots aquests mètodes són útils principalment en la realització de prediccions d'una variable resposta, ja sigui contínua o discreta, a partir d'una matriu de distàncies entre individus. Aquesta pot ser calculada a partir dels valors que prenen les variables explicatives (per exemple si algunes són variables contínues i altres són categòriques), o directament en el cas que les dades ja siguin distàncies.

L'eina de treball per reproduir els resultats ha estat l'R. Es reflecteix tot el procés de creació d'un paquet estadístic en R, així com la implementació de les tècniques estudiades en un paquet anomenat `dbstats`. La llibreria està en el repositori CRAN, accessible per a tots els usuaris. És en la part de la implementació del paquet on s'ha intentat il·lustrar tots els coneixements adquirits, tant pel que fa a la part més metodològica, com és la programa-

ció en R, com en la de recerca i investigació dels mètodes basats en distàncies.

Mitjançant dos exemples de dades reals s'il·lustra com els mètodes basats en distàncies són una via per extreure informació de les dades. En el primer es realitzen prediccions d'una variable resposta numèrica on les variables explicatives són dades funcionals. En el segon es modelitza un cas d'identificació de grups de risc per a companyies asseguradores d'automòbils. En els dos casos s'incideix en el funcionament de `dbstats`, així com en la interpretació dels resultats obtinguts.

Keywords: distància, regressió, regressió no paramètrica, R, paquet estadístic, Multidimensional scaling, DBLM, DBPLS, DBGLM, mètrica.

Abstract

The distance concept has been applied in several fields and it is essential in many existing statistical methods. In 1989 and 1990, Professor Carles Cuadres and Arenas, who are researches at the University of Barcelona, published two papers where the idea of the distance-based method was born, developing, among others, the distance-based regression. In the last few years, other researchers have worked together in search of new statistical techniques based of distance, with many published papers, communications at meetings and courses. This study attempts to illustrate extensively the fundamentals of these methods and it shows the implementation of the developed theoretical results on the free software R.

Multidimensional scaling, distance-based regression, local distance-based regression, distance-based generalized regression, local distance-based generalized regression and partial least squares have been developed.

All these methods are very useful in making predictions of a response variable, either continuous or discrete, from the matrix of distances between individuals. These distances can either be calculated from observed explanatory variables (for instance when some of them are continuous and others are categorical) or directly input as an interdistances matrix.

The working tool to reproduce the theoretical results has been R. This memory reflects the process of statistical package creation in R and the implementation of the studied techniques in a package called `dbstats`. This package is at CRAN repository, accessible to all users. In the implementation of the package is where it is illustrated all the acquired knowledge. Programming in R (the methodological part) and the research of distance-based statistical methods.

Using two real data examples it shows in how distance-based methods could be used as a way to extract information from data. In the first example we make predictions of a numeric response variable when the explanatory variables are functional data. In the second one, it was modeled the identification of risk groups in an assurance automobile company. In both examples the `dbstats` performance is studied, and the results are interpreted.

Capítol 1

Introducció

El concepte de distància entre individus o poblacions ha estat una eina molt utilitzada en diferents camps, no sols en àmbits com les matemàtiques o la física. Per exemple en ciències com l'antropologia, la biologia, la genètica, la psicologia o la lingüística també hi ha tingut un paper important. En el terreny de l'estadística, ha estat utilitzada en moltes de les tècniques que es fan servir en l'actualitat, sobretot se'n destaquen l'anàlisi de components principals o l'escalatge multidimensional.

Cuadras (1989) i Cuadras and Arenas (1990) presenten les bases dels mètodes basats en distàncies, distance-based (DB) methods. Actualment, mètodes com la regressió basada en distàncies, regressió local basada en distàncies (Boj, Delicado, and Fortiana 2010) o mínims quadrats parcials basats en distàncies (Boj, Grané, Fortiana, and Claramunt 2007) han estat desenvolupats, amb articles publicats en revistes de ressò.

Aquests mètodes són vàlids pels casos on les variables explicatives no són numèriques. Sobretot en el camp de la bioestadística (medicina, biometria, psicologia, etc) es troben molts conjunts de dades que contenen variables numèriques i categòriques, i el tractament pels mètodes usuals no és immediat. Els mètodes basats en distàncies són una via de treball molt adequada en el cas que els predictors tinguin valors no reals.

La distància, que pot ser definida com la longitud del camí més curt entre dues entitats, pren una rellevància major a mesura que la dimensió augmenta. És a dir, per mesurar la distància entre dos punts en una recta, la resta entre el valor d'un i altre punt pot ser una expressió adequada (almenys la més rigorosa). Si s'incorpora una altre dimensió les maneres de quantificar la semblança entre els dos punts augmenten. Es pot definir la distància en

valor absolut, com la suma de diferències en valor absolut en cada una de les dimensions. Una altra és la distància euclidiana, que seria el mòdul del vector que uneix els dos punts. Fins i tot es podria calcular la distància de Mahalanobis, que té present la matriu de covariàncies entre dimensions. Els mètodes basats en distàncies, per tant, tindran uns resultats diferents depenen de l'expressió utilitzada per calcular la matriu de distàncies entre individus.

A més dels resultats teòrics, abans d'iniciar aquest treball es disposava d'una sèrie de codi en R i matlab amb l'aplicació de tècniques estadístiques basades en distàncies. La idea principal del treball realitzat ha estat recopilar tots els codis i fer-ne una adaptació orientada a la creació d'un paquet estadístic en R (`dbstats`). D'aquesta manera, tota la informació inherent als diferents articles queda ordenada i habilitada per l'ús de qualsevol usuari, amb la documentació necessària per entendre com usar el paquet i les seves funcions.

No s'intenta contraposar la utilització dels mètodes estadístics basats en distàncies envers els mètodes usuals. Simplement s'obre una altra via d'anàlisi, que pot ser d'utilitat en molts casos. A més, les dues vies poden treballar conjuntament i es poden complementar.

Objectius

A l'elaborar aquest treball, bàsicament es pretenien assolir els següents objectius:

- Donar les bases teòriques dels mètodes estadístics basats en distàncies. Primer de tot, estudiar la tècnica d'escalatge multidimensional (MDS), el punt de partida per entendre i desenvolupar les següents metodologies: model lineal basat en distàncies, model lineal local basat en distàncies, model lineal generalitzat basat en distàncies, model lineal local generalitzat basat en distàncies i mínims quadrats parcials basats en distàncies.

- Plantejar com crear un paquet en el programari R: realitzar un breu manual amb els elements més importants per poder elaborar una llibreria.

- Realitzar una llibreria en R amb els mètodes estadístics basats en distàncies estudiats: elaboració del package `dbstats`. A més, també es volia tenir alguns exemples d'utilització completa del `dbstats`.

Capítol 2

Metodologia

Els mètodes estadístics basats en distàncies són una alternativa o complement a les tècniques clàssiques més utilitzades en l'estadística com el model lineal, el model lineal generalitzat, o les seves versions locals. En aquest capítol s'explica detalladament en què consisteixen tals mètodes, així com el desenvolupament matemàtic que s'ha de realitzar per entendre'ls com un problema basat en distàncies. La notació que s'ha utilitzat és lleugerament diferent de l'emprada habitualment, la matriu de variables explicatives es denota com a Z i la X fa referència a la configuració euclidiana de la matriu de distàncies al quadrat Δ^2.

2.1 Escalatge multidimensional

Història de l'escalatge multidimensional

Donada una matriu $n \times n$ amb les distàncies o *dissimilituds* entre individus d'un conjunt de dades, l'objectiu de l'escalatge multidimensional (multidimensional scaling MDS) és representar aquesta matriu a partir de q variables ortogonals $X_1, ..., X_q$ no observables ($q < n$), tal que les distàncies euclidianes calculades en X siguin iguals, o el màxim de similars possibles, a les distàncies o *dissimilituds* inicials.

La primera proposta d'un algorisme de multidimensional scaling va ser presentada l'any 1958 per l'investigador J.W Torgerson. Torgerson (1958) proposa un mètode per construir una matriu X amb les coordenades cartesianes dels punts en un espai euclidià, únicament coneixent les distàncies δ_{ij} entre cada parella de punts o observacions.

Gower and Krzanowski (1966) estenen el mètode de Torgerson, donant les bases del mètode mètric d'anàlisi de coordenades principals. Shepard (1977) desenvolupa el nonmetric MDS (MDS ordinal) i Kruskal (1964) dóna un criteri o mesura de l'adequació de la representació en les q noves dimensions respecte la matriu de distàncies en N dimensions inicial. Carroll and Chang (1970) introdueixen el programa INDSCAL (Individual Scaling). Per últim, destaca l'aportació de Ramsay (1977) que introdueix proves de significació basades en mètodes de màxima versemblança del MDS mètric.

El MDS és una tècnica que té un paper important en l'actualitat i un camp on encara hi ha molta investigació. Per exemple, es pot destacar el llibre *Modern Multidimensional Scaling* (Borg and Groenen 2005).

Concepte de *similitud* i *dissimilitud*

Es defineix la proximitat entre dos objectes com la quantificació de la semblança que hi ha entre ells. Si es pren p_{ij} com una mesura de proximitat entre dos objectes (i, j) qualssevol, si es compleix que la magnitud dels valors de p_{ij} està ordenada de tal manera que les parelles d'objectes més semblants siguin les que tenen un valor més alt i les parelles mes diferents un valor més baix, llavors es considera que p_{ij} és una mesura de *similitud* i es denota per s_{ij}. Alternativament, si l'ordenació està girada, és a dir els p_{ij} més baixos són les parelles d'objectes que més s'assemblen, es considera que és una mesura de *dissimilitud* i es denota per δ_{ij}.

Coombs (1964) defineix el concepte *similitud* com la forma que les persones perceben la semblança o la no semblança entre objectes. L'escalatge multidimensional és una família de mètodes que analitza dades de proximitat: *similituds* i *dissimilituds*.

Mètriques euclidianes i no euclidianes

Una mètrica en un conjunt Z és una funció de distàncies que per a qualssevol punts x, y i $z \in Z$ compleix:

1. **Diagonal nul·la**: la distància entre un punt i ell mateix és 0

$$d(x, x) = 0.$$

2. **Simetria**: La distància de x a y és la mateixa que de y a x

$$d(x, y) = d(y, x).$$

3. **No degenerada**: si la distància entre un punt x i un punt y és zero, llavors x i y fan referència al mateix punt

$$d(x, y) = 0, \quad \text{llavors } x = y \ \forall x, y \in Z.$$

4. **Desigualtat triangular**: la distància entre un punt x i un punt y ha de ser menor o igual a la distància entre x i y passant entremig per z

$$d(x, y) \leq d(x, z) + d(z, y).$$

Una mètrica δ en un conjunt de dades Z és euclidiana si existeix un espai euclidià $\mathbb{E}$ (espai vectorial amb $<,>$ i $||x|| =< x, x >$) i per cada $z \in Z$ un vector $X_z \in \mathbb{E}$, amb la propietat de isometria: per dos punts z_i i $z_j \in Z$ qualssevol, X_{z_i} i $X_{z_j} \in \mathbb{E}$ compleixen que

$$||X_{z_i} - X_{z_j}|| = \delta_{z_i, z_j}.$$

El conjunt de vectors

$$\mathbb{E}(Z) = \{X_z \in \mathbb{E} : z \in Z\} \subset \mathbb{E}$$

s'anomena configuració euclidiana de Z donada la mètrica δ i, particularment, X és la matriu de coordenades euclidianes de (Z, δ).

Per exemple, en un conjunt de dades amb tres observacions, (a, b, c), donada la matriu de dissimilituds entre els tres punts:

$$\Delta = \begin{pmatrix} 0 & & \\ 1 & 0 & \\ \lambda & \lambda & 0 \end{pmatrix}$$

per $\lambda < 1/2$ la mètrica no és euclidiana. De fet no és una mètrica atès que la desigualtat triangular falla (es pot observar en el primer gràfic de la Figura 2.1):

$$d(a, b) = 1 > d(a, c) + d(c, b) = \lambda + \lambda.$$

Altrament, si $\lambda > 1/2$ (per exemple $\lambda = \sqrt{2}$) la mètrica sí que compleix la desigualtat triangular:

$$d(a, b) = 1 < \lambda + \lambda.$$

i és euclidiana (s'observa en el segon gràfic de la Figura 2.1).

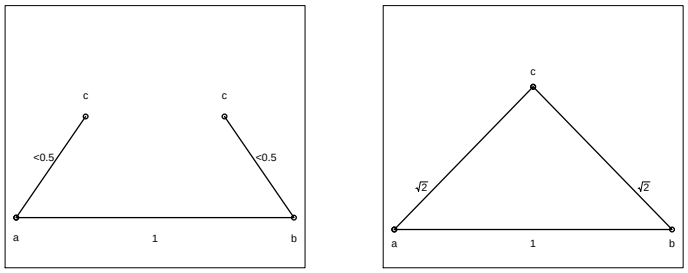

Figura 2.1: La dissimilitud de l'esquerra no compleix la desigualtat triangular i per tant no és una mètrica ben definida. En canvi, la de la dreta és una mètrica euclidiana.

Fonaments del MDS

Les tècniques de MDS estudien les relacions entre elements o parelles d'elements i resulten útils com una eina de visualització. Per exemple, reduint la dimensió inicial N a un pla ($q = 2$), es pot representar gràficament, i per tant d'una manera molt més intel·ligible, de quina manera s'assemblen els individus o poblacions.

Donat un conjunt de dades amb N individus, existeixen $S = \frac{N(N-1)}{2}$ dissimilituds entre parells d'individus δ_{ij} com s'observa en la següent matriu:

$$\begin{pmatrix} \delta_{1,1} & \delta_{1,2} & ... & \delta_{1,N} \\ \delta_{2,1} & \delta_{2,2} & ... & \delta_{2,N} \\ \vdots & \vdots & & \vdots \\ \delta_{N,1} & \delta_{N,2} & ... & \delta_{N,N} \end{pmatrix}$$

L'objectiu del MDS és trobar els q vectors $X^{(1)}$, $X^{(2)}$, $\cdots$, $X^{(q)} \in R^N$ tal que la distància euclidiana per a cada parella de punts d_{ij} en el nou espai s'aproximi a la dissimilitud inicial corresponent δ_{ij}:

$$d_{ij} = ||X_i - X_j|| \approx \delta_{ij}.$$

Quan l'aproximació es compleix com igualtat es tracta del MDS mètric i la solució obtinguda és exacta. En aquest apartat s'estudia el primer algorisme de MDS mètric (Torgerson 1958). Cal indicar que el mètode de MDS mètric més utilitzat és el càlcul d'Anàlisi de coordenades principals (Gower and Krzanowski 1966). Aquest és menys restrictiu en el conjunt de dades que es vol tractar ja que permet qualsevol mesura de comparació entre els objectes. En el mètode inicial de Torgerson els elements de la matriu han de ser estrictament distàncies euclidianes.

Quan hi ha diferencies entre d_{ij} i δ_{ij}, l'objectiu és trobar les coordenades que minimitzin aquestes discrepàncies (MDS no mètric). Hi ha diversos mètodes proposats segons quina mesura d'aproximació s'optimitza. Es poden diferenciar dues metodologies. La primera, Sammon (1969), és molt propera al MDS mètric. Intenta minimitzar una funció de cost que descriu com de bé es conserven les dissimilituds originals un cop calculades les distàncies $d(i, j)$. La funció de cost de Sammon és

$$STRESS = \sum_{k \neq l} \frac{[d(k,l) - \delta(k,l)]^2}{\delta(k,l)}.$$

La diferència respecte el mètode mètric és que els errors són normalitzats. D'aquesta manera, errors on les dissimilituds inicials són petites són emfatitzats.

La segona metodologia, Shepard (1977) i Kruskal (1964), presenta una restricció addicional a d. Considera la matriu de dissimilituds com un rang ordenat entre parelles d'observacions. Es a dir, la parella de punts més semblant té valor 0 i la menys semblant valor 1. Es pretén trobar una configuració de les dades on aquest ordre es mantingui. El millor rang ordenat possible en la nova configuració pot ser garantit introduint una funció monòtona decreixent f, que actua en δ i aconsegueix trobar els valors en d que millor

conserven l'ordenació inicial.

MDS mètric segons Torgerson

Torgerson defineix la possibilitat de determinar les coordenades en l'espai euclidià X a partir d'una configuració euclidiana de les dades. De fet, demostra que existeix una matriu de productes escalars $G = XX'$, que per cada parella (i, j):

$$||X_i - X_j|| = \delta_{ij}.$$

Per realitzar aquesta configuració únicament es necessiten dos passos:

1. Usant la *regla del cosinus* es converteix la matriu de distàncies Δ a una matriu G de productes escalars.

2. Aplicant SVD (singular value descomposition) de G s'obté X.

Regla del cosinus
Suposant tres individus (i,j,k) en l'espai euclidià amb distàncies entre ells $\delta_{i,j}, \delta_{i,k}, \delta_{j,k}$ i angle θ_{jik} entre els vectors de distància $\delta_{i,j}$ i $\delta_{i,k}$:

$$cos\theta_{jik} = \frac{1}{2\delta_{i,j}\delta_{i,k}}(\delta_{i,j}^2 + \delta_{i,k}^2 - \delta_{j,k}^2)$$

Prenent $g_{jk} = \frac{1}{2}(\delta_{i,j}^2 + \delta_{i,k}^2 - \delta_{j,k}^2)$ és dedueix que $g_{jk} = \delta_{i,j}\delta_{i,k}cos\theta_{jik}$.

Donat $c = (\mathbb{1}'X)/n$ on $\mathbb{1}$ és un vector columna amb n uns. Per tant, c és un vector fila amb les mitjanes de les columnes de X. Es defineix X^* com el centrament de X a partir del vector de mitjanes c:

$$X^* = X - \mathbb{1}c = X - (\mathbb{1}\mathbb{1}'X)/n.$$

$G^* = X^*X^{*'}$ s'obté mitjançant el doble centrat de la matriu de Δ^2 que conté les distàncies euclidianes al quadrat. G es pot definir com els productes interiors centrats de la matriu de distàncies Δ^2. Desenvolupant la formula s'obté:

$$\begin{aligned}
G^* &= X^*X^{*'} \\
&= (X - \mathbb{1}c)(X - \mathbb{1}c)' \\
&= (X - (\mathbb{1}\mathbb{1}'X)/n)(X - (\mathbb{1}\mathbb{1}'X)/n)' \\
&= (X - \mathbb{1}\mathbb{1}'X/n)(X - X'\mathbb{1}\mathbb{1}'/n)' \\
&= (XX' - XX'\mathbb{1}\mathbb{1}'/n - \mathbb{1}\mathbb{1}'XX'/n + \mathbb{1}\mathbb{1}'XX'\mathbb{1}\mathbb{1}'/n^2) \\
&= (G - G_{\cdot} - G_{\cdot}' + G_{\cdot\cdot})
\end{aligned}$$

En el cas particular del producte escalar entre dos individus (i, j) es pot formular de la següent manera:

$$g^*_{ij} = g_{ij} - \frac{1}{n}\sum_{k}^{n} g_{ik} - \frac{1}{n}\sum_{k}^{n} g_{kj} + \frac{1}{n^2}\sum_{r}^{n}\sum_{h}^{n} g_{rh}.$$

Tenint present que $g_{jk} = \frac{1}{2}\left(\delta^2_{ij} + \delta^2_{ik} - \delta^2_{jk}\right)$ l'expressió de g^*_{ij} es simplifica:

$$g^*_{ij} = -\frac{1}{2}\left[\delta^2_{ij} - \frac{1}{n}\sum_{k}^{n} \delta^2_{ik} - \frac{1}{n}\sum_{k}^{n} \delta^2_{kj} + \frac{1}{n^2}\sum_{r}^{n}\sum_{h}^{n} \delta^2_{rh}\right].$$

Es tracta de $-1/2$ vegades la distància al quadrat entre les parelles de punts (i, j) traient la mitjana de les distàncies al quadrat de la fila i-èssima i la mitjana de les distàncies al quadrat de la columna j-èssima, i sumant, finalment, la mitjana de distàncies al quadrat global.

Aplicant una descomposició en valors singulars (SVD) de la matriu $G^* = U^*\Lambda^* U^{*'}$, permet definir $X^* = U^*\Lambda^{*1/2}$. Aquí Λ^* és diagonal amb els autovalors de G^* ordenats de forma decreixent, i les columnes de U^* són els autovectors de G^*. Es pot resumir l'algorisme de Torgerson en tres passos:

1. Realitzar el doble centrat de la matriu de distàncies al quadrat per tal d'obtenir G.

2. Resoldre el problema de descomposició en valors singulars de tal forma que $G = U\Lambda U'$

3. Escollir la mida efectiva q corresponent als q primer valors singulars.

MDS no mètric de Kruskal

Kruskal (1964) dóna un criteri per saber quant s'aproxima la representació en les q noves dimensions a la matriu de dissimilituds inicial $(n \times n)$. La mesura s'anomena *STRESS*:

$$STRESS = \frac{\sum_{k\neq l}[d(k,l) - f(\delta(k,l))]^2}{\sum_{k\neq l}[f(\delta(k,l))]^2}.$$

L'objectiu és buscar la configuració òptima, es a dir la representació en l'espai de dimensió q que faci mínim el valor de la funció *stress*. Per tant, es tracta d'un problema d'optimització amb el següent algorisme simplificat:

1. Assignar punts a coordenades de l'espai de dimensió q de manera arbitrària.

2. Calcular les distàncies euclidianes $d(k, l)$ en totes les parelles de punts, i comparar-la amb les $\delta(k, l)$ inicials per tal d'avaluar la funció *stress*.

3. Ajustar les coordenades per cada punt que minimitzin el *STRESS*.

Repetir el pas 2 i 3 fins que el *STRESS* ja no canviï significativament.

Comparació entre l'anàlisi de components principals i l'escalatge multidimensional

L'anàlisi de components principals (ACP) i l'escalatge multidimensional (MDS) són tècniques que segueixen un camí paral·lel en molts aspectes, però amb marcades diferències, tant des del punt de vista metodològic, com en la seva intenció.

Els dos mètodes no requereixen cap suposició de distribucions de probabilitat en les dades, per tant es poden aplicar a conjunts de dades amb distribució desconeguda. A més, comparteixen, en part, alguna de les fases en la computació, atès que en les dues es busca la diagonalització d'una matriu de productes escalars, trobant els seus autovalors i autovectors. Per últim, comparteixen una de les seves principals finalitats, reduir la dimensió inicial del conjunt de dades a una dimensió mínima, amb els mateixos problemes a l'hora de decidir amb quina dimensió quedar-se.

Difereixen en la configuració en productes escalars. En MDS, les noves coordenades es calculen a partir de la matriu de distàncies entre individus i en ACP a partir de la matriu de covariàncies entre variables. A més, en ACP es disposa d'un conjunt de puntuacions dels n individus a les p variables on les dimensions representen característiques observades. En MDS el resultat dels càlculs és un conjunt de puntuacions dels individus en unes variables latents (no observades): les coordenades de la configuració euclidiana.

2.2 Regressió basada en distàncies

Història de la regressió basada en distàncies

La regressió basada en distàncies és una metodologia relativament recent desenvolupada principalment per professors/investigadors de la Universitat

de Barcelona. Es considera el problema de predicció d'una variable contínua a partir de variables explicatives qualitatives. Són conegudes les dificultats d'aplicar la regressió lineal clàssica en aquests casos. Normalment, la solució utilitzada és escalar les variables qualitatives de manera adequada, considerant les variables resultants com a quantitatives. Cuadras (1989) resol el problema basant-se en les tècniques de multidimensional scaling vistes en la secció anterior. Especialment, aplica el mètode proposat per Gower and Krzanowski (1966) d'anàlisi de coordenades principals. La idea és construir una matriu de similituds o dissimilituds a partir de les variables explicatives inicials, aplicar MDS i considerar les coordenades principals resultants X com els predictors amb els quals es calcula el model.

Cuadras and Arenas (1990) estenen el cas plantejat inicialment pel problema de predicció d'una variable contínua a partir d'una sèrie de variables explicatives (algunes contínues i d'altres categòriques). Amb aquest article ja es tenen totes les bases per poder aplicar regressió basada en distàncies en qualsevol problema clàssic de regressió.

Model de regressió clàssic

El model de regressió clàssic és un mètode que modelitza la relació entre una variable dependent (resposta) contínua i una sèrie de variables independents (explicatives) sumat a un terme d'error aleatori amb distribució $N(0, \sigma)$.

$$Y = Z\gamma + \varepsilon. \tag{2.1}$$

On Z és la matriu coneguda de variables explicatives (o predictors) $n \times p$ i de rang p, i γ és un vector columna de $p + 1$ escalars desconeguts. A més, es suposa que $E[\varepsilon] = 0$ i la variança $E[\varepsilon\varepsilon'] = \sigma^2 I_n$ desconeguda. Per mínims quadrats (OLS) s'estima γ:

$$\hat{\gamma} = (Z'Z)^{-1}(Z'Y).$$

Model en forma centrada i ortogonal

Manipulant el model (2.1) obtingut anteriorment es poden fer deduccions que ajudaran en apartats posterior a entendre la regressió basada en distàncies.

Es suposa que la variable resposta està centrada. És a dir, la mitjana de Y és igual a zero. Es considera la descomposició espectral de $Z'Z$, matriu de dimensió $p \times p$, tal que $Z'Z = U\Lambda U'$. Aquí Λ és una matriu diagonal que

conté els valors propis $(\lambda_1, \lambda_2, ..., \lambda_p)$ i U és una matriu $p \times p$ amb els vectors propis de $Z'Z$, el model (2.1) es pot reescriure com:

$$Y = X\beta + \varepsilon, \tag{2.2}$$

on $X = ZU$ és la matriu de predictors ortogonalitzada $(n \times p)$, i $\beta = U^{-1}\gamma$ és el vector de coeficients. És pot veure que el model continua sent el mateix:

$$X\beta = ZUU^{-1}\gamma = Z\gamma.$$

Al model (2.2) està centrat i ortogonalitzat respecte a (2.1). El vector de coeficients β, es pot estimar com: $\hat{\beta} = \Lambda^{-1}(X'Y)$. Es dedueix substituint adequadament:

$$\begin{aligned}\hat{\beta} &= (X'X)^{-1}(X'Y) \\ &= (U'Z'ZU)^{-1}(X'Y) \\ &= (U'U\Lambda U'U)^{-1}(X'Y) \\ &= \Lambda^{-1}(X'Y).\end{aligned}$$

S'ha de tenir present que s'ha ortogonalitzat $X'X$ de tal forma que el producte escalar entre $U'U$ és la matriu identitat de dimensió $p \times p$. El model (2.2) satisfà:

$$X_i'X_i = \lambda_i; \quad X_i'X_j = 0, i \neq j; \quad X_i'\mathbb{1} = 0.$$

Pel que fa a la predicció d'un nou individu, suposant que pren valors en les variables explicatives $\tilde{z} = (\tilde{z}_1, \tilde{z}_2, ..., \tilde{z}_p)$, s'ha de fer la transformació del vector $\tilde{z}$ al vector ortogonalitzat respecte el model (2.2) $\tilde{x}$:

$$\tilde{x}' = \tilde{z}U.$$

La predicció $\tilde{y}(\tilde{z})$ s'obté substituint:

$$\tilde{y}(\tilde{z}) = \tilde{x}\Lambda^{-1}(X'y) = \tilde{x}\hat{\beta}$$

L'estimació $\hat{\beta}$ verifica les següents propietats:

$$\hat{\beta}_i = X_i'Y/\lambda_i; \qquad E(\hat{\beta}_i) = \beta_i; \qquad var(\hat{\beta}_i) = \sigma^2/\lambda_i$$

Model de regressió basat en distàncies

Es pretén fer prediccions d'una variable resposta contínua (Y) a partir d'una matriu de p predictors (Z).

Sigui Δ^2 la matriu de distàncies al quadrat entre individus, calculada a partir dels predictors Z amb una certa funció de distàncies. Es suposa que Δ^2 té la propietat de mètrica euclidiana (definida a la pàgina 5). Es defineix la matriu A, tal que els elements $a_{ij} = -(\delta_{ij}^2)/2$ i $G = JAJ$ (J matriu de centratge). G és una matriu semidefinida positiva (Mardia, Kent, and Bibby 1979), i assumint que té rang p, s'ha vist que

$$G = XX'.$$

X és una matriu $(n \times p)$ i de rang p que conté les coordenades principals en l'espai euclidià de Z. A més $X'X = \Lambda$, matriu diagonal amb els valors propis positius de G $(\lambda_1, ..., \lambda_p)$. Aleshores es verifica que

$$\delta_{ij} = (x_i - x_j)'(x_i - x_j).$$

Es pot fixar $X = (X_{(k)}, W)$, on $X_{(k)}$ conté les k primeres coordenades principals. Considerant $X_{(k)}$ com la matriu de predictors lineals se'n deriva:

$$Y = \sum_{i=1}^{k} \beta_i X_i + \epsilon_k. \tag{2.3}$$

En el cas que $k = p$, és a dir el número de coordenades principals escollit és igual al nombre total de variables i la funció de distàncies és euclidiana, aleshores es tracta del model complet.

El model (2.3) és el model centrat i ortogonalitzat vist anteriorment, i s'estima β de la següent manera:

$$\hat{\beta} = \Lambda^{-1} X'Y.$$

Predicció d'un nou individu (I)
Per tal de donar la predicció $\hat{y}_{n+1}$ d'una nova observació, sigui $\delta^2_{[n+1]}$ el vector de distàncies euclidianes al quadrat entre l'individu $(n+1)$ i els individus utilitzats per definir el model. Per la fórmula d'interpolació de Gower (1968) es defineix el punt x_{n+1} amb dimensió $(p \times 1)$:

$$x_{n+1} = \frac{1}{2}\Lambda^{-1} X' \left(g - \delta^2_{[n+1]}\right),$$

on g és la diagonal de la matriu de productes interiors G. Substituint, la predicció pel nou individu s'obté:

$$\begin{aligned}\hat{y}_{n+1} &= x_{n+1}\beta \\ &= \tfrac{1}{2}\Lambda^{-1}X'\left(g-\delta^2_{[n+1]}\right)\Lambda^{-1}X'Y \\ &= \tfrac{1}{2}\left(g-\delta^2_{[n+1]}\right)'X\Lambda^{-2}X'Y\end{aligned}$$

Relació amb el model lineal ordinari

Quan els predictors són continus i s'utilitza la distància euclidiana, el model de regressió lineal basat en distàncies i el clàssic són equivalents.

Donada la matriu de predictors lineals Z, que conté les p variables explicatives, es defineix la distància euclidiana entre cada parella d'individus (i,j) com

$$\delta_{ij} = (z_i - z_j)'(z_i - z_j) = z_i z_i + z_j z_j - 2z_i z_j,$$

A més ja s'ha comprovat que

$$G = JAJ = JZZ'J = XX'.$$

Si arriba un nou individu amb valors en els predictors z_{n+1}, la distància al quadrat d'aquest respecte l'individu i-èssim és:

$$\delta^2_{i,n+1} = (z_{n+1} - z_i)'(z_{n+1} - z_i) = (x_{n+1} - x_i)'(x_{n+1} - x_i).$$

Com que $g_{ii} = x_i'x_i$, aleshores

$$\delta^2_{i,n+1} = x'_{n+1}x_{n+1} + g_{ii} - 2x'_{n+1}x_i.$$

Generalitzant per tots el individus del model

$$\left(g-\delta^2_{[n+1]}\right)' = 2x'_{n+1}X' - x'_{n+1}x'_{n+1}\mathbb{1}'.$$

Tenint present que $\beta = \Lambda^{-1}(X'y)$ i substituint a la predicció de $\hat{y}_{n+1}$ basada en distàncies:

$$\begin{aligned}\hat{y}_{n+1} &= 1/2\left(g-\delta^2_{[n+1]}\right)'X\Lambda^{-2}X'Y \\ &= 1/2\left(2x'_{n+1}X' - x'_{n+1}x'_{n+1}\mathbb{1}'\right)X\Lambda^{-2}X'Y \\ &= x_{n+1}X'X\Lambda^{-2}X'Y - 1/2x'_{n+1}x'_{n+1}\mathbb{1}'X\Lambda^{-2}X'Y\end{aligned}$$

Per definició, $\mathbb{1}'X = 0$ i per tant l'expressió es simplifica:

$$\hat{y}_{n+1} = x_{n+1}X'X\Lambda^{-2}X'Y,$$

Com que $X'X = \Lambda$,

$$\hat{y}_{n+1} = x_{n+1}\Lambda^{-1}X'Y = x_{n+1}\beta.$$

En definitiva, s'ha demostrat que l'estimació de y per un nou individu, utilitzant una funció de distàncies euclidiana per construir la matriu G, és equivalent a la predicció del model lineal clàssic centrat i en forma ortogonal (COF) vist a (2.2).

No obstant, cal assenyalar que fins i tot si la mètrica l^2 és aplicable, en dades de tipus numèric, altres funcions disponibles poden ser molt vàlides i millorar el model lineal obtingut per mínims quadrats.

Hat matrix i model lineal ponderat basat en distàncies

Es defineix la hat matrix (matriu barret) com la matriu d'influència dels valors observats per obtenir els valors previstos. També coneguda com "*puts a hat on y*"

$$\hat{Y} = Hy,$$

i queda definida com:

$$H = X(X'X)^{-1}X'.$$

Algebraicament, la hat matrix és la projecció ortogonal en l'espai columna de X. En el cas que no tots els individus tinguin el mateix pes en el model, es defineix D_ω com la matriu diagonal $(n \times n)$ amb els pesos de cada observació. Llavors la hat matrix esdevé:

$$H_\omega = X_\omega \left(X'_\omega D_\omega X_\omega\right)^{-1} X'_\omega D_\omega$$

Com s'ha vist en la secció anterior, G és la matriu de productes interiors de X tal que $G = XX'$. Es defineix F_ω com l'estandardització de la matriu $G_\omega = -\frac{1}{2}J_\omega\Delta^2 J_\omega$:

$$F_\omega = D_\omega^{1/2} G_\omega D_\omega^{1/2}$$

En conseqüència es pot reescriure la hat matrix com:

$$H_\omega = G_\omega \left(D_\omega^{1/2} F_\omega^{+} D_\omega^{1/2}\right)$$

On F_ω^+ és la Pseudo-inversa de Moore-Penrose (es defineix a continuació) dels productes interiors estandarditzats F_ω. La H_ω no depèn dels valors de les variables explicatives, únicament és funció dels pesos i la G_ω calculada a partir de la matriu de distàncies al quadrat.

Moore-Penrose Pseudoinverse
Definit per Penrose (1955), la matriu pseudo-inversa troba la solució d'un sistema d'equacions lineals amb múltiples solucions amb norma euclidiana mínima. Una matriu N és la Pseudo-inversa de Moore-Penrose de M si i només si compleix:

- MNM = M.
- NMN = N.
- MN i NM són simètriques.

El càlcul de la pseudo-inversa, en aquest cas, es realitza per descomposició en valors singulars (SVD).

Predicció d'un nou individu (II)
Es pot calcular la predicció del nou individu $[n+1]$ de la següent manera:

$$\tilde{y}_{n+1} = \frac{1}{2}\left(g_\omega - \delta_{[n+1]}\right)\left(D_\omega^{1/2} F_\omega^{+} D_\omega^{1/2}\right) Y.$$

L'equació és l'anàloga vista a la pàgina 13:

$$\tilde{y}_{n+1} = \frac{1}{2}\left(g - \delta^2_{[n+1]}\right)' X\Lambda^{-2}X'Y,$$

on el model està ponderat segons el pes de cada observació i la pseudo-inversa de Moore-Penrose de F_ω és obtinguda aplicant descomposició en valors singulars tal que $F_\omega^{+} = U\Lambda^{-1}U'$, elegint les k primeres components (dimensió efectiva)

$$F_\omega^{(k)+} = U^{(k)}(\Lambda^{(k)})^{-1}U'^{(k)}.$$

2.3 Regressió local basada en distàncies

Regressió lineal local clàssica

El model de regressió paramètric és un mètode que modelitza la relació entre una variable resposta (Y) i unes variables explicatives (Z) suposant que les dades proven d'una variable aleatòria amb distribució coneguda exceptuant un número finit de paràmetres a estimar:

$$Y = \beta Z + \varepsilon, \qquad \varepsilon \sim N(0, \sigma I_n)$$

on β es pot indexar per un conjunt de paràmetres desconeguts a $\mathbb{R}^p$. Per tant es tracta d'un model estadístic amb $p+1$ paràmetres desconeguts.

El model de regressió no paramètric modelitza la relació entre Z i Y com

$$Y = m(z) + \varepsilon, \tag{2.4}$$

on $m(z) = E(Y|Z = z)$ és una funció suau (normalment segona derivada contínua) de z que no es pot indexar per un nombre finit de paràmetres. A més, ε és una variable aleatòria amb $E[\varepsilon] = 0$ i $V[\varepsilon] = \sigma^2$.

Per estimar la funció de regressió $m(z)$ no paramètricament es poden aplicar diferents tècniques. Aquí s'estudia el cas d'ajust local amb models paramètrics.

L'ajust local lineal es basa en la idea que, en un entorn petit de z, la relació que hi ha entre la variable resposta i les variables explicatives és aproximadament lineal. En el gràfic de l'esquerra de la Figura 2.2 s'hi defineix un model on només hi ha una variable explicativa que té relació cúbica amb la resposta. En l'entorn d'un individu t, que pren valor $z_t = 1$ en la variable explicativa, s'hi pot intuir un comportament lineal. Per tant ajustant un model lineal usual centrat en t i que involucra només les observacions (z_i, y_i) amb $z_i \in (t-\varepsilon, t+\varepsilon)$, es podria estimar y_t.

Estimadors nucli

En el cas anterior, tots els individus que entren en el model lineal per aproximar l'estimació de la resposta en l'individu t hi intervenen amb el mateix pes. Sembla raonable pensar que les observacions més properes tindran un comportament més similar a t que no les més llunyanes. La solució és assignar un pes diferent a cada observació de manera que les més properes tinguin un pes en l'estimació de y_t més elevat. El pes de (z_i, y_i) en un entorn de t és usualment assignat mitjançant una funció coneguda com a nucli o Kernel K (tal que $K(u)$ decreix amb $|u|$):

$$\omega_i = \omega(t, z_i) = \frac{K(\frac{z_i - t}{h})}{\sum_{j=1}^{n} K(\frac{z_i - t}{h})},$$

on la h (paràmetre de suavitzat o ample de banda) regula la concentració de pes en l'entorn de t. Una h gran suposa donar pes a observacions més allunyades de t i tendeix al model de regressió lineal paramètric usual. Una h petita concentra el pes en les observacions més properes a t i tendeix a la interpolació. Trobar la h òptima en cada cas és la part més important de la

modelització local ja que caracteritza totalment les estimacions resultants. Hi ha diferents tècniques implementades que estudien aquest problema, l'elecció de la h per validació creuada o per substitució (plug-in) són algunes vies d'elecció automàtica.

De funcions nuclis se'n han definit varies. Algunes de les més importants són: Epanèchnikov, Biweight, Triweight, Gaussià, Triangular o Uniforme (presentades en la Figura 2.3). En el gràfic de la dreta de la Figura 2.2 s'observa un cas on el nucli utilitzat és Gaussià.

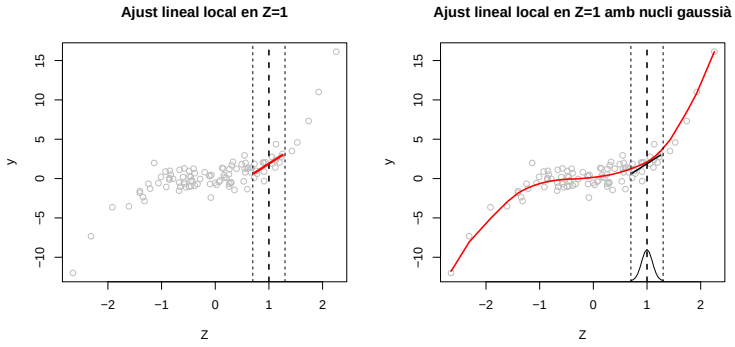

Figura 2.2: Ajust lineal local en z=1. En el primer gràfic tots els individus en el interval marcat en línees discontinues entren en el model amb el mateix pes, en el segon la ponderació es realitza mitjançant una funció nucli Gaussiana centrada en t (z=1).

Es pot ampliar el cas inicial on només hi havia una variable explicativa pel cas multivariant on la dimensió de la z és p. La forma de treballar amb tals models és pràcticament igual, tenint present que el paràmetre de suavitzat h esdevé una matriu H. Per p gran els models no paramètrics perden la fiabilitat, donant-se el que es coneix com maledicció de la dimensionalitat. És a dir, en un entorn de t en les variables explicatives és poc probable trobar-hi altres observacions. L'única manera de trobar dades als entorns de cada punt és operar amb conjunts de dades enormes, o donar pes a observacions llunyanes que poden no ser explicatives del que succeeix en l'entorn de l'individu en qüestió.

Model lineal local basat en distàncies

El model lineal local pretén modelitzar la resposta en funció dels regressos per mitjà d'ajustar models lineals (LM) locals, tants com observacions es

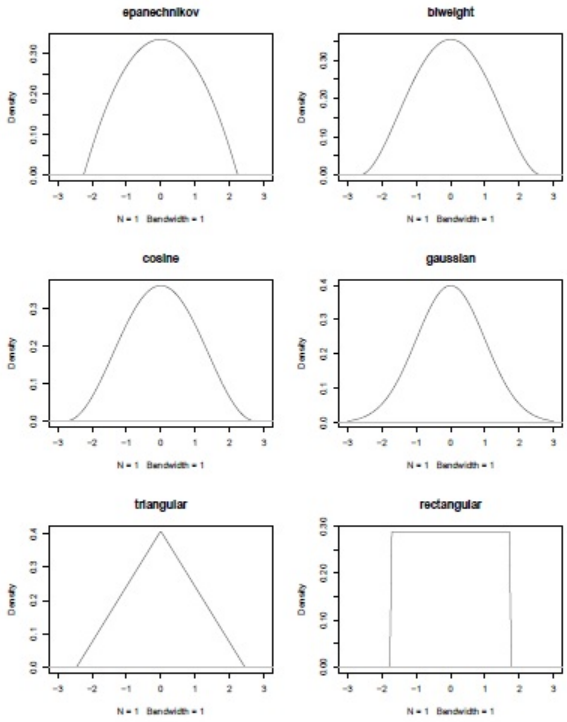

Figura 2.3: Funcions nucli possibles: Epanèchnikov, Biweight, Triweight, Gaussià, Triangular o Uniforme.

tingui (n). Per cada observació t s'ajusta un model lineal ponderat, centrat en t, de tal forma que els pesos són donats per una funció Kernel, estimant per cada cas el valor de la resposta en t.

El model lineal local basat en les distàncies (Boj, Delicado, and Fortiana 2010), en comptes d'ajustar n models lineals per la formulació clàssica, aplica la regressió basada en distàncies ponderada vista en la secció 2.2.

Es defineixen dues matrius de distàncies al quadrat entre individus. La primera ($\Delta^2_{(1)}$), caracteritza el pes de les observacions en cada ajust local:

$$\omega_i(\delta) = \frac{K(\delta_1(z_t, z_i)/h)}{\sum_{j \neq t} K(\delta_1(z_t, z_j)/h)},$$

on z_t és el vector que conté els valors en les variables explicatives de l'observació t que es vol estimar i K és una funció Kernel definida (per exemple la d'Epanèchnikov). Per tant, ω_i canvia en cada un dels n models ajustats.

L'elecció de la h té la mateixa importància i dificultats que en la metodologia clàssica i les tècniques d'elecció automàtica són igualment aplicables.

La segona matriu $\Delta^2_{(2)}$ s'utilitza per ajustar tots els DBLM (model lineal basat en distàncies) ponderats. La ponderació és fa tenint present el possible vector de pesos (*weights*) definit inicialment i també els pesos ω_i calculats en cada cas per la funció Kernel.

Ambdues matrius de distàncies poden coincidir o no. Per exemple es pot utilitzar $\delta_1(z_i, z_j) = ||z_i - z_j||$ i $\delta_2(z_i, z_j) = ||(z_i, z_i^2) - (z_j, z_j^2)||$, tal que els pesos Kernels són definits per una mètrica l^2 i el model s'ajusta per regressió local quadràtica. En el cas que les dues mètriques coincideixin i siguin calculades per la funció de distàncies euclidiana, la regressió lineal local basada en distàncies coincideix amb la regressió lineal local usual.

L'estimació per regressió local basada en distàncies és vàlida per a qualsevol conjunt de dades que sigui possible calcular una matriu de distàncies entre individus. Per exemple dades multivariants amb unes variables contínues i d'altres categòriques, dades textuals o dades funcionals.

2.4 Model lineal generalitzat basat en distàncies

Model lineal generalitzat clàssic

En aquest apartat s'estudia una extensió als models lineals usuals de forma que es pot modelitzar una família de models estadístics més general. Es defineix per Nelder and Wedderburn (1972) i abasta la modelització de variables respostes tant numèriques com categòriques. Per tant es poden tractar respostes amb distribucions com la binomial, multinomial, poisson, gamma, etc. A més, el model lineal és un cas particular del model lineal generalitzat amb distribució gaussiana per la variable resposta.

El model lineal usual (LM) es pot expressar com una combinació entre una component sistemàtica i una altra aleatòria

$$Y = Z\beta + \varepsilon,$$

on la primera component és constituïda per una combinació lineal de Z, i que es coneix com a predictor lineal ($\eta = Z\beta$), i la segona està formada pel vector aleatori y, caracteritzat per una distribució normal amb vector d'esperances

μ tal que $E(y) = \mu$ i variança σ^2.

El model lineal generalitzat (GLM) es basa igualment en la combinació de les dues components del LM (sistemàtica i aleatòria), juntament amb una funció d'enllaç que les relaciona.

Per una banda, la component aleatòria està formada pel vector de respostes amb elements independents i idènticament distribuïts que pertanyen a una distribució de la família exponencial uniparamètrica:

$$f(y_i; \theta, \psi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a_i(\psi)} + c(y_i, \psi)\right\},$$

on θ i ψ són els paràmetres de forma i escala, i $a_i(\psi)$, $b(\theta_i)$ i $c(y_i, \psi)$ són funcions conegudes.

Per altra banda, la component sistemàtica bé donada pel predictor lineal de la mateixa manera que en els models lineals ($\eta = Z\beta$). Les components sistemàtica i aleatòria es connecten amb una funció d'enllaç (o link function) $g(\mu)$ tal que

$$\eta = g(\mu) = Z\beta,$$

on Z conté les variables predictores i $g(\mu)$ és la funció d'enllaç que relaciona directament el valor del predictor lineal η amb el valor esperat de la variable resposta $E[y|Z] = \mu$.

La funció $g(\mu)$ és coneguda, monòtona i diferenciable de η, i té inversa

$$\mu = g^{-1}(\eta).$$

En el cas del model lineal, on la component aleatòria és gaussiana, μ està definida en tota la recta real i la funció d'enllaç és la identitat ($\eta = \mu$). Si la distribució és poisson, pel fet que μ només esta definida per a valors positius, un enllaç adequat (encara que no l'únic) és la funció logaritme ($\eta = \ln(\mu)$). Per la distribució binomial, ja que el domini de μ és el interval [0,1] la funció d'enllaç més utilitzada és la coneguda com a funció logit ($\eta = \ln(\mu/(1-\mu))$).

Estimació dels paràmetres

L'estimació dels paràmetres o coeficients β per màxima versemblança no té una expressió analítica tancada, per això s'usen algorismes iteratius per aproximar l'estimador màxim versemblant. El més habitual, encara que no l'únic, és el IWLS: Mínims Quadrats Ponderats Iterats. El software de referència en aquest treball (l'R) estima per defecte els paràmetres dels models lineals generalitzats per IWLS. L'algorisme és pot resumir en els següents passos:

1. Es consideren $r^{(t)}$ com les respostes actives (working responses) en la iteració t tal que:

$$r^{(t)} = \eta^{(t)} + (y - \mu^{(t)}) \left(\frac{d\eta}{d\mu}\right)^{(t)},$$

on $\eta^{(t)}$ s'obté del ajust d'un model lineal ponderat amb pesos $\omega^{(t-1)}$ (step 3) i $\mu^{(t)} = g^{-1}(\eta^{(t)})$. Pels models estàndards (normal, binomial, etc) $\frac{d\eta}{d\mu}$ és fàcil d'implementar. La $r^{(t)}$ és l'aproximació del polinomi de Taylor de primer ordre de la funció d'enllaç avaluada en les dades y.

2. El vector de pesos actius (working weights):

$$\omega^{(t)} = \left[\left(\frac{d\eta}{d\mu}\right)^2 V^{-1(t)}\right],$$

on $V^{(t)}$ és la funció de variança i pot dependre de $\mu^{(t)}$. McCullagh and Nelder (1989) mostren que la variança de la resposta es pot formular com una multiplicació entre la funció de variança ($b''(\theta)$) i un paràmetre de dispersió ($a(\psi)$). Les funcions de variàncies pels models més utilitzats són les següents:

	Normal	Binomial	Poisson	Gamma
$V(\mu)$	1	$\mu(1-\mu)$	μ	μ^2

3. Ajustar un model de regressió ponderat amb resposta $r^{(t)}$, covariables Z i pesos $\omega^{(t)}$. Actualitzar els coeficient $\hat{\beta}^{(t)}$ i processar la nova iteració.

L'algorisme s'atura quan hi ha convergència en la seqüència $\hat{\beta}^{(t)}$ o en el logaritme de la versemblança, és a dir, quan els canvis en l'estimació d'una iteració a l'altra són insignificants. Té importància l'elecció dels valors inicials dels paràmetres. Usualment s'ajusta un model lineal per determinar la $\eta^{(0)}$ i la $\mu^{(0)}$, però hi poden haver casos on sigui recomanable fixar-los a uns altres valors (per exemple, per prevenir en enllaços logístics el possible cas de logaritmes d'un nombre negatiu).

Mesures de bondat de l'ajust

La mesura més utilitzada es forma en base el logaritme del ràtio de versemblances, la desviància. Es considera el model nul, on la variació de la resposta és deguda a la component aleatòria de les dades ($y_i = \bar{y} + \varepsilon_i$). El model complet, en canvi, es dóna quan la variació de la resposta és deguda íntegrament a

la component sistemàtica no deixant lloc a la aleatorietat (sobreajustament).

La idea és avaluar el màxim del logaritme de la versemblança en μ, fixant el paràmetre de dispersió (ψ), tant en el cas extrem del model complet, com en el cas del model que es vol estudiar. La discrepància del model ajustat és proporcional a dues vegades la diferències entre les dues logversemblances:

$$\sum 2z_i \left\{ y_i(\theta(y)_i - \theta(\hat{\mu})_i) - b(\theta(y)_i) + \theta(\hat{\mu})_i \right\} / \psi = D(y; \hat{\mu})/\psi,$$

on $D(y; \hat{\mu})$ és la desviància del model a estudiar, i la funció θ és el link canònic (és enllaç canònic si $\theta = \eta$). Pels models de les famílies de probabilitat que s'estudien, el càlcul de la desviància és reprodueix amb una formula tancada (pàgina 34 McCullagh and Nelder 1989).

Versió del GLM basat en distàncies

El model lineal generalitzat basat en distàncies és una variant del GLM usual on la informació dels predictors està expressada com a distàncies entre individus. El mètode és aplicable per a les mateixes famílies de distribucions que en el GLM. A més, s'hereten totes les funcions inherents a cada família necessàries per a la modelització. És a dir, si es vol modelitzar una variable resposta binària i per tant s'ajusta un DBGLM de la família binomial, la funció d'enllaç, la variança de la component aleatòria, la derivada $\frac{d\eta}{d\mu}$ i la desviància residual tenen la mateixa expressió que en el GLM clàssic. Això és així pel fet que cap d'elles depèn dels valors dels predictors Z directament. La funció d'enllaç només depèn de μ (o η si és la seva inversa). La variança només és funció de μ, la derivada de η i la desviància de y, μ i els pesos ω (que són a la vegada funció de la derivada i la variança).

Només s'ha d'assegurar que és viable obtenir el predictor lineal η quan es té una matriu de distàncies entre individus en el paper de predictor. Un cop estimat η, mitjançant la funció d'enllaç s'estima μ i es tenen tots els elements per poder ajustar un model lineal generalitzat.

En el GLM clàssic, el predictor lineal inicial η^0 s'obté aplicant mínims quadrats tal que $\eta^0 = Z\beta$, o en el cas que no tots els individus tinguin el mateix pes en el model, l'estimació de η és per mínims quadrats ponderats. Anàlogament, si en comptes de Z es té Δ^2 (matriu de distàncies al quadrat) η s'estima ajustant un model de regressió basat en distàncies (DBLM). Com ja s'ha vist en l'apartat 2.2, en el cas que la mètrica utilitzada per calcular les distàncies sigui euclidiana les estimacions del predictor lineal són equivalents

en ambdós mètodes.

Pel que fa a l'algorisme IWLS vist pel cas clàssic per estimar els paràmetres, en el cas del DBGLM perd el sentit estimar β. Els mètodes basats en distàncies són prioritàriament mètodes de predicció. No té sentit preguntar-se l'interpretabilitat de les variables explicatives en la resposta perquè aquestes no hi són, i en conseqüència els coeficients β no es poden estimar. Tot i això, es pot aplicar l'algorisme per tal de conèixer els valors previstos ($\hat{y}$) per una determinada Δ^2 i fer prediccions per a futures observacions. Δ^2 pot ser calculada per a qualsevol funció de distàncies, no únicament per mètrica l^2.

L'algorisme que s'aplica en el GLM basat en distàncies consta dels següents passos:

1. $r^{(t)} = \eta^{(t)} + (y - \mu^{(t)}) \left(\frac{d\eta}{d\mu}\right)^{(t)}$,
2. $\omega^{(t)} = \left[(\frac{d\eta}{d\mu})^2 V^{-1(t)}\right]$,
3. Ajustar un model de regressió ponderat basat en distàncies amb resposta $r^{(t)}$, distàncies al quadrat Δ^2 i pesos $\omega^{(t)}$. Actualitzar η i processar la nova iteració.

Usualment $\eta^{(0)}$ s'estima per regressió basada en distàncies, encara que és possible que prengui un altre valor. Pel que fa al criteri d'aturada, es poden definir dues vies per parar el procés. Quan la $\mu^{(t)}$ convergeix (és a dir, canvia insignificativament d'una iteració a una altra), o bé quan la desviància estimada en cada iteració convergeix.

Predicció d'un nou individu
Donat un nou individu, es defineix $\delta^2_{[n+1]}$ com el vector de distàncies al quadrat de dimensió n que conté les distàncies al quadrat entre el nou individu i els n individus utilitzats per ajustar el model.

La predicció $\tilde{y}_{n+1}$ es troba avaluant el model de regressió lineal ajustat en l'última iteració de l'algorisme IWLS descrit anteriorment per a les noves dades $\delta^2_{[n+1]}$.

2.5 Model lineal generalitzat local basat en distàncies

Model de versemblança local i GLM local

En aquest apartat es veu què és un model de versemblança local, què és un GLM local i com el model lineal local vist a l'apartat 2.3 n'és un cas particular dels dos.

L'objectiu d'un model de regressió no paramètrica generalitzat és trobar una funció suau dels regressors Z tal que

$$E\left[Y_i|z_{i1}, z_{i2}, ...z_{ip}\right] = m\left(z_{i1}, z_{i2}, ..., z_{ip}\right),$$

on $(Y_i|Z = z_i)$ pertany a una família de models estadístics coneguda (binomial, poisson, gamma, normal, etc).

La funció $m(z)$ es pot estimar de manera paramètrica, donant lloc al model paramètric general (GLM) vist en l'apartat 2.4. En molts casos, l'ajust d'un model paramètric no s'adapta bé en tot el domini de les variables regressores, aleshores és quan esdevé útil fer un ajust no paramètric. Si $(y_i|Z = z_i) \sim N(m(z_i), \sigma)$ es tracta del model no paramètric (estimable localment) vist en l'apartat 2.3.

L'idea dels models de versemblança i GLM locals és la mateixa que en el cas particular on la família és gaussiana. Encara que en tot el domini de Z un model lineal paramètric no s'adapti bé, localment, en un entorn d'un punt z_i, una modelització paramètrica pot ser suficient.

Versemblança local
Aplicat en el cas que $(y_i|Z = z_i)$ pertany a una família determinada (per exemple binomial), donat un individu que pren valors en les variables explicatives z_i, en el seu entorn es pot ajustar un model lineal generalitzat amb una funció d'enllaç fixada (per exemple la logística). És a dir, en un entorn de z_i, la resposta y_i segueix un model logístic:

$$y_i = \log\left(\frac{p_i}{1-p_i}\right) = z_i\beta,$$

on β és ajustat pel valor que fa màxim la versemblança, tenint present que

no totes les observacions que entren en el model tenen el mateix pes:

$$l_{z_i}(\beta) = \sum_{j=1}^{n} \omega_j y_j \log\left(\frac{p_j}{1-p_j} + log(1-p_j)\right).$$

Els pesos ω_j són calculats per una funció nucli (o Kernel), o en el cas que inicialment ja hi hagi un vector de pesos donat ($weights$), la interacció entre els pesos per Kernel i els pesos inicials:

$$model.weight_j = \omega_j \cdot weights_j.$$

Trobar els pesos per un Kernel comporta de nou l'elecció de quina funció de Kernel usar (per exemple Triweight) i quin ample de banda (h) és el més encertat per cada cas. Trobant el màxim de la funció de versemblança s'obté l'estimador de β adequat per un veïnatge de z_i: $\hat{\beta}_{z_i}$.

GLM local
La mecànica del GLM local és la mateixa que en el cas de versemblança local, en un entorn de z_i (regressors de l'individu i-èssim) un model de regressió paramètric es pot ajustar bé. La diferència recau únicament en l'estimació de β. En l'apartat 2.4 s'ha vist que no es tenia una fórmula tancada per a l'estimació de β per màxima versemblança. L'alternativa plantejada per McCullagh and Nelder (1989) aplica l'algorisme iteratiu IWLS. Per tant, l'ajust d'un model local lineal generalitzat (LGLM) es tracta d'estimar y_i (per tot i), mitjançant l'ajust d'un model generalitzat ponderat paramètric amb pesos ω_i definits per una funció Kernel i aplicant IWLS per estimar β.

GLM local basat en distàncies

El model lineal generalitzat basat en distàncies (LDBGLM) combina conceptes de tots els mètodes basats en distàncies vistos en apartats anteriors.

El model lineal local basat en distàncies és un cas particular del LDBGLM i la majoria d'aspectes que s'han vist en l'apartat 2.3 on s'estudiava el LDBLM també són vàlids en aquest apartat. Comparteixen conceptes com tenir dues matrius de distàncies al quadrat entre individus (Δ_1^2 i Δ_2^2) i el seu ús és exactament el mateix. Amb la primera es calculen els pesos mitjançant una funció Kernel i amb la segona es codifica la informació de les variables regressores en una matriu de distàncies per tal d'ajustar cada DBGLM local. A més, tenen les mateixes problemàtiques a l'hora d'elegir el valor del paràmetre de suavitzat h (on són utilitzables varies tècniques de selecció automàtica).

Amb el DBGLM comparteixen el fet que modelitzen el mateix. Tot el que es pot modelitzar per un model local generalitzat (LDBGLM), formalment, també es pot modelitzar per un GLM basat en distàncies. De fet, el GLM basat en distàncies és un cas particular del GLM local basat en distàncies on la h és molt gran, i per tant cada model de regressió local s'estima per un DBGLM amb totes les observacions (totes amb el mateix pes).

Per últim, té sentit citar també el model lineal basat en distàncies atès que s'utilitza a l'hora d'ajustar cada model local generalitzat. De fet, l'ajust d'un model LDBGLM, un cop sabuda la distribució de les dades, encadena l'ús del GLM basat en distàncies i el LM basat en distàncies. Per cada individu s'estima localment el valor de y_t amb l'ajust d'un model generalitzat basat en distàncies amb pesos donats per una funció Kernel. Cada DBGLM estimat es basa en l'algorisme IWLS modificat i que s'ajuda de l'ajust de models lineals basats en distàncies (DBLM).

2.6 Mínims quadrats parcials basats en distàncies

Motivació dels mínims quadrats parcials

Mínims quadrats parcials (partial least square o PLS) és un mètode de regressió que es troba a meitat del camí entre els mètodes d'anàlisi de components principals i la regressió múltiple clàssica. Wold (1966) desenvolupa les bases del PLS en l'entorn de les ciències socials (sobretot economia), però ha esdevingut popular en altres camps, com per exemple en la química computacional. De la mateixa manera que en els models lineals o models lineals generalitzats clàssics, modelitza la relació d'una variable resposta Y i una matriu de predictors Z.

El PLS és una eina molt útil quan en la matriu de predictors hi ha més variables que observacions i es produeix el fenomen de multicolinealtitat en Z. En aquests casos la regressió clàssica falla.

Una alternativa és aplicar ACP de la matriu de predictors Z, i considerar les components principals de Z en la nova dimensió com els predictors en la modelització de la resposta. Tot i que d'aquesta manera la multicolinealitat desapareix (al formar projeccions ortogonals), els problemes d'elecció de quantes components s'utilitzen persisteixen. A més, l'elecció es deu únicament a les Z, la variable resposta Y no hi intervé. S'escullen les components que guarden la màxima informació de Z, però això no vol dir que

siguin les més rellevants per explicar la Y.

El PLS, en canvi, té present la relació entre la resposta i els predictors per elegir les components. La idea és buscar una direcció en l'espai multidimensional de Z tal que es maximitzi la covariança entre la resposta i les variables predictores. Posteriorment, ajustar un model lineal amb les variables predictores i la resposta projectades en el nou espai.

Fonaments del PLS

Donades X i y, on X és la matriu de predictors otogonalitzada amb p variables i y és el vector resposta centrat de dimensió n. Sigui $k < p$. Es pot considerar la descomposició de X i de y,

$$X = \hat{X}_k + \tilde{X}_k, \qquad y = \hat{y}_k + \tilde{y}_k,$$

tals que $\hat{X}_k$ i $\hat{y}_k$ són els valors ajustats per X i y, i $\tilde{X}_k$ i $\tilde{y}_k$ són els residus. Els valors ajustats s'estimen de la següent manera:

$$\begin{aligned}\hat{X}_k &= f_1 a_1' + f_2 a_2' + \cdots + f_k a_k',\\ \hat{y}_k &= f_1 b_1 + f_2 b_2 + \cdots + f_k b_k.\end{aligned}$$

Les $f_i \in \mathbb{R}^n$ són conegudes com scores (puntuacions) i a_k i b_k els coeficients de la regressió. Són calculats mitjançant un algorisme recursiu, de manera que f_k, a_k i b_k depenen dels residus anteriors $\tilde{X}_{k-1}$ i $\tilde{y}_{k-1}$ de la següent forma:

$$\begin{aligned}a_k &= (\tilde{X}_{k-1}' f_k)/(f_k' f_k),\\ b_k &= (\tilde{y}_{k-1}' f_k)/(f_k' f_k),\end{aligned}$$

on f_k és la combinació lineal

$$f_k = \tilde{X}_{k-1}\omega_k, \qquad \omega_k = \tilde{X}_{k-1}'\tilde{y}_{k-1}.$$

La nova iteració comença a l'actualitzar els residus:

$$\begin{aligned}\tilde{X}_k &= \tilde{X}_{k-1} - f_k a_k',\\ \tilde{y}_k &= \tilde{y}_{k-1} - f_k b_k.\end{aligned}$$

De fet l'algorisme s'inicia amb els residus $\tilde{X}_0$ i $\tilde{y}_0$ prenen els valors originals de X i y. En cada iteració es troba la projecció de $\tilde{X}_j$ i $\tilde{y}_j$ amb variança màxima, s'actualitza a la component j-èssima de $\hat{X}_j$ i $\hat{y}_j$, i per tal que es compleixi que $X = \hat{X}_j + \tilde{X}_j$ s'extreu de la component j-èssima dels residus. Aquests passos es realitzen successivament fins a la k-èssima iteració, on es decideix parar.

PLS basat en distàncies

En els mètodes basats en distàncies que s'han vist fins al moment, X era considerada la configuració euclidiana centrada calculada a partir de la matriu de distàncies entre individus al quadrat Δ^2. No obstant, per implementar l'algorisme del DBPLS no es necessita X, tot el que depèn de X pot ser configurat per la matriu de productes interiors G, tal que

$$G = XX'.$$

L'algorisme de recurrència vist pel PLS clàssic es pot reescriure en el cas que la informació dels predictors estigui configurada com una matriu de distàncies al quadrat.

En la primera iteració, com ja s'ha vist, $\tilde{X}_0 = X$ i $\tilde{y}_0 = y$. Per tant:

$$f_1 = \tilde{X}_0\omega_0 = \tilde{X}_0\tilde{X}_0'y_0 = XX'y = Gy.$$
$$b_1 = \frac{y'f_1}{f_1'f_1} = \frac{y'Gy}{y'G^2y},$$
$$\hat{y}_1 = f_1b_1 = \frac{Gy(y'Gy)}{y'G^2y}.$$

En termes de $\hat{y}_1$ es pot reescriure tal que quedi representada com la combinació lineal entre una matriu i el vector y. Seria el concepte anàleg a la hat matrix (H) vist en el cas de regressió lineal:

$$\hat{y}_1 = \frac{Gy(y'Gy)}{y'G^2y} = \frac{f_1f_1'}{||f_1||^2}y = P_1y.$$

P_1 és la projecció ortogonal sobre f_1 en l'espai lineal de dimensió 1. El residu de y en la primera iteració és la resta entre y i la $\hat{y}_1$:

$$\tilde{y}_1 = y - \hat{y}_1 = Q_1y,$$

on la Q_1 és el complementari del projector ortogonal P_1, tal que $Q_1 = 1 - P_1$. Similarment amb les ($\hat{y}_1$, $\tilde{y}_1$) i el coeficient b_1 es poden reescriure les ($\hat{X}_1$, $\tilde{X}_1$) i el coeficient a_1:

$$a_1 = \frac{X'f_1}{f_1'f_1} = \frac{X'Gy}{y'G^2y} = \frac{X'f_1}{||f_1||^2},$$
$$\hat{X}_1 = f_1a_1' = P_1X,$$
$$\tilde{X}_1 = X - \hat{X}_1 = Q_1X.$$

En la iteració k-èssima ($k \geq 1$) la recurrència segueix la mateixa estructura. Donada $\tilde{G}_0 = XX' = G$, en el pas k:

$$\tilde{G}_k = \tilde{X}_k \tilde{X}'_k, \qquad P_k = \frac{f_k f'_k}{||f_k||^2}, \qquad Q_k = 1 - P_k.$$

A més, el vector de scores en el pas k (f_k), els residus $\tilde{y}_k$ i la matriu de productes interiors $\tilde{G}_k$ es calculen a partir dels valors de la iteració anterior:

$$\begin{aligned} f_k &= \tilde{G}_{k-1} + \tilde{y}_{k-1}, \\ \tilde{y}_k &= \tilde{y}_{k-1} - f_k b_k = Q_k \tilde{y}_{k-1}, \\ \tilde{G}_k &= Q_k \tilde{G}_{k-1} Q_k. \end{aligned}$$

El vector de valors previstos en la iteració k esdevé la suma dels productes de f_j i b_j

$$\hat{y}_k = f_1 b_1 + \cdots + f_k b_k = \sum_{j=1}^{k} f_j b_j.$$

En el DBPLS també es pot calcular la hat matrix en cada iteració tal que $\hat{y}_k = H_k y$. La hat matrix ve donada per la suma de projectors ortogonals P_j

$$H_k = P_1 + P_2 + \cdots + P_k.$$

Es realitza la demostració. Per definició P_j compleix $f_j b_j = P_j \tilde{y}_{j-1}$, i l'ortogonalitat implica $P_j Q_{j-1} = P_j$. Tenint present que $\tilde{y}_{j-1} = Q_{j-1}\tilde{y}_{j-2}$ es desenvolupa l'equació:

$$\begin{aligned} f_j b_j &= P_j \tilde{y}_{j-1} \\ &= P_j Q_{j-1} \tilde{y}_{j-2} \\ &= P_j \tilde{y}_{j-2} \\ &= P_j Q_{j-2} \tilde{y}_{j-3} \\ &= P_j \tilde{y}_{j-3} \\ &\cdots \\ &= P_j \tilde{y}_0 = P_j y. \end{aligned}$$

La $\hat{y}_k$ es troba, conseqüentment, a partir de la matriu barret H_k:

$$\hat{y}_k = \sum_{j=1}^{k} f_j b_j = \sum_{j=1}^{k} P_j y = H_k y$$

A més la matriu de productes interiors prevista en la iteració k ($\hat{G}_k$) es pot reescriure en funció de la H_k tal que

$$\hat{G}_k = H_k G H_k.$$

Això es pot demostrar tenint present que $f_j a'_j = P_j X$ i $\hat{X}_k = H_k X$, i que per definició $\hat{G}_k = \hat{X}_k \hat{X}'_k$:

$$\begin{aligned}\hat{G}_k &= \hat{X}_k\hat{X}_k' \\ &= H_kXX'H_k \\ &= H_kGH_k.\end{aligned}$$

De la manera que s'ha presentat es reprodueix l'algorisme de PLS clàssic en el cas que no es tingui una matriu de predictors, únicament amb les distàncies entre individus i configurant la matriu de productes interiors G, el procediment pot ser desenvolupat. A més, si s'usa la mètrica euclidiana per calcular la matriu de distàncies al quadrat, el PLS clàssic i el PLS basat en distàncies són equivalents.

Predicció d'un nou individu

Es tracta de fer prediccions sobre la resposta per a una nova observació que té distàncies conegudes respecte als n individus amb els que s'ha construït el model. És pretén obtenir la $\tilde{x}_{n+1}$, vector de coordenades interpoladores, i la predicció $\tilde{y}_{n+1}$

$$\tilde{x}_{n+1} = f_{n+1}A',$$

$$\tilde{y}_{n+1} = f_{n+1}b'.$$

Boj et al. (2007) defineixen l'estimació de $\tilde{y}_{n+1}$ en termes de distàncies de la següent manera:

$$\tilde{y}_{n+1} = \frac{1}{2}(\hat{g} - \delta^2_{[n+1]})FN^{-1}F'y,$$

on $\hat{g}$ és el vector diagonal de $\hat{G}$, δ el vector de distàncies entre els individus del model i l'individu $n+1$, F la matriu de puntuacions o scores obtinguda en el model. Per últim N^{-1} és l'inversa de N, on

$$N = F'\hat{G}F.$$

La demostració es pot veure en Boj et al. (2007) (pàgina 247).

Capítol 3

Realització d'una llibreria en R

Les llibreries (o packages) en R són un mecanisme de recopilació de codi, dades i documentació d'una forma organitzada. Un paquet en R ha de seguir unes pautes molt marcades, tant en el codi com en la documentació. En aquest capítol s'exposa de manera breu els elements més importants que s'han de tenir presents a l'hora de fer una llibreria en R. En el manual de Leisch (2009), es pot estendre la informació presentada.

3.1 Motivació per realitzar una llibreria en R

L'R és un software i també un llenguatge de programació orientat al camp de l'estadística. Es tracta d'un programari de lliure accés, de descarrega gratuïta, creat l'any 1993 i que en els últims anys ha entrat en erupció convertintse en un dels softwares més importants i més utilitzat pels estadístics. L'R usa una interfície de línees de comandes, i el llenguatge que utilitza és una herència del llenguatge de programació S, orientat al C i Fortran.

El punt fort de l'R i el motiu principal que ara sigui un programari capdavanter és que es nodreix del coneixement de tots els usuaris mitjançant els paquets estadístics. Els paquets són una via per agrupar coneixements en format de codi de tal manera que qualsevol usuari els pugui utilitzar. L'R disposa d'una sèrie de paquets base que ja van incorporats en la instal·lació del programa, es tracten dels mètodes estadístics més quotidians i tots els softwares orientats a l'estadística els tenen implementats (per exemple SAS, minitab o SPSS). Aquests mètodes estan en format funció, és a dir, a partir d'uns certs paràmetres d'entrada que dóna l'usuari (o que estan per defecte) es processa un codi, que pot ser visible o no, i genera una informació de sortida. A més a més dels paquets base, se'n disposa de generats pels usuaris, de lliure descàrrega en uns repositoris anomenats CRAN (el més general) o bioconductor (més utilitzat en el sector de la bioestadística).

L'R no deixa de ser un llenguatge de programació, on l'usuari es genera el codi que l'interessa pel que està investigant o processant. La idea dels paquets estadístics és precisament aprofitar-se del coneixement i investigació de cada usuari. El que ha fet un usuari pot ser d'interès per a un altre. Per tant, donant accés al codi evita a l'altre perdre el temps per generar un codi similar. Es fonamenta en la idea que tots els coneixements sumen.

Hi ha un altre aspecte que fa la realització d'una llibreria en R quelcom seductor. Obliga a generar el codi i la documentació en un cert ordre, que pot ser bo no tant sols per a l'ús d'altres persones, sinó per a un mateix. El format d'un paquet estàndard organitza el codi generat en funcions i obliga ser molt curós en la seva documentació. És una bona eina per fer més interpretable el què es pretén fer en cada cas, i sobretot facilita la seva utilització en un temps futur.

Tot i això, no tot és positiu. El problema que tothom pugui fer un paquet és la sobreinformació. En el CRAN hi ha actualment més de 3600 llibreries, i cada setmana se n'incorporen de noves. No hi ha un control en el contingut

de cada una d'elles, pel què molta informació és repetida, i el que encara és més perillós, no es pot tenir la certesa que els procediments implementats siguin vàlids.

3.2 Programació orientada a objectes: S3 i S4

La millor manera d'entendre què és la programació orientada a objectes és mitjançant un exemple. Es pretén construir una base de dades amb els treballadors que té una empresa, un treballador es distingeix per una llista d'atributs, per exemple el seu nom, el DNI, l'edat, el telèfon, sou, el cotxe, etc. El treballador és una classe, i una instància o observació de la classe treballador és un objecte. La classe, per tant, defineix unes característiques comunes entre objectes.

La codificació dels paquets en R, particularment, ha de ser orientada a objectes, definint dos aspectes: classes i mètodes. Hi ha tres sistemes diferents per fer-ho, el S3, S4 i R5. En aquest apartat es discuteixen breument les diferències entre els dos primers, el tercer, pràcticament, no és utilitzat.

Programació en S3

La programació en S3 té un contingut molt buit, però és el que més s'utilitza a l'hora de generar codi. El motiu principal: és molt fàcil d'entendre i de programar. La idea és generar funcions que retornin objectes d'una certa classe amb una llista d'atributs. Per exemple si es vol codificar les operacions bàsiques: divisió, multiplicació, resta i suma entre dos números, una possible funció en S3 seria:

```
operacions.dos.num <- function(a,b)
{
  suma <- a+b
  resta <- a-b
  multiplicacio <- a*b
  divisio <- a/b

  oper <- list(suma=suma,resta=resta,multiplicacio=multiplicacio,
               divisio=divisio)
  class(oper)="operacions.dos.num"
  return(oper)
}
```

Els atributs `suma`, `resta`, `multiplicacio` i `divisio` formen part d'un objecte de la classe `operacions.dos.num` i es poden recuperar amb el caràcter $

(`object$suma`). A més, interessa definir mètodes per a cada una de les classes. Els mètodes són funcions especials aplicables als objectes d'una certa classe (per exemple de la classe `operacions.dos.num`).

En R hi ha uns mètodes molt específics anomenats "*generic methods*", aquests comparteixen el nom per diferents classes però poden tenir definicions diferents. El mètode `print`, per exemple, és una mètode genèric. La sortida per pantalla d'un objecte de la classe `operacions.dos.num` serà diferent al de la classe `treballador`, no obstant, en ambdós casos el que es mostra en pantalla està definit per la mateixa instrucció (`print`). El `print` d'una classe `operacions.dos.num` es pot redefinir:

```
print.operacions.dos.num <- function(x, ...){
   cat("Operacions entre dos números:\n")
   cat("Suma:", paste(x$suma), "\n")
   cat("Resta:", paste(x$resta), "\n")
   cat("Multiplicació:", paste(x$multiplicacio), "\n")
   cat("Divisió:", paste(x$divisio), "\n")
}
```

Un possible exemple on es volen operar els números 3 i 5 el resultat seria:

```
> operacions.dos.num(3,5)
 Operacions entre dos números:
 Suma: 8
 Resta: -2
 Multiplicació: 15
 Divisió: 0.6
```

Un altre cas molt il·lustratiu és el mètode `mean`. Si s'executa l'expressió `methods("mean")` a la línea de comandes d'R s'obté el següent:

```
> methods("mean")
 [1] mean.data.frame mean.Date       mean.default    mean.difftime
 [5] mean.POSIXct    mean.POSIXlt
```

La funció `mean` es pot utilitzar per diferents classes d'objectes. Per exemple, és diferent buscar la mitjana d'un `data.frame` amb 3 variables o columnes que la mitjana d'un vector de números de classe `"numeric"`. Mentre que al primer cas s'espera que surtin 3 números, en el segons només se'n espera un. Els mètodes genèrics més importants i que s'acostuma a redefinir quan es construeix un paquet són els següents:

- `print`: defineix el que es mostra per pantalla d'un objecte d'una classe específica. Es crida amb la instrucció `print(object1)`.
- `plot`: quin(s) gràfic(s) estan definits per un objecte d'una certa classe (`plot(object1)`).
- `summary`: resumeix els resultats més importants d'una certa classe en una nova classe (`summary.operacions.dos.num`).
- `predict`: Predicció per a noves observacions d'una certa classe.

Programació en S4

La programació en S4 és més completa i estricte que la S3, encara que lleugerament més complicada de realitzar. El manual de Genolini (2008) explica molt bé de què es tracta. La separació entre classes i mètodes queda molt clara i estructurada. Per il·lustrar com es defineix una classe en S4 es presenta un exemple, es vol definir una classe `time` que conte el dia, mes i any:

```
setClass(
     + Class="time",
     + representation=representation(
     + dia = "numeric",
     + mes = "numeric",
     + any = "numeric"
     + )
   + )
```

Un objecte particular de la classe `time` es genera amb la instrucció `new`:

```
   avui<-new(Class="time",dia=21,mes=6,any=2011)
```

El `dia`, `mes` i `any` son anomenats *slots* i es poden cridar amb el caràcter @ (`avui@dia`). Igualment que en S3 pot tenir diferents mètodes. El mètode genèric `show` n'és un exemple:

```
 > setMethod("show","time",
    + function(object){
    + cat("*** en quin segle som? *** \n")
    + cat("* any =", paste(object@any[1]),"\n")
    + segle <- trunc(any/100)+1
    + cat("* segle =",paste(segle[1]),"\n")
   }
 )
```

En el cas de mostrar el dia avui s'obté:

```
> avui
  *** en quin segle som? ***
  * any = 2011
  * segle = 21
```

Comparació entre S3 i S4

La programació en S4 és molt utilitzada per packages al repositori bioconductor, en el CRAN la majoria són en S3. Tot el que es pot fer en S3 es pot fer en S4 i l'inrevés també, encara que d'una forma menys elegant. El llenguatge S4 és més formal i rigorós en la majoria d'aspectes, per exemple comprova automàticament si els atributs que l'usuari introdueix en un objecte d'una certa classe són correctes. En l'exemple de classe `time` vist anteriorment, si l'atribut `dia` fos introduït com un caràcter (`"dimecres"`) el programa pararia directament:

```
> avui<-new(Class="time",dia="dimecres",mes=6,any=2011)
   Error en validObject(.Object) :
    invalid class "time" object: invalid object for slot
     "dia" in class "time": got class "character", should
     be or extend class "numeric"
```

Si es vol fer aquesta comprovació en S3, s'ha de definir a dins la funció:

```
if(class("object1")!="correct.class")
        stop("explicació de perquè s'ha parat el procés")
```

Tot i que S4 sembla més potent, per a la majoria d'implementacions programar en S3 és suficient i facilita molt la seva aplicació als usuaris, que al final és el que interessa. És més, la gran part de consumidors de l'R desconeixen el sistema S4, pel que sembla raonable fer les contribucions en S3 fins que això no canviï.

3.3 Programes necessaris en Windows per elaborar una llibreria

Per poder elaborar una llibreria des d'una computadora amb sistema operatiu Windows és necessari tenir instal·lat els següents programes (o utilitats).

- Eines executables per línees de comandes : Les Rtools.
- MinGW-w64 32/64-bit si es vol compilar codi C, Fortran i C++.
- LaTeXper compilar PDF.

Aquestes eines depenen de la versió d'R que s'utilitzi, aquí es tracta el cas de la darrera versió fins el moment, l'R-2.15.0.

Modificació del PATH

Es important que en el PATH del sistema s'hi especifiquin les eines instal·lades.

```
PATH = %PATH%;C:\windows;C:\windows\system32;
       C:\R\R-2.15.0\bin;c:\Rtools\bin; C:\Rtools
       \perl\bin; C:\Rtools\MinGW;C:\MiKTeX 2.9\
       miktex\bin;
```

No en tots els ordinadors els programes queden instal·lats als mateixos directoris, pel que s'ha de cercar en cada cas la ubicació exacta de la carpeta bin. A més, cal anar molt en compte a l'hora d'escriure exactament els noms de les carpetes. Aquests són alguns aspectes remarcables a tenir en compte:

- En Windows és equivalent escriure els noms de les carpetes en minúscules i en majúscules.
- Pot passar que a l'instal·lar els programes, algunes carpetes ja estiguin incloses al PATH. Cal mirar que la versió que es vol utilitzar sigui la primera al PATH.
- Es busca la subcarpeta bin ja que conté els fitxers executables.
- Rutes on el nom dels fitxers contengui algun signe no Anglès (per exemple els signes d'accentuació) poden provocar errors (no troba la ruta adequada al no interpretar bé aquests signes).

El PATH en el sistema operatiu Windows es pot canviar de forma definitiva si es segueixen els següents passos:

1. Anar al Tauler de Control → sistema.
2. Fer clic a configuració avançada dels sistema → avançat → variables d'entorn.
3. Buscar en variables del sistema la variable PATH i enganxar l'expressió anterior.

Instal·lació de les Rtools, MinGW toolchain i LaTeX

- El distribuïdor de LaTeX, el MiKTeX (http://www.miktex.org/) ja inclou un port adequat de pdflatex. Amb la instal·lació bàsica és suficient.

- Les R-tools es poden obtenir accedint a la web http://cran.r-project.org/bin/windows/Rtools/. S'ha de mirar la versió adequada a descarregar (en l'R-2.15.0 s'ha de descarregar les Rtools215).

- En la versió comercial de l'R-2.15.0 les toolchain formen part del Rtools215.exe. Aquesta usa una versió beta del gcc 4.6.3 i la versió 2.0.1 del MinGW-w64 (les dues components han d'estar especificades al path).

3.4 Estructura d'una llibreria

Un paquet en R consisteix en un subdirectori que conté una sèrie d'arxius: el DESCRIPTION, INDEX, NAMESPACE, configure, cleanup, license, news. A més conté els subdirectoris: R, data, demo, exec, inst, man, po, src, i tests.

El fitxer DESCRIPTION i els subdirectoris R i man són obligatoris per a qualsevol paquet en R. El NAMESPACE també és necessari en la majoria dels cassos. Els altres són opcionals.

Mitjançant la funció d'R `package.skeleton` l'estructura de la llibreria es crea directament.

```
package.skeleton(name = "paquet.exemple", list,
                 environment = .GlobalEnv,
                 path = ".", force = FALSE, namespace = TRUE,
                 code_files = character())
```

L'atribut `name` indica el nom del paquet i a la vegada el nom del subdirectori pel paquet. En el `list` es citen els noms de les funcions que conté el paquet (han de ser creades en la sessió de R abans d'executar la instrucció `package.skeleton`). El `environment` és l'entorn dels objectes (normalment es deixa per defecte). En el `path` es declara el directori on es crearà el paquet. L'argument `force` és un booleà que val `TRUE` si xafa el directori anterior o `FALSE` en cas contrari. S'ha de fixar el `namespace` a `TRUE` si es vol crear tal fitxer i `FALSE` en cas contrari. El `codefiles` és l'alternativa al `list` per

indicar quin codi contindrà el paquet. Es dóna el camí (path), que conté els fitxers d'extensió .r amb el codi, entre cometes. La carpeta creada conté els següents fitxers i subdirectoris:

1. Els arxius de definició i informació:
   - **Read-and-delete-me** Informació indicativa per a la creació d'una llibreria.
   - **DESCRIPTION** Descripció de les característiques principals del paquet.
   - **NAMESPACE** Importa paquets necessaris i exporta les funcions generades.

2. Les carpetes amb el següent contingut:
   - **R** Funcions de la llibreria.
   - **man** Ajuda o helps de les funcions i dades generades.
   - **src** Codi en un llenguatge de baix nivell com FORTRAN (opcional).
   - **data** Base de dades del paquet (opcional).

Fitxer Description

El fitxer Description conté la informació bàsica del paquet. En destaquen els següents punts i format:

```
Package:     pkgname
Version:     0.0.1
Date:        2012-06-06
Title:       el meu paquet estadístic
Author:      autor 1 i autor 2
Maintainer:  autor 1 <autor1@domini.net>
Depends:     R (>= 1.8.0), altres.paquets
Suggests:    MASS
Description: Descripció del que fa el paquet
License:     GPL (>= 2)
URL:         http://www.r-project.org, http://www.another.url
```

Els camps `Package`, `Version`, `License`, `Description`, `Title`, `Author`, i `Maintainer` són obligatoris, els altres camps són opcionals.

En el camp `Package` s'hi ha de citar el nom del paquet (només pot contenir lletres, números i punts). En el camp `Version`, la versió del package (usualment números positius separats per un punt o una barra (. , -)). En `License` ha de contenir una o un conjunt de les següents llicències :

```
GPL-2 GPL-3 LGPL-2 LGPL-2.1 LGPL-3 AGPL-3 Artistic-1.0
Artistic-2.0
```

El $GPL(>= 2)$ de l'exemple ja funciona bé. El camp `Description` ha de contenir una descripció general del què fa el package. En el `title` s'especifica una descripció molt breu del què fa la llibreria (un màxim de 65 caràcters). L'atribut `Author` indica qui ha escrit el paquet, i el `Mantainer` el destinatari de les preguntes dels usuaris respecte el funcionament del paquet. El fitxer `DESCRIPTION` per anar bé ha de ser escrit en format ASCII.

Fitxer Namespace

S'usa per carregar les llibreries que necessita el paquet (import) i per indicar quines de les funcions/variables realitzades es vol que siguin públiques i usades pels altres usuaris (export).

- `export(f,g)` per exportar les funcions f i g.
- `import(paq1,paq2)` per importar funcions dels paquets paq1 i paq2.
- `importfrom(paq1,f,g)` per importar del paquet paq1 les funcions f i g.
- `S3method(print,f)` per registrar mètodes S3 (per exemple el `print`).

3.5 Documentació de les funcions i dades

La documentació és una de les parts més importants en la creació d'un paquet. Es tracta d'explicar el què has codificat d'una manera intel·ligible. Si la documentació és pobre o inexistent la llibreria no es podrà utilitzar.

Els fitxers d'ajuda es troben en la carpeta man, i tenen una extensió Rd (R documentation). Per a cada una de les funcions implementades en la carpeta R o conjunt de dades en la carpeta dades s'ha d'escriure un document d'ajuda per a la seva utilització. S'ha de diferenciar entre funcions internes o externes, l'ajuda és diferent en un i altre cas. Les funcions externes són totes aquelles indicades en el namespace mitjançant l'expressió `export`, les demés són internes. Es poden definir les funcions internes com funcions auxiliars que no tenen un interès remarcable de cara el públic, però són estrictament necessàries pel funcionament de les externes.

El llenguatge que utilitza l'R per la generació d'aquests fitxers Rd és una mica especial, i el pas de fitxers d'extensió Rd a html, LaTeX o text ho executa ell directament. El llenguatge s'aproxima al LaTeX d'una forma evident, tot i que no és necessari saber LaTeX per escriure els documents d'ajuda. És més, si s'ha generat l'estructura del paquet per la comanda de R `package.skeleton` en la carpeta man ja apareixen amb una plantilla Rd tots els fitxers que precisen del help.

Un fitxer de documentació Rd es pot dividir en tres parts. En la primera part és demana la informació bàsica de l'objecte: el nom, el títol, una petita descripció i la informació necessària per a la seva utilització. En la segona, el cos del document, descriu la informació d'entrada i de sortida del objecte. De forma opcional, es pot escriure una secció de detalls, que sobretot en la documentació de funcions és útil per acabar d'explicar el seu ús, problemes que pot tenir, i qualsevol informació secundària que acabi de fer entenedor el significat de la funció. A més, es sol donar una sèrie d'exemples d'utilització de l'objecte. Per últim la tercera part és opcional i conté les paraules claus (keywords).

Les ajudes estan separades per seccions, amb un nom estàndard fixat, encara que també hi ha la possibilitat de definir-ne de noves. Una de molt usual és la secció *Warnings*, en ella s'alerta de quines situacions l'execució de l'objecte pot fallar:

```
\section{Warning}{
        You must not call this function unless ...
      }
```

Documentació d'una funció

Un document d'ajuda de funcions conté normalment les següents seccions:

- `\name{nom}` : Nom de l'objecte.
- `\alias{topic}`: Per utilitzar el mateix fitxer d'ajuda per varis objectes. Un cas d'exemple serien les funcions `"rnorm"`, `"dnorm"`, `"pnorm"`, `"qnorm"`.
- `\title{Títol}`: Títol informatiu del fitxer Rd (no més llarg de 65 caràcters.

- `\description{...}`: Breu resum del que fa la funció. Si es necessita més espai per descriure la problemàtica i l'ús es pot utilitzar la secció `\details`.
- `\usage{fun(arg1,arg2,...)}`: Especifica com es defineix la funció. La informació en la secció `usage`, ha de definir la funció exactament igual que en el codi R (sinó no compilarà).
- `\arguments{...}`: Descripció dels arguments o paràmetres de la funció: `\item{arg_i}{Descripció de arg_i}`.
- `\details{...}`: Extensió de la informació de la secció `\description`.
- `\value{...}`: Descripció del què retorna la funció. Si retorna una llista amb múltiples valors es poden escriure de la forma següent: `\item{val_i}{Descripció de val_i}`.
- `\author{...}`: Informació sobre els autors.
- `\seealso{...}`: Mitjançant la instrucció `\code{\link{...}}` crea un link a l'objecte descrit entre claus.
- `\examples{...}`: Exemples de com usar la funció. S'escriu en el format R, no Rd. Per tant el que hi hagi en l'apartat d'exemples s'expressa com a codi R.
- `\keyword{clau}`: Pot no haver-hi cap paraula clau, o més d'una. Si s'escriu més d'una clau, s'ha d'especificar més d'una secció keyword on cada una només contingui un element.

En la següent taula es presenta una plantilla de document d'ajuda d'una funció externa:

```
% Exemple de documentació Rd de una funció externa.
  \name{exemple}
  \alias{rexemple}
  \title{Exemple de documentació}
  \description{
     Això és una funció externa d'exemple per apendre a elaborar fitxers de
     documentació Rd.
  }
  \usage{
     exemple(parametre.1, parametre.2 = 2, parametre.3 = c("a","b"), ... )
  }
  \arguments{
     \item{parametre.1}{què és el paramètre.1, numèric o caràcter, què
                        significa en la funció, etc.}
     \item{parametre.2}{el paràmetre.2 és numèric, i pren per defecte
```

```
                    el valor 2, etc}
   \item{parametre.3}{el paràmetre.3 és un vector de caràcters, i pren per
                    defecte el valor c("a","b"), etc}
}
\details{
    La funció d'exemple es pot utilitzar sempre,
         serveix especialment per ...
}
\value{
        Conté una llista amb els següents elements:
   \item{value.1}{què és el value.1}
   \item{value.2}{què és el value.2}
}
\author{
   Adrià Caballé Mestres <adria.caballe@upc.edu>
}
\seealso{
  \code{\link{una.altre.funció}}.
}
\examples{
  ## utilització funció exemple I
  exemple(paràmetre.1 = 1,paràmetre.2 = 10)
  ## utilització funció exemple II
  exemple(paràmetre.1 = 1,paràmetre.3 = c("r","k"))
}
\keyword{file}
```

Les funcions externes poden dependre de funcions internes que no es volen documentar, per tal que l'R no es queixi aquestes s'han de presentar en un fitxer Rd com en el següent exemple:

```
% Exemple de documentació Rd de funcions internes.
 \name{funcio.externa-inl}
 \alias{funcio.externa-internal}
 \alias{funcio.interna.1}
 \alias{funcio.interna.2}
 \alias{funcio.interna.k}
 \title{Internal  functions}
 \description{Internal  functions}
 \usage{
    funcio.interna.1(a,b,c)
    funcio.interna.2(d,e)
    funcio.interna.k(a)
 }
\details{Not to be called by users}
\keyword{internal}
```

Documentació d'un conjunt de dades

L'estructura de documentació d'un conjunt de dades és lleugerament diferent a la vista per les funcions. Les seccions que defineix són les següents:

- `\name{nom}` : Nom del fitxer de dades.

- `\docType{...}`: Indica el tipus de fitxer de documentació (`data` per conjunt de dades)
- `\alias{topic}`: Altres objectes amb el mateix fitxer d'ajuda.
- `\title{Títol}`: Títol informatiu del conjunt de dades.
- `\description{...}`: Descripció del contingut de les dades.
- `\usage{nom}`: Com es crida el conjunt de dades.
- `\format{...}`: Descripció del format del conjunt de dades. S'especifica si és un vector, matriu, *data.frame*, etc.
- `\source{...}`: Detalls de la procedència de les dades.
- `\references{...}`: Referències, permet complementar la informació de la secció `source`.
- `\keyword{datasets}`: opcional, moltes vegades no es descriu.

En la següent taula es presenta una plantilla de document d'ajuda d'un conjunt de dades.

```
% Exemple de documentació Rd de conjunt de dades.
       \name{data.exemple}
       \docType{data}
       \alias{data.exemple}
       \title{Exemple de conjunt de dades}
       \description{
          Que conté el data set ...
       }
       \usage{data.exemple}
       \format{A vector containing k observations.}
       \source{llibre: per exemple
            World Almanac and Book of Facts, 1975, page 406.}
       \references{
    McNeil, D. R. (1977) \emph{Interactive Data Analysis}.
    New York: Wiley.
       }
       \keyword{datasets}
```

Instruccions d'interès

Hi ha una sèrie d'instruccions en format Rd que poden ser d'utilitat per fer més agradable la lectura d'un document d'ajuda. En el capítol 2 del manual R extensions estan descrites totes. Les més destacades són les següents:

`\emph{text}`	El text entre claus esdevé en font *italic*.
`\bold{text}`	El text entre claus esdevé en **negreta**.
`\code{text}`	Indica que el text és un exemple literal de `codi R`.
`\pkg{package_name}`	Indica el nom del paquet en format LaTeX.
`\email{email_address}`	Indica una adreça de correu electrònic.
`\cite{reference}`	Indica una referència sense un accés directe (link) amb format LaTeX.
`\link[pkg]{funtion}`	Indica un enllaç directe a la funció descrita del paquet indicat.

Depuració dels fitxers Rd

A l'hora de compilar un paquet en R, els missatges d'error pels arxius de documentació poden ser realment difícils de desxifrar. El consell és generar els fitxers d'ajuda de forma seqüencial. És a dir, no escriure tots els documents al mateix temps fins que no es tingui una certa pràctica. Sinó és així, el depurat de tot el conjunt serà molt difícil que funcioni a la primera i costarà molt trobar a on s'han produït els errors. Verificar un a un els documents Rd és possible mitjançant una de les següents instruccions en la consola de Windows:

```
R CMD Rd2txt funcio1.Rd     R CMD Rd2pdf funcio1.Rd
```

Et genera un fitxer txt o pdf, respectivament, en el cas que el document Rd estigui realitzat correctament.

Si un cop definida l'estructura del paquet amb la instrucció `package.skeleton`, es vol afegir una nova funció o conjunt de dades, és pot fer directament amb l'R. En el cas de les dades s'utilitza la funció `save()`:

```
save(dades, file="C:/paquet.exemple/data/dades.rda")
```

En el cas de voler gravar codi R, s'utilitza la funció `dump()`:

```
dump("funcio.1", file="C:/paquet.exemple/R/funcio.1.R")
```

Finalment, per redactar el fitxer de documentació associat a l'objecte creat s'usa la funció `prompt()`:

```
prompt(funcio.1,filename="C:/paquet.exemple/man/funcio.1.rd")
```

3.6 Compilació del paquet

Un cop s'hagin realitzat tots els documents d'ajuda, i els fitxers description i namespace, es pot procedir a construir el paquet. Amb Windows el procés de creació és fa mitjançant la consola CMD (indicador d'ordres). Primer de tot s'aconsella "*verificar*"el paquet, és a dir, provar si el paquet podria ser instal·lat correctament. Des de la consola, s'ha d'estar ubicat en el directori que conté la carpeta amb la llibreria i escriure la següent instrucció:

```
R CMD check paquet.exemple
```

El procés `check` crea els arxius de documentació en format txt, LaTeXi html. Compila el codi R, comprova que no hi hagin errors ni inconsistències entre codi i documentació, executa els exemples realitzats en les ajudes i adjunta un manual pdf amb tota la documentació. Tots els elements creats es poden trobar en una carpeta `paquet.exemple.Rdcheck`. Si hi ha errors, i no es pot compilar la prova `check`, es crea un arxiu `00install` amb la indicació dels errors que s'ha trobat. Aquests poden ser realment críptics, pel què és important efectuar el check un cop ja s'hagi comprovat que tota la documentació és correcta. Cal tenir molta cura a l'hora d'escriure els fitxers description i namespace.

El següent pas, és construir la llibreria. Es realitza amb la instrucció en la consola

```
R CMD build paquet.exemple
```

Es crea un fitxer tar.gz que és el que s'ha d'enviar per ser actualitzat al CRAN. De cara als amics i companys de feina que es vulgui distribuir el paquet, també és possible la seva creació amb un fitxer .zip. Es realitza mitjançant una de les següents instruccions:

```
R CMD build --binary  paquet.exemple
R CMD INSTALL --build paquet.exemple (recomanada)
```

El paquet, un cop ja creat, es pot instal·lar directament des de la consola amb la següent instrucció:

```
R CMD INSTALL paquet.exemple.
```

Alternativament, des d'R, amb el paquet .zip, també es pot realitzar. Entrant en R, anant al menú Packages → Install package(s) from local zip files i seleccionant el fitxer .zip en qüestió. De les dos maneres, es pot carregar el paquet amb la instrucció R `library(paquet.exemple)` i ja està disponible per a la seva utilització.

3.7 Distribució del paquet a CRAN

CRAN és una xarxa WWW que conté codi R distribuït, especialment els paquets d'R. Un cop testat el paquet (sobretot és important passar el check), es pot enviar al repositori CRAN. S'ha d'adjuntar el fitxer tar.gz a la direcció de correu ftp://cran.R-project.org/incoming, amb nom d'usuari anonymous i password el correu electrònic. Posteriorment s'ha d'enviar un missatge a cran@r-project.org informant sobre la incorporació del paquet. Els mantainers del CRAN executaran una sèrie de tests abans de posar el paquet en el repositori principal. Si tot va com és d'esperar, en menys d'una setmana el paquet ja estarà disponible al CRAN.

Capítol 4

Llibreria dbstats

En aquest capítol és on es posa de manifest tot el que he aportat de nou al realitzar aquest projecte. Quan em vaig incorporar en el grup de treball, tots el mètodes estudiats en el capítol 2 ja estaven publicats en revistes. També es trobaven programades una sèrie de subrutines (la majoria en R) amb els procediments bàsics per a cada una de les metodologies. La llibreria `dbstats` intenta agrupar tot aquest codi de forma organitzada, documentat de manera intel·ligible i orientat a l'ús d'usuaris acostumats als mètodes usuals d'R com el `lm`, el `glm` o el `plsr`. Aspectes com definir mètodes genèrics (el `print`, `plot`, `summary`, etc), així com implementar tècniques de selecció automàtica de la dimensió efectiva (pel `dblm`), de l'ample de banda òptim (pel `ldblm` i `ldbglm`) o el nombre de components adequades (pel `dbplsr`) han estat un punt afegit als procediments que s'havien desenvolupat prèviament. Al llarg del capítol s'entrarà en detall en tots aquests conceptes procedimentals.

`dbstats`, Boj, Caballé, Delicado, and Fortiana (2012), és una llibreria en R que conté l'aplicació de mètodes estadístics basats en distàncies. Aquests són mètodes de predicció on la informació de les variables explicatives està codificada com una matriu de distàncies entre individus. A més, la resposta, a l'igual que en els models lineals clàssics, és una variable unidimensional. El `dbstats` permet a l'usuari introduir la informació de la matriu de distàncies per diferents vies, directament expressada com una matriu de distàncies (Δ), així com calculada a partir d'una matriu de distàncies al quadrat (Δ^2), a partir de la matriu de productes interiors (G), o pel cas més proper als models usuals, a partir de les variables explicatives (Z).

Les conversions entre una matriu de distàncies al quadrat i una matriu de productes interiors, i també d'una matriu de distàncies al quadrat i una matriu de distàncies, estan programades en el paquet. Es discuteix en la secció 4.1. En canvi, `dbstats` no conté cap funció per passar directament de variables explicatives a distàncies, donat que hi ha altres paquets i funcions en R que fan aquesta tasca. Es recomanen les següents:

- Funció `dist` del paquet base `stats`.
- Funció `dist` del paquet `proxy`.
- Funció `daisy` del paquet `cluster`.

En la secció 4.2 s'entra en detall amb el seu contingut.

Les funcions principals programades al `dbstats` són les descrites de forma metodològica en el capítol 2:

1. Models lineals i locals lineals amb resposta contínua:
    - `dblm`: distance-based linear model (secció 4.3).
    - `ldblm`: local distance-based linear model (secció 4.4).
    - `dbplsr`: distance-based partial least squares regression (secció 4.7).
2. Model lineal generalitzat i local lineal generalitzat:
    - `dbglm`: distance-based generalized linear model (secció 4.5).
    - `ldbglm`: local distance-based generalized linear model (secció 4.6).

L'ajust d'un model basat en distàncies utilitzant la llibreria `dbstats`, per exemple el `dblm`, genera un objecte d'una certa classe (`dblm`), amb l'opció d'utilitzar els mètodes genèrics: `print`, `summary`, `plot` i `prediction`.

En tot moment s'ha intentat que, tant en la manera d'introduir la informació d'entrada en les funcions descrites, com en el contingut de sortida un cop ja executades, els mètodes del `dbstats` siguin tant semblants com sigui possible als mètodes usuals d'R: `lm` (pel `dblm`), `glm` (`dbglm`), `loess` (`ldblm`) i `plsr` (`dbplsr`).

Cal assenyalar alguns aspectes de notació. Per tal de ser coherents amb el codi R i la documentació de `dbstats`, es distingeix entre variables explicatives observades Z o z, davant coordenades euclidianes X o x (es justifica en el capítol 2). A més, la matriu de distàncies es denota com a D i la de distàncies al quadrat com a $D2$. Per últim, els productes interiors s'expressen com a G o $Gram$.

El `dbstats` està disponible al repositori d'R CRAN. L'únic que s'ha de fer per poder utilitzar els seus mètodes és descarregar el paquet, instal·lar-lo i posteriorment, amb la instrucció en la línea de comandes d'R

```
library(dbstats)
```

carregar la llibreria.

4.1 Conversions: `GtoD2`, `D2toG`, `disttoD2` i `D2toDist`

En els mètodes basats en distàncies la informació de les variables explicatives es codifica amb una matriu de distàncies. A més, ja s'ha vist en l'apartat 2.1 que $G = XX'$, matriu de productes interiors de les coordenades euclidianes, només depèn de la matriu de distàncies entre individus al quadrat: $G = -\frac{1}{2}J\Delta^2 J$, on J és la matriu de centrat. Es defineixen els objectes que contenen els productes interiors G com objectes de la classe `Gram`. En aquest apartat s'estudia com `dbstats` permet fer conversions entre objectes `D2` i matrius de productes interiors (G), i entre objectes `D` i `D2`. Els objectes de classe `D2` i `Gram` són de la següent manera:

- `D2`: una matriu quadrada, simètrica, amb elements no negatius i zeros en la diagonal.
- `Gram`: una matriu quadrada i simètrica.

D2toG

Es suposa coneguda (com en la majoria de casos) la matriu de classe `D2` amb les distàncies al quadrat entre individus, així com el vector numèric de pesos `weights` que defineix la importància *a priori* que ha de tenir cada individu en el model (per defecte tots els individus tenen el mateix pes). Mitjançant la funció `D2toG` es pot transformar un objecte de classe `D2` a un objecte de classe `Gram` que conté els producte interiors centrats respecte la matriu de classe `D2` i ponderats pels pesos en `weights`:

```
D2toG(D2,weigths)
```

Aquí es mostra un exemple:

```
> X <- matrix(rnorm(100*3),nrow=100)
> D2 <- as.matrix(dist(X)^2)
> class(D2)<-"D2"
> G <- D2toG(D2,weights=NULL)
```

GtoD2

Es suposa conegut un objecte `G` de la classe `Gram`, aquest conté els productes interiors centrats i ponderats de la matriu de distàncies al quadrat $D2$, l'objectiu és trobar quant val $D2$. La funció del paquet `dbstats` que ho realitza es crida amb la comanda

```
GtoD2(G)
```

Donada la matriu $n \times n$ de classe `Gram` com a paràmetre d'entrada, retorna una matriu $n \times n$ de classe `D2` amb les distàncies al quadrat entre individus. Continuant l'exemple anterior:

```
> class(G)<-"Gram"
> D22 <- GtoD2(G)
```

disttoD2

Converteix un objecte de classe `dist` o `dissimilarity`, que conté els elements situats per sota de la diagonal de la matriu de distàncies/dissimilituds entre individus, a un objecte de la classe `D2` amb la matriu de distàncies al quadrat. Es realitza mitjançant la funció

```
disttoD2(distance)
```

Un exemple seria:

```
> X <- matrix(rnorm(100*3),nrow=100)
> distance <- daisy(X,"manhattan")
> D2 <- disttoD2(distance)
```

D2toDist

Donada una matriu de distàncies al quadrat, retorna un objecte de classe `dist`:

```
D2toDist(D2)
```

Continuant l'exemple:

```
> distance2 <- D2toDist(D2)
```

4.2 Matrius de distàncies: funcions `dist` i `daisy`

La conversió que queda per definir és la que associa una matriu $n \times p$ de variables explicatives Z a una matriu $n \times n$ amb les distàncies entre individus (o dissimilituds). Aquest aspecte no està codificat directament en el paquet `dbstats` atès que hi ha altres funcions definides en R que estudien extensament com quantificar la proximitat entre objectes. En aquest apartat s'il·lustra quines alternatives es tenen a l'hora de codificar la Z com a matriu de distàncies. S'analitzen les funcions d'R `dist` del paquet `stats` (inclòs en la instal·lació bàsica de l'R), `daisy` del paquet `cluster` i `dist` del paquet `proxy`.

Funció `dist` del package `stats`

La funció `dist` retorna una matriu de distàncies entre files (de classe `"dist"`) respecte una matriu numèrica $n \times p$ i a partir d'una certa funció de distàncies. S'utilitza de la següent manera:

```
dist(x=Z, method = "euclidean", diag = FALSE,
     upper = FALSE, p = 2)
```

L'atribut `x` és una matriu numèrica $n \times p$, en format `matrix` o `data.frame`. En la nomenclatura `dbstats`, el paràmetre `x` defineix la matriu de predictors

Z. El `method` determina la funció de distàncies que es pretén utilitzar (per defecte l'euclidiana), l'atribut `diag` indica si es vol que s'escrigui per pantalla els valors de la diagonal de la matriu. Si es tracta d'una distància ben definida (es justifica en l'apartat 2.1) aquests han de valer 0:

$$d(Z_i, Z_j) = 0, \qquad \text{si } i = j.$$

En `upper` s'especifica si es vol mostrar per pantalla el triangle superior. Si es tracta d'una distància ben definida ha de ser igual a la trasposta del triangle inferior de la matriu:

$$d(Z_i, Z_j) = d(Z_j, Z_i)$$

Per últim, `p` determina el grau de la norma de la distància de Minkowski, utilitzada únicament si l'atribut `method="Minkowski"`.

Les mètriques disponibles en la funció `dist`, donats dos individus que prenen valors z_i i z_j, són les següents:

- `euclidian` (per defecte): distància de norma dos entre z_i i z_j:

$$\sqrt{\sum_{r=1}^{p}(z_{ir} - z_{jr})^2}.$$

- `maximum`: distància màxima entre els dos vectors z_i i z_j. També coneguda com a norma del màxim:

$$\max(|z_{i1} - z_{j1}|, |z_{i1} - z_{j1}|, \cdots, |z_{ip} - z_{jp}|).$$

- `manhattan`: suma de les diferències en valor absolut entre els dos vector z_i i z_j. També coneguda com a norma 1:

$$\sum_{r=1}^{p} |z_{ir} - z_{jr}|.$$

- `canberra`: similar a la distància de Manhattan, s'utilitza per casos on les dades són molt disperses en l'entorn de l'origen:

$$\sum_{r=1}^{p} \frac{|z_{ir} - z_{jr}|}{|z_{ir} + z_{jr}|}.$$

- `binary`: els vectors són considerats com a bits binaris tal que si el valor de l'element és no-zero val `"on"` i si és 0 val `"off"`. La distància entre p i q mesura la proporció de símbols diferents entre les dos seqüències de bits.

- `minkowsky`: es tracta de la distància de norma p:

$$(\sum_{r=1}^{p} |z_{ir} - z_{jr}|^p)^{1/p}.$$

Funció `daisy` del package `cluster`

La funció `daisy`, de la mateixa manera que la funció `dist`, retorna una matriu de dissimilituds entre parelles d'elements (de classe `"dissimilarity"` i `"dist"`) utilitzant una funció de distància (mètrica) específica. La comanda d'R és la següent:

```
daisy(x=Z, metric = c("euclidean", "manhattan", "gower"),
      stand = FALSE, type = list(), weights = rep.int(1, p))
```

L'atribut `x` (que en la nomenclatura `dbstats` indiquen els predictors Z) ha de ser una matriu o `data.frame` $n \times p$ tal que les dissimilituds són calculades a partir de les files de `Z`. La `Z`, contràriament a la funció `dist`, pot contenir variables numèriques, nominals, variables ordinals o fins i tot algun altre tipus de variable (s'ha d'especificar en l'atribut `type`). El paràmetre `metric` determina la mètrica utilitzada. `stand` indica si la mesura en `Z` és estandarditzada o no, si val TRUE, Z_{ip}, corresponent al valor de l'individu i-èssim en la variable p, esdevé

$$Z^*_{ip} = \frac{Z_{ip} - \bar{Z}_{.p}}{(1/n)\sum_{j=1}^{n} |Z_{jp} - \bar{Z}_{.p}|}.$$

En `type` és on s'especifica la classe de variables que hi ha en `x`, les variables possibles són: `"ordratio"`: variables d'escala de raó ordinals. Aquestes es defineixen en un interval continu on l'ordre és important. Tenen la condició que el punt 0.0 indica una absència de valor en la variable (per exemple el pes o l'alçada són variables `ordratio`). `"logratio"`: variables d'escala de raó que se'ls aplica la transformació logarítmica. `"asymm"`: variables binàries asimètriques (0/1) i `"symm"`: variables binàries simètriques (-1/1). Altrament, es considera que la variable és numèrica, factor nominal o factor ordinal. `type` és una llista de vectors, on cada element de la llista és un tipus (per exemple `"symm"` o `"ordratio"`), i s'indica el tipus de cada columna assignant-lo a l'element de la llista corresponent. Per exemple

```
type = list(ordratio=c(1,2),symm=3)
```

indica que les columnes 1 i 2 són de classe `ordratio` i la tercera de classe `symm`. Per últim, el paràmetre `weights` permet ponderar la importància de cada variable en el càlcul de la distància.

Les mètriques disponibles són `"euclidean"`, `"manhattan"` i `"gower"`. Les dues primeres ja s'han detallat en l'apartat anterior, ja que també estan programades en la funció `dist`. Cal introduir la funció de distàncies per medi de la mètrica de Gower.

La distància de Gower permet utilitzar variables explicatives de diferents tipus, no sols numèriques, com en totes les altres mètriques vistes fins el moment. De fet, la funció `daisy`, quan les variables no són totes de classe `numeric` i no s'especifica res a l'argument d'entrada `type`, aplica forçosament la mètrica de Gower. El coeficient de similitud general, Gower (1971), és una de les mesures de proximitat més destacades per dades mixtes. Donats dos individus i i j es defineix per:

$$s_{ij} = \frac{\sum_{k=1}^{c}(1 - |z_{ik} - z_{jk}|/R_k) + m_{ij} + a_{ij}}{c + q + (b - d_{ij})},$$

on c és el número de variables contínues, z_{ik} i z_{jk} són els valors de (i, j) per a cada variable contínua k, R_k és el rang de valors de la variable k, m_{ij} és el número de coincidències dels individus i i j en les q variables categòriques i a_{ij} i d_{ij} són el número de coincidències positives i negatives respectivament per les b variables dicotòmiques.

El coeficient de Gower es pot transformar en una matriu de distàncies al quadrat de la següent manera:

$$d(i, j)^2 = 1 - s_{ij}.$$

Aquesta és una mètrica euclidiana (Gower and Legendre 1986).

Exemple 5.1

Perquè quedi clar el càlcul del coeficient de Gower es realitza un petit cas d'exemple, es planteja en Everitt (1993). Es tracta d'un conjunt de dades de pacients de psiquiatria:

Case	Weight	Anxiety	Depression	Hallucination	Age
Patient 1	120	1	0	0	1
Patient 2	150	2	1	0	2
Patient 3	110	3	1	1	3
Patient 4	145	1	0	1	3
Patient 5	120	1	0	1	1

La variable `Weight` es considera contínua, `Anxiety` i `Age` són nominals, `Depression` i `Halluciantion` binàries (0 o 1).

Pels pacients 1 i 2 el coeficient de Gower és calculat com

$$s_{12} = \frac{(1 - |120 - 150|/40) + 0 + 0}{1 + 2 + (2 - 1)} = 0.0625,$$

atès que de variables contínues només n'hi ha una (c=1), el rang de valors d'aquesta és $R = 150 - 110 = 40$, el número de coincidències de les dos variables nominals és 0, el número de coincidències positives de les dicotòmiques és 0 i el de negatives és 1. La taula de dissimilituds per a tots els parells de pacients és:

Case	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5
Patient 1	1				
Patient 2	0.0625	1			
Patient 3	0.1500	0.2000	1		
Patient 4	0.3438	0.1750	0.4250	1	
Patient 5	0.7500	0.0500	0.3500	0.5938	1

Funció `dist` del package `proxy`

La funció `dist` calcula les distàncies/similituds entre files o columnes d'una matriu $n \times p$, així com les distàncies creuades entre dos matrius o vectors diferents. Les instruccions d'R per a la seva utilització són les següents:

```
dist(x=Z, y = NULL, method = NULL, ..., diag = FALSE,
     upper = FALSE, pairwise = FALSE, by_rows = TRUE,
     convert_similarities = TRUE,
     auto_convert_data_frames = TRUE)

simil(x=Z, y = NULL, method = NULL, ..., diag = FALSE,
      upper = FALSE, pairwise = FALSE, by_rows = TRUE,
      convert_distances = TRUE,
      auto_convert_data_frames = TRUE)
```

L'atribut `x` (predictors Z), igualment que en les altres dues funcions, determina la matriu que es vol transformar. La `y` (per defecte val `NULL`) és una altra matriu, com `Z`, per calcular distàncies entre `Z` i `y`. El paràmetre `method` defineix la mètrica utilitzada. `diag` i `upper` ensenya la diagonal i la part superior de la matriu en cas que valguin `TRUE`. L'argument `pairwise` indica si es vol calcular les distàncies entre parelles `Z` i `y`, `by_rows` val `TRUE` si es calculen les distàncies per files i `FALSE` si es computen per columnes. `convert_similarities` indica si es volen convertir les distàncies en similituds (per defecte $s = 1/(1 + d)$), i `convert_distances` indica si es volen convertir similituds en distàncies (per defecte $d^2 = 1 - s$).

De nou cal incidir en els mètodes que proporciona la funció `dist` per calcular distàncies i similituds. La llista de mètriques disponibles és molt llarga. Se'n destaquen les següents:

```
* Similarity measures:
Braun-Blanquet, Chi-squared, correlation, cosine, Cramer,
Dice, eJaccard, Fager, Faith, fJaccard, Gower, Hamman,
Jaccard, Kulczynski1, Kulczynski2, Michael, Mountford,
Mozley, Ochiai, Pearson, Phi, Phi-squared, Russel,
simple matching, Simpson, Stiles, Tanimoto, Tschuprow,
Yule, Yule2

* Distance measures:
Bhjattacharyya, Bray, Canberra, Chord, divergence,
Euclidean, Geodesic, Hellinger, Kullback, Levenshtein,
Mahalanobis, Manhattan, Minkowski, Podani, Soergel,
supremum, Wave, Whittaker
```

Les mètriques vistes anteriorment, en les funcions `dist` (del paquet base) i `daisy` (del paquet `cluster`), són les més destacades. No s'entrarà en detall en el càlcul d'aquestes. Només recordar que les mesures de similitud per ser concebudes com a distàncies al quadrat s'han de retocar:

$$d(i, j)^2 = 1 - s_{ij}.$$

Alternativament, si la mesura ja és de distància, les distàncies al quadrat s'aconsegueixen simplement elevant la matriu al quadrat.

4.3 `dblm`: distance-based linear models

El `dblm` és la versió basada en distàncies del model lineal ordinal `lm` (s'explica en l'apartat 2.2). És un mètode de predicció d'una variable resposta

contínua, on la informació dels predictors està codificada en una matriu de distàncies entre individus. Aquesta pot ser introduïda directament com a inter-distàncies al quadrat o calculada a partir d'una matriu de variables explicatives.

Les variables explicatives poden contenir un conjunt de variables contínues, qualitatives o textuals. Pel cas més general el `dblm` és una extensió al `lm`.

S'ajusta un model lineal basat en distàncies mitjançant l'execució d'una de les següents quatre comandes:

```
## S3 method for class 'formula'
dblm(formula,data,...,metric="euclidean",method="OCV",
        full_search=FALSE,weights,rel.gvar=0.95,eff.rank)

## S3 method for class 'dist'
dblm(distance,y,...,method="OCV",full_search=FALSE,
        weights,rel.gvar=0.95,eff.rank)

## S3 method for class 'D2'
dblm(D2,y,...,method="OCV",full_search=FALSE,weights,
        rel.gvar=0.95,eff.rank)

## S3 method for class 'Gram'
dblm(G,y,...,method="OCV",full_search=FALSE,weights,
        rel.gvar=0.95,eff.rank)
```

En el cas més proper al model lineal usual, on es vol modelitzar una variable resposta a partir d'una matriu de variables explicatives, s'ajusta un model basat en distàncies amb la funció `dblm` de classe `formula`. En el cas que la informació dels predictors estigui concebuda en una matriu de distàncies de classe `dist`, la funció que s'utilitza és la `dblm` de classe `dist`. El mètode `dblm` de classe `D2` es crida quan la matriu de distàncies és al quadrat i és un objecte de classe `D2`. Per últim, pel cas que es vulgui ajustar un model lineal a partir de la matriu simètrica de productes interiors `G`, de classe `Gram`, s'utilitza la funció `dblm` de classe `Gram`.

Paràmetres d'entrada

El primer atribut `formula` és un objecte de classe `formula`, de la forma `y~Z`, on `y` conté la variable resposta i `Z` les variables explicatives. S'ha inclòs per compatibilitat amb la funció `lm`. L'atribut `data` és opcional i indica el *data frame* que conté les variables del model, tant explicatives com la resposta. El paràmetre `metric` defineix la mètrica a utilitzar per calcular la matriu

de distàncies a partir de `Z`, es proporcionen tres opcions, `"euclidean"` (per defecte), `"manhattan"` i `"gower"`. Les tres estan programades a la funció `daisy` del paquet `cluster`. En el cas que les variables explicatives no siguin numèriques s'usa immediatament la mètrica de Gower (es justifica a l'apartat 4.2). Si el primer argument de la funció `dblm` és de classe `formula`, s'utilitza directament la primera de les funcions especificades.

En el cas que la informació de Z estigui descrita en una matriu de distàncies, es separa la introducció del vector resposta i tal matriu de distàncies. L'atribut `y` conté la resposta (vector estrictament numèric) i `distance` l'objecte de classe `dist` amb les inter-distàncies o dissimilituds entre individus. Si el primer argument de la funció `dblm` és de classe `dist`, s'utilitza la segona de les funcions que s'han detallat. Recomanar les funcions `daisy`, i `dist` del package `stats` i `proxy`, estudiades en l'apartat anterior, per computar la matriu de distàncies.

Si les distàncies són al quadrat s'usa la tercera de les funcions (`dblm` de classe `D2`) i el primer atribut `D2` és qui les conté en format matriu i de classe `D2`. Fàcilment, és possible passar d'un objecte de la classe `dist` a un de classe `D2` mitjançant la funció de conversió `disttoD2` descrita en l'apartat 4.1. Alternativament, si la matriu ja conté les distàncies al quadrat, cal indicar amb anterioritat que aquesta és de classe `D2` amb la comanda d'R:

```
class(D2)<-"D2"
```

Si la informació de les distàncies la conté la matriu `G` amb els productes interiors ponderats obtinguts a partir de `D2`, s'usa la quarta de les funcions. `G` ha de ser un objecte de la classe `Gram` i es pot calcular a partir de la funció de conversió estudiada `D2toG`.

Els altres paràmetres són comuns per a les quatre vies d'ajust d'un model lineal basat en distàncies. L'argument `method` determina la manera que es fixarà el rang efectiu (`eff.rank`) per ajustar el model. El rang efectiu, o dimensió efectiva, correspon al número de dimensions escollit al realitzar multidimensional scaling d'una matriu de distàncies o dissimilituds entre individus:

$$G^{(k)} = U^{(k)}\Lambda^{(k)}U^{(k)'},$$

on la k és precisament la dimensió efectiva i la G és calculada com el doble centrat de la matriu de distàncies el quadrat:

$$G = -\frac{1}{2}J\Delta^2 J'.$$

De mètodes d'elecció automàtica del rang efectiu n'hi ha sis.

1. `"aic"`: El que minimitza el criteri d'informació d'Akaike:

$$AIC(\text{eff.rank}) = n * \log(SQR/n) + 2 * \text{eff.rank},$$

   on SQR fa referència a la suma de quadrats residual. L'AIC minimitza el compromís entre com de bons són els valors ajustats i el número de components utilitzat per estimar el model.

2. `"bic"`: El que minimitza el criteri d'informació Bayesiana:

$$BIC(\text{eff.rank}) = n * \log(SQR/n) + log(n) * \text{eff.rank}.$$

   Es tracta d'una reconsideració del criteri AIC penalitzant de forma més estricta la sobreparametrització. Per conjunts de dades grans la penalització és major que en el AIC on el coeficient és una constant (2).

3. `"OCV"`: El que minimitza el criteri de validació creuada ordinària. La validació creuada, o el mètode de leave one out, consisteix en treure consecutivament cadascuna de les observacions del conjunt de dades, estimar el model sense aquesta observació, fer la predicció en tal punt i comparar-la amb el valor real. Si això es fa per a cadascun dels possibles rangs efectius s'obté:

$$ECM_{CV}(\text{eff.rank}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i^{(-i)})^2.$$

   Es tria el rang efectiu que fa mínim el ECM_{CV}. No obstant, es tracta d'un procediment molt costós computacionalment ja que estima n models per a cada rang efectiu. S'ha estudiat una variant on no és necessari estimar les n regressions lineals per a cada dimensió efectiva. Es pot demostrar que, en el cas d'estimadors lineals,

$$ECM_{CV}(\text{eff.rank}) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right)^2.$$

   On h_{ii} és l'element i-èssim de la diagonal de la matriu barret H (definida en la pàgina 15). La part del numerador de la fracció indica la discrepància entre el valor real i l'estimat de la component i-èssima. En el denominador, l'element de la hat matrix h_{ii} caracteritza la importància que té l'observació i-èssima en el càlcul de la predicció per a

la mateixa observació i-èssima. Models sobreparametritzats comporten que la diagonal de la H s'acosti cada vegada més a 1. Per tant, encara que les estimacions, com és d'esperar, siguin més bones i el valor de la part del numerador es faci petit, la part del denominador disminueix consegüentment. Es contempla el cas extrem on no hi ha graus de llibertat per estimar la variança dels residus, on les $h_{ii} = 1$ i el criteri de validació creuada val 0/0. Això succeeix sempre quan el rang efectiu és igual al número d'observacions n.

4. `"GCV"`: El que minimitza la validació creuada generalitzada. És una modificació al criteri `OCV`. Aquesta consisteix en substituir la h_{ii} de la fórmula del `OCV` per la mitjana de la diagonal de H. Considerant ν la traça de H

$$\nu = \sum_{i=1}^{n} h_{ii},$$

l'error quadràtic mig corresponent al `GCV` per a cada rang efectiu esdevé:

$$ECM_{GCV}(\text{eff.rank}) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - \nu/n}\right)^2.$$

De nou reflecteix un compromís entre la bondat de l'ajust i la dimensió efectiva utilitzada. Per rangs efectius massa elevats la part del denominador tendeix a 0.

Utilitzar el mètode `GCV` pot evitar problemes computacionals respecte el `OCV`. El `OCV`, quan un sol element de h_{ii} val 1, la formula ja no té sentit, produint cassos 0/0 o $e_i/0$. En el `GCV` això no passa al prendre la mitjana de la diagonal de H.

5. `"eff.rank"`: S'especifica la dimensió efectiva desitjada en el paràmetre `eff.rank`.

6. `"rel.gvar"`: S'especifica la proporció de variabilitat geomètrica de les dades (definida a continuació) en què es vol ajustar el model mitjançant el paràmetre `rel.gvar` (valor entre 0 i 1). S'escull el rang efectiu mínim tal que la variabilitat geomètrica explicada és superior a l'explicitada en `rel.gvar`.

La variabilitat geomètrica és una de les mesures de dispersió en l'aplicació de multidimensional scaling. Al diagonalitzar la matriu G de productes escalars s'obtenen els valors i vectors singular de G. La suma

de tots els valors singulars (la traça) és una mesura de la variabilitat geomètrica total que conté el conjunt de dades. Per tant, al triar un cert rang efectiu, s'indica de forma percentual la informació del conjunt de dades que es manté en la nova dimensió. És el concepte anàleg a la variabilitat explicada al realitzar un ACP.

És crucial triar de manera encertada el rang efectiu. Per a `eff.rank` massa petits, el model no serà suficient per explicar la resposta i el coeficient de determinació (R^2) serà baix. Contràriament, per a `eff.rank` excessivament alts, s'està sobreajustant, per la qual cosa l'error de predicció per a futures observacions incrementarà tot i que l'R^2 hagi augmentat. L'R^2 és calculat en tots el mètodes *dbstats* de la següent manera:

$$R^2 = 1 - \sum_{i=1}^{n} \frac{w_i(\hat{y}_i - y_i)^2}{w_i(y_i - \bar{y})^2}.$$

Finalment, el paràmetre d'entrada `full_search` determina el procés d'optimització que s'utilitzarà per modelitzar el criteri de selecció del rang efectiu descrit per l'atribut `method`. Només té validesa en el cas que el mètode seleccionat sigui l'`"AIC"`, el `"BIC"`, l'`"OCV"` o el `"GCV"`. Es tracta d'un booleà que fixat a `TRUE` optimitza els criteris calculant el valor per a cada un dels possibles rangs efectiu (de 1 fins a $n-1$). En el cas que aquest sigui `FALSE`, l'optimització es realitza usant la funció `optimize` d'R. El mètode que usa `optimize` per optimitzar una funció és una combinació de recerca de la secció àuria i la interpolació parabòlica successiva. El temps computacional utilitzant aquesta via disminueix, si bé el resultat òptim pot esdevenir un mínim local. L'argument `full_search`, si s'utilitza la mètrica euclidiana pel càlcul de les distàncies, només és aplicable quan entra més d'una variable en el model. Alternativament, l'optimització es para. El motiu és que amb una sola variable, aplicant la mètrica euclidiana les files de G són linealment dependents i una de sola ja conté tota la informació de les dades. El programa retorna un missatge d'error informant que el valor mínim pel qual minimitza la funció és el mateix que el valor màxim (els dos valen 1).

Valors resultants

Quan s'ajusta un model lineal basat en distàncies `dblm` s'obté un objecte de classe `dblm` que conté les següents components:

`residuals`	La resposta menys els valors ajustats (`fitted values`).
`fitted.values`	Els valors ajustats per a cada individu del model.
`df.residuals`	Els graus de llibertat dels residus: nombre de valors en el càlcul d'un estadístic lliures de variar.
`weights`	Els pesos especificats en el model.
`y`	La resposta usada per ajustar el model.
`H`	La matriu de projecció Hat matrix.
`call`	Guarda la crida efectuada de la funció
`rel.gvar`	Variabilitat geomètrica relativa usada per ajustar el model.
`eff.rank`	Dimensió escollida per a l'estimació del model.
`ocv`	`OCV` estimat.
`gcv`	`GCV` estimat.
`aic`	Criteri d'informació d'Akaike.
`bic`	Criteri d'informació de Bayes.

A més, un objecte de la classe `dblm` té disponibles els mètodes genèrics `print`, `summary`, `plot` i `prediction`. A continuació es veuran detalladament.

Exemple 5.2

Per tal d'il·lustrar com utilitzar els atributs d'entrada i les components resultants es reprodueix un exemple amb un conjunt de dades simulades:

```
> n <- 50                          # nombre d'observacions
> p <- 2                           # nombre de variables
> k <- 5                           # determina la magnitud dels coeficients
> set.seed(12)                     # llavor per obtenir la mateixa seqüència
> z1 <- matrix(rnorm(n),nrow=n)    # variable contínua z1
> z2 <- rbinom(50, 1, 0.5)         # variable binària z2
> Z <-as.matrix(data.frame(z1=z1,z2=z2))
> b <- matrix(runif(p)*k,nrow=p) # coeficients de la regressió
[1,] 4.776754 4.980259
> e <- rnorm(n)                    # errors obtinguts
> y <- Z%*%b + e                   # variable resposta resultant
```

Es pretén fer prediccions per a la variable resposta y a partir dels valors de p predictors, en aquest cas dos (`z1` numèric i `z2` binari). Quant les dades es composen de variables explicatives Z, la funció que s'utilitza per ajustar un model lineal basat en distàncies és la `dblm` de classe `formula`:

```
> dblm1 <- dblm(y~Z,metric="euclidean",full_search=TRUE)
```

`dblm1` és un objecte de classe `dblm`, que conté tota la llista d'atributs expressats anteriorment en la secció de valors resultants. Per exemple, si es vol conèixer el rang efectiu utilitzat, es pot obtenir mitjançant la instrucció

```
> dblm1$eff.rank
```

En aquest cas el rang efectiu val 2. De fet, en el capítol de metodologia (apartat 2.2) s'ha demostrat que ajustar un model `dblm` amb mètrica euclidiana és equivalent a estimar la resposta pel model lineal ordinari. Es comprova restant els valors previstos del `dblm1` amb els valors previstos del `lm` adjacent. Cal adonar-se que la diferència màxima és pràcticament zero.

```
> lm1 <- lm(y~Z)           # model de regressió lineal.
> max(lm1$fitted.values - dblm1$fitted.values)
[1] 8.881784e-15
```

En aquest cas, s'ha utilitzat el mètode per defecte per obtenir la dimensió efectiva òptima (`OCV`) i la mètrica euclidiana pel càlcul de distàncies.

No obstant, el model basat en distàncies és molt flexible, canviant la funció de distàncies es poden obtenir resultats diferents, que en alguns casos poden ser satisfactoris. Per exemple, s'ajusta el model amb una de les mètriques més populars per dades mixtes: la mètrica de Gower. A més, per determinar la dimensió efectiva del model s'utilitza el mètode `rel.gvar` amb una variabilitat geomètrica relativa mínima del 90%.

```
> dblm2 <- dblm(y~Z,metric="gower",method="rel.gvar",rel.gvar=0.9)
```

Ara les estimacions ja no són les mateixes (s'observa en la Figura 4.1). Mentre que amb la mètrica de Gower el rang efectiu que conté el 90% d'informació de les distàncies és de 3, en la mètrica euclidiana amb dues dimensions ja conté el 100% del `rel.gvar`.

A continuació es presenten altres vies per modelar un `dblm` on la informació dels predictors és concebuda com a distàncies entre individus. Utilitzant la funció `dist` del paquet base es pot transformar la `Z` en una matriu de classe `dist` (per exemple de mètrica Minkowsky de grau 3):

```
> D <- dist(Z,method="minkowski",p=3)
```

Per ajustar el model, el primer atribut de la funció `dblm` ha de ser `D` (de classe `dist`). Les quatre funcions `dblm` es criden per la mateixa comanda (`dblm`) i és el primer argument el que caracteritza quina de les vies usar. En aquest cas, al ser de classe `dist` s'executa la segona de les funcions `dblm` presentades. El mètode d'elecció del rang efectiu serà l'`"AIC"`:

```
> dblm3 <- dblm(D,y,method="AIC")
```

L'ajust del `dblm3` també difereix respecte als altres dos models (es pot veure en la Figura 4.1).

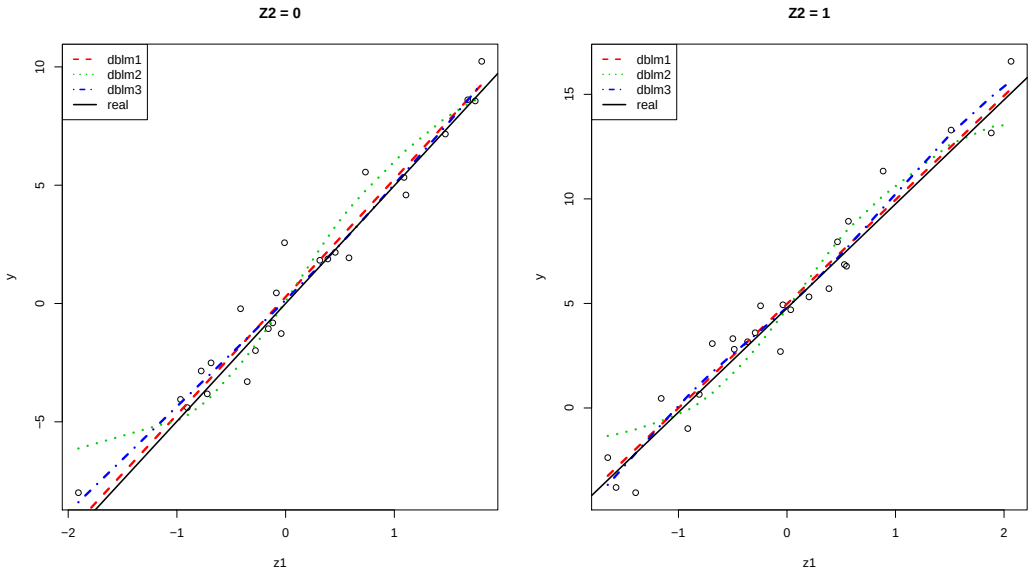

Figura 4.1: Regressió basada en distàncies (Gower: `dblm2`, Euclidean: `dblm1` i Manhattan p=3: `dblm3`).

Finalment, també es pot ajustar el model quan els predictors estan codificats com a objectes de classe `D2` i de classe `Gram`. És fàcil obtenir una matriu de distàncies al quadrat i una matriu G de productes interiors utilitzant les conversions `disttoD2(D)` i `D2toG(D2)`. Els models ja es podrien ajustar (els dos models són equivalents):

```
> D2 <- disttoD2(D)
> G <- D2toG(D2)
```

```
> dblm5 <- dblm(D2,y,method="AIC",full_search=T)
> dblm6 <- dblm(G,y,method="AIC",full_search=T)
> max(dblm6$fitt-dblm5$fitt)
[1] 0
```

Mètode `print.dblm`

El `print` d'un objecte de classe `dblm` defineix la sortida per pantalla d'un objecte `dblm`.

Continuació Exemple 5.2

En el cas d'exemple, la sortida obtinguda del model ajustat `dblm1` és:

```
> print(dblm1)
call:   dblm.formula(formula = y ~ Z, metric = "euclidean", full_search = T)

method: OCV ,   search: full
metric: euclidean
Optimal effective rank = 2
Relative geometric variability = 1.000000
Ordinary cross-validation estimate of the prediction error : 1.212931e+00
```

Mètode `summary.dblm`

El `summary` d'un objecte de classe `dblm` calcula i retorna una sèrie d'atributs que poden ser d'interès per resumir el model. Els processa en una llista de classe `summary.dblm`, que conté els següents elements:

`residuals`	Resposta menys valors previstos.
`sigma`	Error residual estandarditzat.
`r.squared`	Coeficient de determinació R^2.
`adj.r.squared`	R^2 ajustat.
`rdf`	Graus de llibertat residuals.
`call`	Crida de la funció.
`gvar`	Variabilitat geomètrica de la matriu de distàncies al quadrat ponderada pels pesos definits en l'atribut d'entrada `weights`.
`gvec`	Vector numèric corresponent a la diagonal de la matriu de productes interiors ponderats G.
`method`	Mètode utilitzat per determinar el rang efectiu.
`eff.rank`	Rang efectiu en l'ajust del model.
`rel.gvar`	Variabilitat geomètrica relativa en l'ajust del model.

`crit.value` Valor del criteri en `method` que optimitza el rang efectiu.

Continuació Exemple 5.2

En el cas d'exemple, pel que fa referència al model `dblm2`, el `summary` obtingut es podria resumir amb el mètode `print` d'un objecte `summary` (`print.summary`).

```
> summary(dblm2)
call:   dblm.formula(formula = y ~ Z, metric = "gower", method = "rel.gvar",
    rel.gvar = 0.9)

Weighted Residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-3.03e+00 -1.06e+00 -2.05e-02 -1.04e-15  9.14e-01  3.01e+00

R-squared: 0.930146      Adjusted R-squared: 0.925590
Weighted Geometric Variability: 0.192985

Used effective rank = 3
Relative geometric variability = 0.903003
```

S'ha procurat que la sortida obtinguda sigui pràcticament la mateixa que dóna R quan es fa el `summary` d'un objecte `lm`.

Mètode `plot.dblm`

El mètode genèric `plot` d'un objecte de classe `dblm` té disponibles sis gràfics diferents: residus envers els valors previstos, Q-Qplot per a normalitat, el valor absolut dels residus envers els valors previstos (*scale location plot*), distàncies de Cook, residus envers *leverages* i rang efectiu òptim. Per defecte només són proporcionats els tres primers i el cinquè. Com succeeix amb el plot dels objectes de classe `lm` i `glm`, el mètode `plot` dels objectes `dblm` i `dbglm` (que es presenta més endavant) és el mateix. La crida de la funció `plot` es realitza amb la següent instrucció:

```
plot(x,which=c(1:3, 5),id.n=3,main="",
     cook.levels = c(0.5, 1),cex.id = 0.75,
     type_glm=c("link","response"),...)
```

`x` conté l'objecte de classe `dblm`, el paràmetre `which` és un vector on s'indica els gràfics que es vol que surtin per pantalla (per defecte de l'1 al 3 i el 5). l'argument `id.n` determina el número d'observacions extremes que es volen marcar en els gràfics. En el `main` s'hi especifica de forma opcional el títol del gràfic. `cook.levels` indica els nivells de la distància de Cook pels quals es dibuixa el contorn de confiança, `cex.id` permet magnificar els nivells dels

punts i `type_glm` és un paràmetre pel mètode `plot.dbglm` (no té cap efecte en el cas d'un `dblm`).

Els cinc primers gràfics són molt útils en l'anàlisi dels residus per validar un model i són exactament els mateixos que proporciona el `plot` de la classe `lm`. En el gràfic dels residus contra valors previstos es pot mirar si la variança és constant, en el qq-plot s'hi verifica la normalitat dels residus, el plot *Scale-Location* envers valors previstos pren l'arrel quadrada dels residus en valor absolut per disminuir l'asimetria. El gràfic de les distàncies de Cook mesura la influència de cada observació en el model. Punts amb distància de Cook elevada indiquen la necessitat de ser estudiats amb més detall en l'anàlisi. Per últim *Residual-leverage* ensenya els valors més extrems en termes de distàncies.

Continuació Exemple 5.2

En el cas d'exemple, els gràfics referents al model `dblm5` estan detallats a la Figura 4.2. Per tal que surtin tots al mateix gràfic s'usen les comandes:

```
> par(mfrow=c(3,2))
> plot(dblm5,which=1:6)
```

Mètode `predict.dblm`

El mètode `predict.dblm` retorna els valors previstos obtinguts de l'avaluació del model de regressió basat en distàncies per a un nou conjunt de dades (`newdata`). `newdata` pot ser expressat a partir dels valors de les variables explicatives en les noves observacions, la matriu de distància el quadrat entre el nous casos i els n inicials, o les files de la matriu de productes interior G dels nous individus respecte dels individus que ajusten el model.

```
predict(object,newdata,type="Z",...)
```

`object` és una instància de la classe `dblm`, `newdata` és un *data frame* o matriu que conté les variables explicatives, les distàncies al quadrat o els productes interiors G. Mitjançant l'atribut `type` s'indica la manera que s'ha explicitat les noves dades. `type="Z"` indica que `newdata` conté les variables explicatives, `type="D2"` indica que conté les distàncies al quadrat i `type="G"` els productes interiors. El format de les noves dades ha de ser coherent en com s'ha ajustat el `dblm` anteriorment. Si el primer atribut del `dblm` era de classe `formula`, les noves dades forçosament han de ser de tipus `"Z"`. Si eren

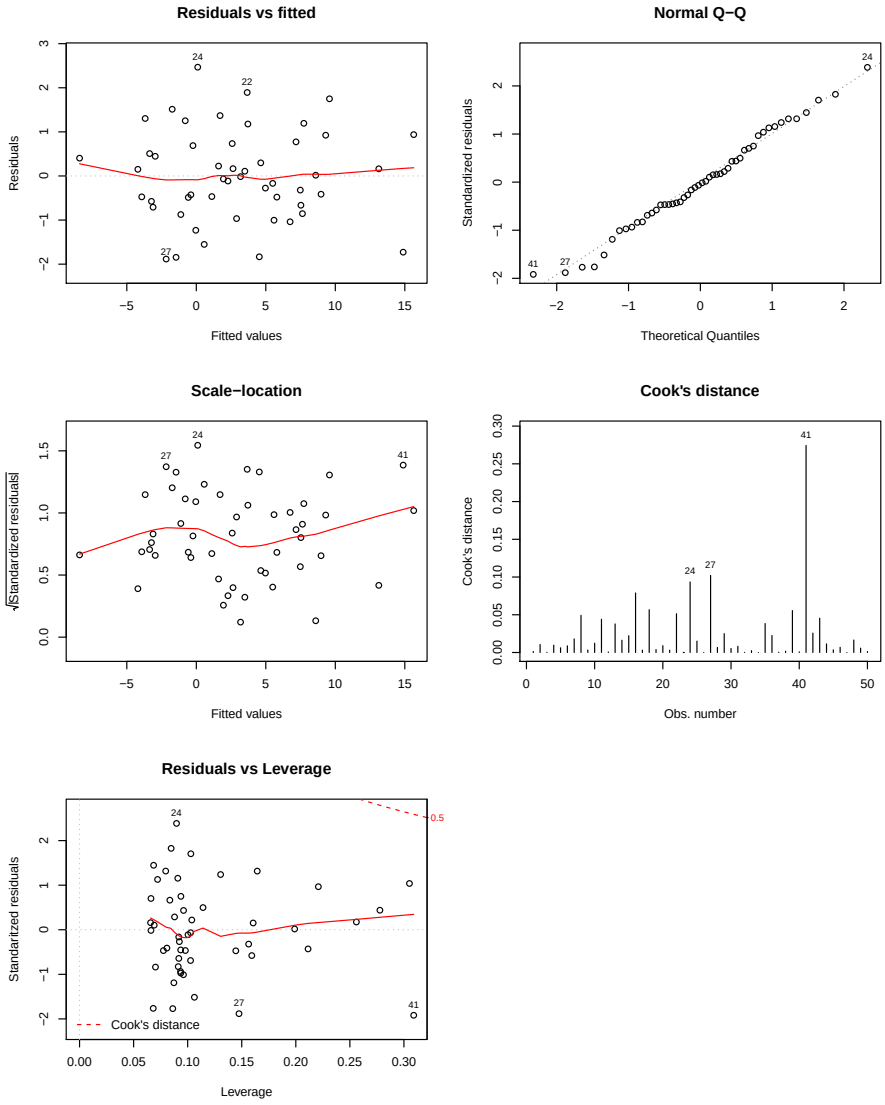

Figura 4.2: Bateria de plots `dblm` pel model `dblm5`.

distàncies de classe `dist` o distàncies al quadrat de classe `D2`, les noves dades han de ser distàncies al quadrat. Si l'ajust s'ha fet per la matriu de la classe `Gram`, les noves dades han de ser de tipus `Gram`. En cas contrari el programa es pot parar o pot donar una sortida errònia.

Continuació Exemple 5.2

Continuant l'exemple, es pretenen fer prediccions en els models `dblm2`, el `dblm5` i el `dblm6`. En el primer cas, s'ha ajustat el model directament a partir dels predictors Z. Per tant el `type` del mètode `predict` ha de ser `"Z"` i `newdata` ha de contenir els valors de les variables explicatives en el nou punt. Els valors en z del nou individu $n+1$ són $z_{n+1} = [0, 0]$. També es vol calcular la predicció del individu $n+2$ tal que $z_{n+2} = [2, 1]$. Les prediccions es realitzen amb les comandes

```
> newdata <- matrix(c(0,2,0,1),ncol=2)
> predict(dblm2,newdata=newdata,type="Z")
```

I els resultats obtinguts són:

```
       Predicció
Z=0    0.1001278
Z=2   13.5256099
```

Si les dades estan en format `D2` com és el cas del model `dblm5`, per a la mateixa $z_{n+1} = [0, 0]$ i $z_{n+2} = [2, 1]$ s'han de calcular les distàncies respecte els 50 individus del model:

```
> Daux<-dist(rbind(Z,newdata),method="minkowski",p=3)
> D2aux<-disttoD2(Daux)
> D2new<-D2aux[(n+1):(n+2),1:50]
```

`D2new` conté les distàncies al quadrat de les noves observacions. Les prediccions són les següents:

```
> predict(dblm5,newdata=D2new,type="D2")
       Predicció
Z=0    0.1478065
Z=2   15.3767033
```

En el `dblm6`, finalment, es té la informació de les dades a partir d'un objecte de la classe `Gram`. Per tant, les prediccions es realitzen a partir de les components de G dels nous individus:

```
> predict(dblm6,newdata=Gnew,type="G")
       Predicció
Z=0    0.1478065
Z=2   15.3767033
```

Les prediccions que dóna el `dblm2` són diferents a les del `dblm5` i `dblm6`. L'ajust `dblm2` ha donat estimacions més baixes en la `y` que els altres dos models.

4.4 ldblm: local distance-based linear models

El model de regressió local basat en distàncies, `ldblm`, és la versió local del model lineal basat en distàncies `dblm` i és el concepte anàleg al local linear model on la informació dels predictors està expressada com a distàncies entre individus (s'estudia en l'apartat 2.3). Es pretén modelitzar una variable resposta contínua a partir d'una sèrie de variables explicatives. En el programari R, la funció que executa aquesta tasca és diu `loess`, o el seu antecessor `lowess`. Alternativament, altres paquets d'interès han elaborat la versió no paramètrica del model lineal. En destaca la funció `sm.regression` del paquet `sm`.

Utilitzant la funció `ldblm`, de la mateixa manera que en el cas del `dblm`, hi ha quatre vies per ajustar un model lineal local basat en distàncies:

```
## S3 method for class 'formula'
ldblm(formula,data,...,kind.of.kernel=1,
        metric1="euclidean",metric2=metric1,method="GCV",
        weights,user_h=NULL,h.range=NULL,noh=10,k.knn=3,
        rel.gvar=0.95,eff.rank=NULL)

## S3 method for class 'dist'
ldblm(dist1,dist2=dist1,y,kind.of.kernel=1,
        method="GCV",weights,user_h=quantile(dist1,.25),
        h.range=quantile(as.matrix(dist1),c(.05,.5)),noh=10,
        k.knn=3,rel.gvar=0.95,eff.rank=NULL,...)

## S3 method for class 'D2'
ldblm(D2_1,D2_2=D2_1,y,kind.of.kernel=1,method="GCV",
        weights,user_h=quantile(D2_1,.25)^.5,
        h.range=quantile(as.matrix(D2_1),c(.05,.5))^.5,noh=10,
        k.knn=3,rel.gvar=0.95,eff.rank=NULL,...)
```

```
## S3 method for class 'Gram'
ldblm(G1,G2=G1,y,kind.of.kernel=1,method="GCV",
        weights,user_h=NULL,h.range=NULL,noh=10,k.knn=3,
        rel.gvar=0.95,eff.rank=NULL,...)
```

El primer argument d'entrada que s'introdueix en la funció `ldblm` és el que determina en quina de les quatre alternatives s'ajusta el model local. Quan els predictor Z són expressats com una matriu $n \times p$ amb els valors de les p variables explicatives s'utilitza la primera funció especificada, expressant la relació entre la y i la Z en format `formula`. Si la informació dels predictors està continguda en una matriu de inter-distàncies entre individus, llavors s'utilitza el mètode S3 `ldblm` de classe `dist` o `D2` (si les distàncies són al quadrat i de classe `D2`). Finalment, si les dades estan emmagatzemades en una matriu G, la funció que modelitza la y amb la G és la `ldblm` de classe `Gram`.

Paràmetres d'entrada

En el local distance-based linear model s'han de definir dos distàncies diferents. La primera és utilitzada per assignar els pesos mitjançant una funció Kernel i l'altra pel càlcul de cada `dblm`.

Si s'ajusta el model per la primera via, és a dir, la versió més propera al model lineal local usual on les dades estan configurades com una variable resposta numèrica i una matriu de p predictors, s'han de definir dos mètriques o semimètriques per calcular les distàncies entre els punts. El paràmetre `metric1`, que per defecte usa la mètrica euclidiana, i el `metric2`, que per defecte és igual al `metric1`. Les mètriques disponibles són les mateixes que en el `dblm`: `"euclidean"`, `"manhattan"` i `"gower"`. Aquests dos paràmetres només cal definir-los a l'utilitzar la funció `ldblm` en format `formula`.

Alternativament, usant la funció `ldblm` de classe `dist` o de classe `D2`, s'ha d'especificar, respectivament, les dos distàncies mitjançant els paràmetres `dist1`,`dist2` o `D2_1`,`D2_2`. Recordar que no necessàriament les dues matrius han de ser iguals. Pel mètode de classe `Gram` s'han de passar, equivalentment, els paràmetres `G1` i `G2`.

Els altres atributs ja són comuns en les quatre vies d'estimació del model. El `kind.of.kernel` és un número enter que determina la funció de suavitzat Kernel utilitzada per determinar els pesos en cada model lineal local.

Hi ha sis opcions diferents: (1) Epanechnikov (Default), (2) Biweight, (3) Triweight, (4) Normal, (5) Triangular, (6) Uniform.

El paràmetre `method`, indica el mètode pel qual l'ample de banda h és optimitzat. Hi ha cinc maneres diferents per determinar la h òptima de forma automàtica: `"AIC"`, `"BIC"`, `"OCV"`, `"GCV"` (per defecte) i `"user_h"`. L'AIC i el BIC prenen la h^* minimitzant el criteri d'informació d'Akaike i Bayesiana respectivament. La h^* per `OCV` i per `GCV` minimitzen la quantia referent a la validació creuada. El darrer, `user_h`, indica que l'usuari pot definir el paràmetre de suavitzat mitjançant l'atribut `user_h` (per defecte val el primer quartil de les distàncies definides en `dist1`).

`weights` és un paràmetre d'entrada opcional que determina la ponderació de cada observació *a priori* en el model. Els pesos definits en `weights` es multipliquen amb els pesos Kernel per determinar els pesos finals en cada ajust local.

L'atribut `h.range` és un vector de mida dos amb el rang de valors que pren la `h` (per defecte: els quantils 0.05 i 0.5 de les distàncies $d(i, j)$ en `dist1`). En `noh` es fixen el número d'amples de banda `h` que són avaluats a dins del rang definit per `h.range`. Els dos s'utilitzen quan es selecciona automàticament el paràmetre de suavitzat (`method` diferent a `user_h`). L'atribut `k.knn` determina el número mínim de veïns amb pes major a zero per ajustar cada model lineal. Està programat per evitar errors de compilació, deguts a amples de banda massa petits que fan que únicament quedi una sola observació amb pes positiu per ajustar el model. Per defecte `k.knn` és tres (tres observacions com a mínim en el veïnatge del punt a estimar).

Els dos últims atributs d'entrada fan referència a com elegir el rang efectiu en cada iteració del `dblm`. El `rel.gvar` indica el percentatge de variabilitat geomètrica que es vol com a mínim en cada model `dblm` (per defecte `rel.gvar = 0.95`). Pel que fa al `eff.rank` indica el número de coordenades euclidianes (dimensió efectiva) que s'usen en cada `dblm`. Per defecte val `NULL`, i quan es declara, es sobreposa a l'especificació del `rel.gvar`.

Valors resultants

A l'ajustar un model local lineal basat en distàncies `ldblm` s'obté un objecte de classe `ldblm` que conté els següents atributs:

`residuals`	La resposta menys els valors ajustats (fitted values).
`fitted.values`	Els valors ajustats per a cada individu del model.
`h_opt`	Ample de banda òptim en el procés d'estimació del model (si `method!=user_h`).
`S`	La matriu de suavitzat projectora Smoothing Hat matrix: `fitted.values = S*Y`
`weights`	Els pesos especificats inicialment.
`y`	La resposta usada per ajustar el model.
`call`	Guarda la crida efectuada de la funció.
`dist1`	La matriu de distàncies usada per calcular els pesos de cada observació. És un objecte de classe `D2` o `dist`.
`dist2`	Matriu de distàncies usada per calcular cada `dblm`. És un objecte de classe `D2` o `dist`.
`ocv`	Valor del OCV pel model estimat.
`gcv`	Valor del GCV pel model estimat.
`aic`	Valor del criteri d'Akaike.
`bic`	Valor del criteri Bayesian.

Igual que en el `dblm`, un objecte de la classe `ldblm` permet usar els mètodes genèrics `print`, `summary`, `plot` i `prediction`.

Exemple 5.3

Es realitza un exemple senzill, amb dades simulades, per tal de veure de manera detallada com utilitzar la funció `ldblm`.

```
> n <- 200                          # nombre d'observacions.
> p <- 2                            # nombre de variables.

> set.seed(2)                       # llavor per Z
> z1 <- matrix(rnorm(n),nrow=n)     # variable explicativa z1
> z2 <- as.factor(rbinom(n, 2, 0.5)) # variable explicativa z2
> z21 <- matrix(ifelse(z2==1,1,0))   # variable dummy 1
> z22 <- matrix(ifelse(z2==2,1,0))   # variable dummy 2
> Z <-as.matrix(data.frame(z1=z1,z21=z21,z22=z22))

# coeficient beta i error
> b1 <-  matrix(c(runif(1)*2,runif(2)*5),nrow=3) .
> b2 <- matrix(runif(p)*k,nrow=p)
> b3 <- matrix(runif(p)*k,nrow=p)

> e <- rnorm(n)                     # error obtingut

# variable resposta resultant
> y <- Z%*%b1 + z1^2%*%b2 +z1^3%*%b3 + e
```

Es pretén ajustar un model de regressió no paramètrica on la informació dels predictors esta continguda en `z1` i `z2`. La variable `z2` és un factor amb 3 nivells (0, 1 i 2). Per tal d'aplicar la distància euclidiana es codifica `z2` en dues variables auxiliars dummies `z21` i `z22`. El model `ldblm1` s'ajusta amb mètrica `euclidean`:

```
> ldblm1<-ldblm(y~Z,kind.of.kernel=1,method="GCV",
                noh=3,k.knn=3)
```

La ponderació de les observacions en cada model lineal local és generada per una funció Kernel Epanechnikov i la h òptima es busca automàticament en 3 punts (`noh=3`) diferents entre la distància del quantil 0.05 i el quantil 0.5 (`h.range` per defecte). Es tria la que minimitza el criteri de validació creuada `GCV`. Com a mínim la h en cada iteració ha de ser tal que almenys 3 individus tinguin pesos positius (`k.knn=3`). L'ample de banda òptim obtingut és de 0.253564 (`ldblm1$h_opt`).

Un ajust alternatiu seria un model on ja no hi ha predictors. La informació que aquests contenen està expressada com una matriu de distàncies al quadrat amb mètrica euclidiana:

```
> D2<-as.matrix(dist(Z)^2)
> class(D2)<-"D2"
```

El `ldblm2` es crida amb la següent instrucció:

```
> ldblm2 <- ldblm(D2_1=D2,D2_2=D2,y,kind.of.kernel=3,
                  method="user_h",k.knn=5)
```

Les distàncies per ajustar els `dblm` locals i els pesos kernel són les mateixes. La funció Kernel és la `Triweight`. El mètode d'elecció automàtica és `user_h`, per tant el bandwidth a utilitzar s'especifica en el paràmetre `user_h`. Com que no s'ha indicat res, es pren el valor per defecte (el primer quartil de les distàncies entre individus en `sqrt(D2_1)`). Per últim, com a mínim cada model s'ha de composar de 5 observacions en el veïnatge del punt a estimar (`k.knn=5`).

En aquest cas la h utilitzada és de 1.055469, major al paràmetre de suavitzat del model `ldblm1`. `ldblm2`, per tant, serà una funció més suau que

`ldblm1` (que tindrà més variabilitat). S'observa en la figura 4.3.

Amb l'últim model que s'ajusta, el `ldblm3`, s'il·lustra el fet que les dues distàncies no han de ser estrictament les mateixes. La primera té mètrica Manhattan (distància de valor absolut) i la segona té mètrica de Gower (calculada directament pel factor `z2` i no per les dummies `z21` i `z22` al ser una mètrica per dades mixtes):

```
> D_1 <- dist(Z,"manhattan")
> D_2 <- daisy(data.frame(z1=z1,z2=z2),"gower")
> ldblm3 <- ldblm(dist1=D_1,dist2=D_2,y,kind.of.kernel=2,method="BIC",
                        noh=7,k.knn=5)
```

Es busca l'h òptima en set punts diferent entre el rang mínim i el rang màxim (que per defecte estan assignats als quantils 0.05 i 0.5 de D_1). El mètode utilitzat és el `BIC` i com a mínim els models locals han de contenir 5 observacions. L'ample de banda obtingut és de 2.087728. Aquest no és comparable amb les h dels models anteriors, ja que les funcions de distàncies són diferents. De fet, tot i que en aquest cas la h és major a la dels altres dos ajustos, defineix una funció molt semblant a la obtinguda pel model `ldblm1`. Es reflecteix en la Figura 4.3.

Finalment, destaquem el fet que, mentre els ajustos locals capten la relació cúbica entre la $z1$ i la y, si es realitza un model lineal usual aquesta no es recull (s'observa en la Figura 4.3).

Mètode `print.ldblm`

El mètode `print` d'un objecte de la classe `ldblm` defineix la sortida per pantalla d'un objecte `ldblm`.

Continuació Exemple 5.3

En el cas d'exemple, la sortida obtinguda del model ajustat `ldblm1` és:

```
> print(ldblm1)
call:   ldblm.formula(formula = y ~ Z, kind.of.kernel = 1, method = "GCV",
    noh = 3, k.knn = 3)

method= GCV,     kind of kernel= (1) Epanechnikov
metric1: euclidean
metric2: euclidean
optimal bandwidth h : 0.253564
Generalized cross-validation estimate of the prediction error : 2.414817e-02
```

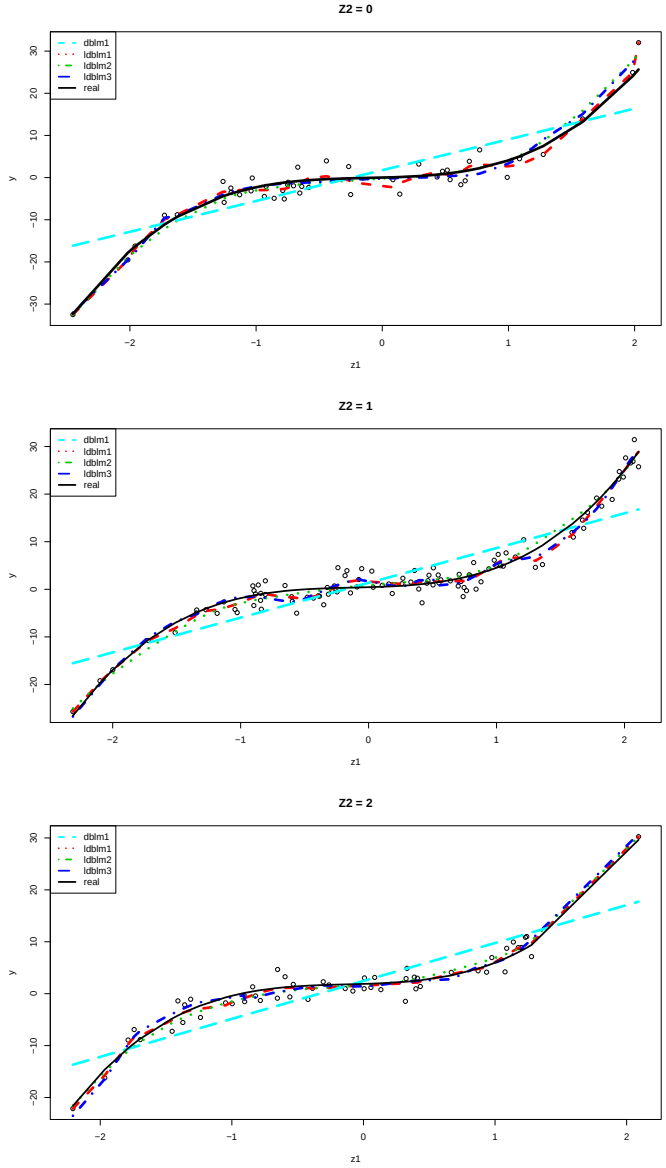

Figura 4.3: Comparació entre els ajustos locals `ldblm1`, `ldblm2`, `ldblm3` i el model lineal `dblm1`: La relació cúbica entre `y` i `z1` no la recull el model lineal.

Mètode `summary.ldblm`

El `summary` d'un objecte de classe `ldblm` retorna una llista d'atributs, de classe `summary.ldblm`, que resumeixen el model. Conté els següents elements:

`nobs`	Número d'observacions.
`r.squared`	Coeficient de determinació R^2: determina la bondat de l'ajust.
`trace.hat`	La traça de la matriu de suavitzat `Shat`.
`call`	Crida de la funció.
`residuals`	La resta de resposta i els valors previstos.
`family`	La família de distribucions que pertany el model: en aquest cas `gaussian`.
`kind.kernel`	Funció Kernel utilitzada pel càlcul dels pesos.
`method`	Criteri seleccionat per optimitzar l'ample de banda.
`h_opt`	Ample de banda òptim i escollit per ajustar el model.
`crit.value`	Valor del criteri d'optimització del bandwidth.

Continuació Exemple 5.3

En el cas d'exemple, pel que fa referència al model `ldblm2`, el `print` del objecte `summary` obtingut és el següent:

```
> summary(ldblm2)
call: ldblm.D2(y = y, D2_1 = D2, D2_2 = D2,
        kind.of.kernel = 3, method = "user_h", k.knn = 5)

Residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-5.3300 -1.1950 -0.1900 -0.0719  1.2400  5.0300

Number of Observations: 200
R-squared : 0.9479
Trace of smoother matrix: 18.74
family: gaussian

kind of kernel= (3) Triweight
user bandwidth h : 1.055469
```

Mètode `plot.ldblm`

El mètode genèric plot d'un objecte de la classe `ldblm` té disponibles 3 gràfics diferents: resposta envers els valors previstos, residus envers els valors previstos i els valors del criteri pels diferents amples de banda avaluats (només

es possible quan el mètode d'elecció de la h és `OCV`, `GCV`, `AIC` o `BIC`). La crida de la funció `plot` d'un objecte `ldblm` es realitza amb la següent comanda:

```
plot(x,which=c(1,2),id.n=3,main="",...)
```

on `x` és un objecte de classe `ldblm`, `which` indica quins dels gràfics es volen mostrar per pantalla (per defecte els dos primers) i `id.n` indica el nombre d'observacions extremes que es volen senyalar en cada plot. Finalment en el `main` es pot explicitar el títol del gràfic.

Continuació Exemple 5.3

Seguint l'exemple, els 3 gràfics obtinguts en l'ajust del model lineal local `ldblm3` són els que s'aprecien en la Figura 4.4:

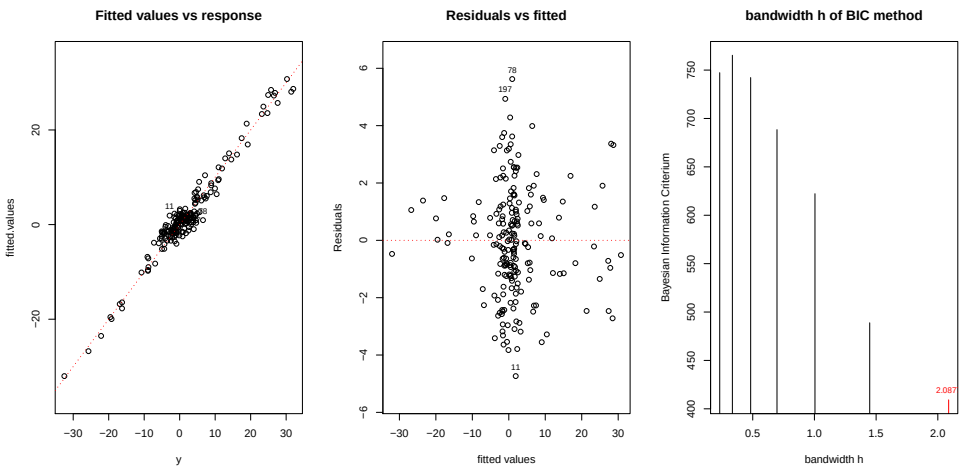

Figura 4.4: Bateria de plots d'un objecte `ldblm` (`ldblm3`)

El paràmetre de suavitzat òptim, en el model `ldblm3` és troba al màxim del rang de valors de h que s'ha avaluat el criteri. Indica que s'està sobre-ajustant el model. S'hauria de tornar a ajustar, augmentant els valors del `range.h`, i elegint de nou la millor h.

Mètode `predict.ldblm`

La forma de fer prediccions per a un objecte de classe `ldblm` és lleugerament diferent al cas `dblm`.

```
predict(object,newdata1,newdata2=newdata1,
        new.k.knn=3,type="Z",...)
```

El model inicial ha estat ajustat en base a dues matrius de distàncies entre individus diferents. Per tant, en la predicció s'ha d'especificar consegüentment les noves distàncies amb les dues mètriques corresponents. `dbstats`, permet expressar les noves dades per diferents vies. Si s'ha ajustat el model configurant la informació de les dades com a variables predictores, `newdata1` i `newdata2` han de ser els valors en Z dels nous individus i s'ha d'especificar el tipus `type="Z"`. Contràriament, quan la codificació és mitjançant distàncies o distàncies al quadrat, les noves dades han de contenir les distàncies al quadrat entre els nous individus i els inicials, a més el `type="D2"`. Finalment es poden expressar com els productes interiors, quan el model s'hagi estimat mitjançant un objecte de classe `Gram` (`type="G"`). S'ha de tenir molta cura a l'hora d'expressar la informació dels predictors o el programa pot produir errors de compilació.

El darrer paràmetre, `new.k.knn` determina el nombre d'observacions que com a mínim entraran a l'ajust local per a les noves observacions. Per defecte, i igual que en la crida del `ldblm`, com a mínim s'hi afegeixen els tres veïns més propers.

Continuació Exemple 5.3

Es volen obtenir les prediccions dels individus que prenen valors en les variables predictores $z1 = (-2, 0, 0, 0, 2)$ i $z2 = (0, 1, 2, 0, 1)$ en els tres models `ldblm` que s'ha ajustat. Les instruccions i resultats en R es presenten a continuació:

```
>  z2.new <- as.matrix(c(0,1,2,0,1))
>  z21.new <- matrix(ifelse(z2.new==1,1,0))
>  z22.new <- matrix(ifelse(z2.new==2,1,0))

>  newZ<-cbind(as.matrix(c(-2,0,0,0,2)),z21.new,z22.new)
>  predict(ldblm1,newdata1=newZ)

 prediction for new data:
[1] -18.588611   1.673624   1.515514  -0.649841  25.003714
```

Pel cas `ldblm2`, s'han de calcular les distàncies dels nous punts respecte els inicials. La mètrica utilitzada és `euclidean`:

```
>  D2_aux <- as.matrix(dist(rbind(Z,newZ),"euclidean"))^2
>  D2_new <- D2_aux[(n+1):(n+5),1:n]
>  predict(ldblm2,newdata1=D2_new,type="D2")
```

```
 prediction for new data:
[1] -18.5614858   0.8597888   1.6417694  -0.2088533  24.8100513
```

Finalment, en el `ldblm3`, s'han de calcular les distàncies entre els nous punts i els inicials tenint present que les dos mètriques utilitzades són diferents (`metric1=manhattan` i `metric2=gower`):

```
>  data.aux<-data.frame(rbind(Z,newZ))
>  D2_aux1 <- as.matrix(daisy(data.aux,"manhattan")^2)
>  D2_new1 <- D2_aux[(n+1):(n+5),1:n]

>  newZ.gow <-data.frame(z1=as.matrix(c(-2,0,0,0,2)),z2=(z2.new))
>  D_aux2 <- daisy(rbind(data.frame(z1=z1,z2=z2),newZ.gow),"gower")
>  D2_aux2 <- disttoD2(D_aux2)
>  D2_new2 <- D2_aux2[(n+1):(n+5),1:n]
>  predict(ldblm3,newdata1=D2_new1,newdata2=D2_new2,type="D2")
prediction for new data:
[1] -19.610907   1.416308   1.726030  -0.200903  25.890223
```

En els tres casos, les prediccions realitzades són diferents.

4.5 dbglm: distance-based generalized linear models

El `dbglm` és una varietat del model lineal generalitzat on la informació dels predictors Z és codificada com una matriu de distàncies entre individus (s'estudia en l'apartat 2.4). A R, quan tenim una variable resposta y d'una certa família i un conjunt de variables explicatives Z, la funció que executa aquest mètode és la funció `glm`.

Per ajustar un model generalitzat basat en distàncies hi ha quatre maneres diferents per fer-ho segons com es tinguin codificades les dades:

```
## S3 method for class 'formula'
dbglm(formula,data,family=gaussian,...,
        metric="euclidean",weights,maxiter=100,eps1=1e-10,
        eps2=1e-10,rel.gvar=0.95,eff.rank=NULL,offset,
        mustart=NULL)

## S3 method for class 'dist'
dbglm(distance,y,family=gaussian,weights,
        maxiter=100,eps1=1e-10,eps2=1e-10,rel.gvar=0.95,
```

```
        eff.rank=NULL,offset,mustart=NULL,...)

## S3 method for class 'D2'
dbglm(D2,y,...,family=gaussian,weights,maxiter=100,
        eps1=1e-10,eps2=1e-10,rel.gvar=0.95,eff.rank=NULL,
        offset,mustart=NULL)

## S3 method for class 'Gram'
dbglm(G,y,...,family=gaussian,weights,maxiter=100,
        eps1=1e-10,eps2=1e-10,rel.gvar=0.95,eff.rank=NULL,
        offset,mustart=NULL)
```

Paràmetres d'entrada

Els paràmetres `formula`, `distance`, `D2` i `G` tenen el mateix significat que en el `dblm` i són els que caracteritzen la funció a escollir: mètode `dbglm` de classe `formula`, `dist`, `D2` o `Gram` respectivament.

La resta, són comuns per a les quatre vies de regressió. L'atribut `family` determina la distribució de probabilitat inherent a la component aleatòria del model. Per defecte `gaussian`, equival al model de regressió basat en distàncies usual. Permet la modelització de totes les famílies de probabilitat definides en la funció `family` de l'R.

El vector numèric `weights` fa possible ponderar la importància a priori de cada individu en el model. `maxiter` indica el nombre d'iteracions màxim que l'algoritme IWLS, adaptat a distàncies, realitza. Per defecte val 100 i el programa s'atura quan l'estimació del valor esperat de la resposta (`mu`) convergeix, la desviància convergeix o quan es sobrepassen les `maxiter` iteracions.

Els atributs `eps_1` i `eps_2` determinen la tolerància de convergència dels criteris d'aturada. El primer dels dos fa referència a la desviància. Convergeix quan

$$\left|\frac{dev - dev_{old}}{dev_{old}}\right| < eps_1.$$

El segon criteri d'aturada té en compte la `mu` i convergeix quan

$$\left|\frac{mu - mu_{old}}{mu_{old}}\right| < eps_2.$$

L'atribut `rel.gvar` indica la variabilitat geomètrica relativa que com a mínim ha de contenir cada iteració `dblm` en l'algoritme IWLS (per defecte

el 95% de variabilitat de les dades). L'argument `eff.rank` determina la segona via per fixar el nombre de coordenades euclidianes en cada `dblm`. Per defecte val `NULL`, i quan es fixa a un cert valor (entre 1 i $n-1$) es sobreposa a `rel.gvar`.

Els paràmetres `offset` i `mustart` determinen un coneixement *a priori* de les dades. En el `offset` s'hi pot indicar una component coneguda per ser afegida en el predictor lineal durant l'ajust. Forçar que l'`offset` valgui zero pot ser necessari en algun dels casos. Pel que fa al `mustart`, s'especifica els valors inicial de les mitjanes en el procés d'ajust. Per defecte, es pren els valors previstos d'un ajust `dblm`.

Per a la distribució Gamma, el domini de la funció d'enllaç canònica (la inversa) no és el mateix que el permès en μ. De fet, el predictor lineal pot donar valors negatius, obtenint de manera inapropiada valors en μ també negatius. Si es produeix aquest cas, `dbglm` para el procés d'estimació amb un missatge d'error. Una alternativa és usar una funció d'enllaç no canònica (per exemple: la inversa quadràtica, la identitat o la funció logaritme).

Valors resultants

En ajustar un model lineal generalitzat `dbglm` s'obté un objecte de classe `dbglm` i `dblm` amb les següents components:

`residuals`	La resposta menys els valors ajustats en l'última iteració del `dblm` (fitted values).
`fitted.values`	Els valors ajustats obtinguts en l'últim `dblm`.
`family`	Distribució de probabilitat de la component aleatòria.
`deviance`	Mesura de la "*maldat*" (badness) de l'ajust. Proporcional a dos vegades la diferència de log-versemblances entre la màxima que es pot aconseguir i la del model actual estimat.
`aic.model`	Criteri d'informació d'Akaike. Igual a dos vegades la resta entre la màxima log-versemblança i el nombre de paràmetres (`eff.rank`). Es calcula diferent per a cada família, en la binomial i la Poisson la dispersió es fixa a 1 i el número de paràmetres és el rang efectiu. Per a les famílies `gaussian`, `Gamma` i `inverse gaussian` la dispersió s'estima a partir de la desviància residual i el número de paràmetres

	és el rang efectiu + 1. Per a les famílies que s'ajusten per quasi-versemblança el valor del AIC és `NA`.
`df.residual`	Els graus de llibertat dels residus.
`df.null`	Els graus de llibertat residuals pel model nul.
`null.deviance`	La desviància pel model nul.
`iter`	Nombre d'iteracions (`dblm`) del algoritme IWLS.
`prior.weights`	pesos originals descrits en el paràmetre d'entrada `weights`.
`weights`	Els pesos actius, els pesos en la última iteració `dblm`.
`y`	La resposta usada per ajustar el model.
`convcrit`	Criteri de convergència: `"DevStat"` (stopping criterion 1), `"muStart"` (stopping criterion 2), `"maxiter"` (si s'ha superat el màxim d'iteracions permés)
`gcv`	GCV estimat.
`aic`	Akaike Value Criterium del model.
`bic`	Bayesian Value Criterium del model.
`H`	La matriu projectora Hat matrix en l'última iteració del `dblm`.
`rel.gvar`	Variabilitat geomètrica relativa en l'últim `dblm` ajustat.
`eff.rank`	Dimensió efectiva utilitzada en l'última iteració `dblm`.

Un objecte de la classe `ldblm` permet usar els mètodes genèrics `print`, `summary`, `plot` i `prediction`.

Exemple 5.4

S'estudien dos casos diferents. En el primer la família de distribució és la Poisson i en el segon és Binomial. Les dades són simulades de la següent manera:

```
> set.seed(45)                    # llavor Z
> z <- rnorm(100)                 # var. explicatives
> yP <- rpois(100, exp(1+z))      # poisson
> yB <- rbinom(100, 1, plogis(z)) # binomial
```

Cas Poisson:

En el cas de la Poisson, si es volgués ajustar un model GLM ordinari per la funció R `glm`, s'hauria d'executar la comanda

```
> glm1 <- glm(yP ~z, family=poisson(link = "log"))
```

En el paràmetre `family` s'especifica la distribució de la component aleatòria, així com la funció d'enllaç que s'utilitzarà (en aquest cas la funció logaritme). De forma idèntica es pot ajustar un model basat en distàncies `dbglm`. Si la mètrica utilitzada és l'euclidiana i es deixen tots els atributs per defecte la crida de la funció és la mateixa que en un `glm`:

```
> dbglm1 <- dbglm(yP ~z, family=poisson(link = "log"))
```

A més, el resultat de l'ajust és exactament el mateix. Calculant la suma de quadrats residual pels dos casos ja es veu que no varien:

```
> sum((glm1$fitted.values-yP)^2)
[1] 757.5214
> sum((dbglm1$fitted.values-yP)^2)
[1] 757.5214
```

En aquest cas, únicament hi ha un regressor `Z` pel què no té molt sentit plantejar la modelització amb una altre funció de distàncies (donarien els mateixos resultats).

Cas Binomial:
La manera d'utilitzar la funció `dbglm` per a una altra família no canvia, únicament cal especificar la distribució correcta en el paràmetre `family` i tots els altres atributs es poden mantenir iguals.

```
D2 <- as.matrix(dist(z,"minkowski",p=1))^2
class(D2) <- "D2"
dbglm2 <- dbglm(D2,yB,family=binomial(link = "logit"))
```

Ara s'ha ajustat el model tenint present que la funció de distàncies és `minkowski` de grau 1. Un cas que encara no s'havia experimentat és ponderar les observacions per uns pesos *a priori*. Es defineixen els pesos tal que valors grans de Z tinguin un pes major que els valors petits en l'ajust del model.

```
 > weight <- abs(z)
 > dbglm3 <- dbglm(dist(z),yB,family=binomial(link = "logit"),
                     weights=weight)
```

Les estimacions canvien considerablement. La tendència creixent en els valors previstos del anterior model queda més suavitzada (s'observa en la Figura 4.5).

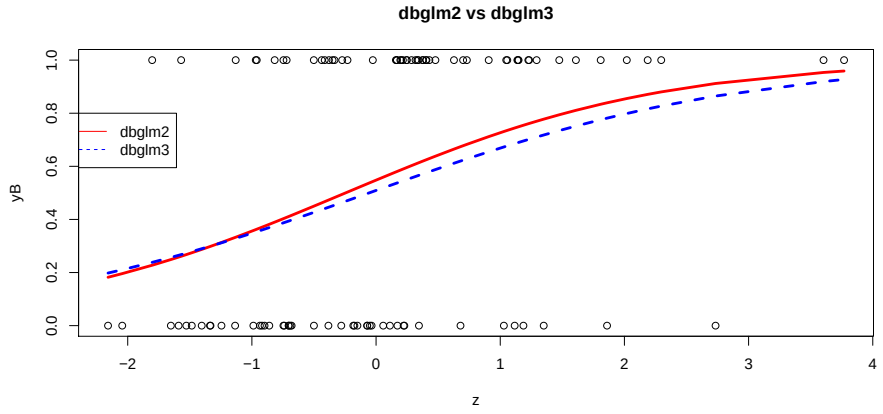

Figura 4.5: Regressió general per un model Binomial i pesos constants (`dbglm2`) i no constants (`dbglm3`).

Mètode `print.dbglm`

El `print` d'un objecte de classe `dbglm` defineix els elements que apareixeran en pantalla a l'ajustar un model lineal generalitzat basat en distàncies per la funció `dbglm`.

Continuació Exemple 5.4

En el cas d'exemple, la sortida obtinguda del model ajustat `dbglm1` és:

```
> print(dbglm1)
Call:  dbglm.formula(formula = yP ~ z, family = poisson(link = "log"))

family: poisson
metric: euclidean

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:      1652
Residual Deviance: 133.3        AIC: 390.3
```

Mètode `summary.dbglm`

El `summary` d'un objecte de classe `dbglm` calcula i resumeix les components més significatives generades al realitzar un `dbglm`. Es crida amb la instrucció

```
summary(object,dispersion,...)
```

on `object` defineix una instància de classe `dbglm` i el paràmetre `dispersion` indica la dispersió en la família usada. Aquest pot ser útil per contrastar els casos de sobredispersió en els models de família Poisson. El `summary` retorna una llista de classe `summary.dbglm` amb els següents elements:

`call`	La crida a la funció `dbglm`.
`family`	Família usada en l'ajust `dbglm`.
`family`	Distribució de probabilitat de la component aleatòria.
`deviance`	Mesura de *badness* of fitt del model `dbglm` estimat.
`aic`	Criteri d'informació d'Akaike.
`df.residual`	Graus de llibertat residuals.
`null.deviance`	Desviància pel model nul.
`df.null`	Graus de llibertat pel model nul.
`iter`	Nombre d'iteracions `dblm` realitzades per ajustar el model.
`deviance.resid`	Desviància residual per a cada observació: `sign(y-mu)*sqrt(di)`.
`pears.resid`	Es calculen dividint els residus per la desviació estàndard de la y.
`dispersion`	La dispersió, 1 per les famílies binomial i Poisson. Altrament, es calculen per l'estadístic residual de Chi-quadrat (només pels casos de pes positiu) dividit pels graus de llibertat residuals.
`gvar`:	Variabilitat geomètrica ponderada de la matriu de distàncies al quadrat.
`gvec`:	Valors de la diagonal de la matriu de productes interiors G.

Continuació Exemple 5.4

El `summary` del model lineal generalitzat basat en distàncies destaca els resultats més importants en el mètode genèric `print.summary`. En el `dbglm2` la sortida per pantalla resultant és la següent:

```
> summary(dbglm2)
Call:  dbglm.D2(y = yB, D2 = D2, family = binomial(link = "logit"))
```

```
Deviance Residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2.20500 -1.03000  0.58110  0.02446  1.00900  1.72100

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 137.63  on 99 degrees of freedom
Residual deviance: 123.21  on 98 degrees of freedom
AIC: 127.21

Number of Fisher Scoring iterations: 4
Convergence criterion: DevStat
```

S'ha intentat, de nou, que la sortida fos el màxim de similar possible a la usual. En aquest cas, la funció `glm`. Exceptuant els coeficients de regressió, els elements són els mateixos.

Mètode `plot.dbglm`

El mètode `plot` d'un objecte de classe `dbglm` reprodueix les mateixes figures que el `plot` de la classe `dblm`. Es crida exactament igual:

```
plot(x,which=c(1:3, 5),id.n=3,main="",
     cook.levels = c(0.5, 1),cex.id = 0.75,
     type_glm=c("link","response"),...)
```

L'únic atribut propi del `dbglm` és el `type_glm`: determina la manera com es volen expressar els `fitted.values`, si en l'escala del predictor lineal (`"link"`) o en l'escala de la variable resposta (`"response"`).

Continuació Exemple 5.4

En l'ajust del model `dbglm3`, la crida del plot reprodueix els gràfics de la Figura 4.6. Els cinc gràfics són els mateixos que permet un objecte `glm`.

Mètode `predict.dbglm`

El darrer mètode genèric disponible pels objectes `dbglm` és el `predict`. Permet fer prediccions de la variable resposta a partir de la informació dels predictors, ja siguin directament les Z, o codificades com a distàncies entre individus. La manera de trobar les previsions és molt similar al cas del model lineal basat en distàncies `dblm`. La funció a executar és la següent:

```
predict(object,newdata,type=c("link", "response"),
        type_var="Z",...)
```

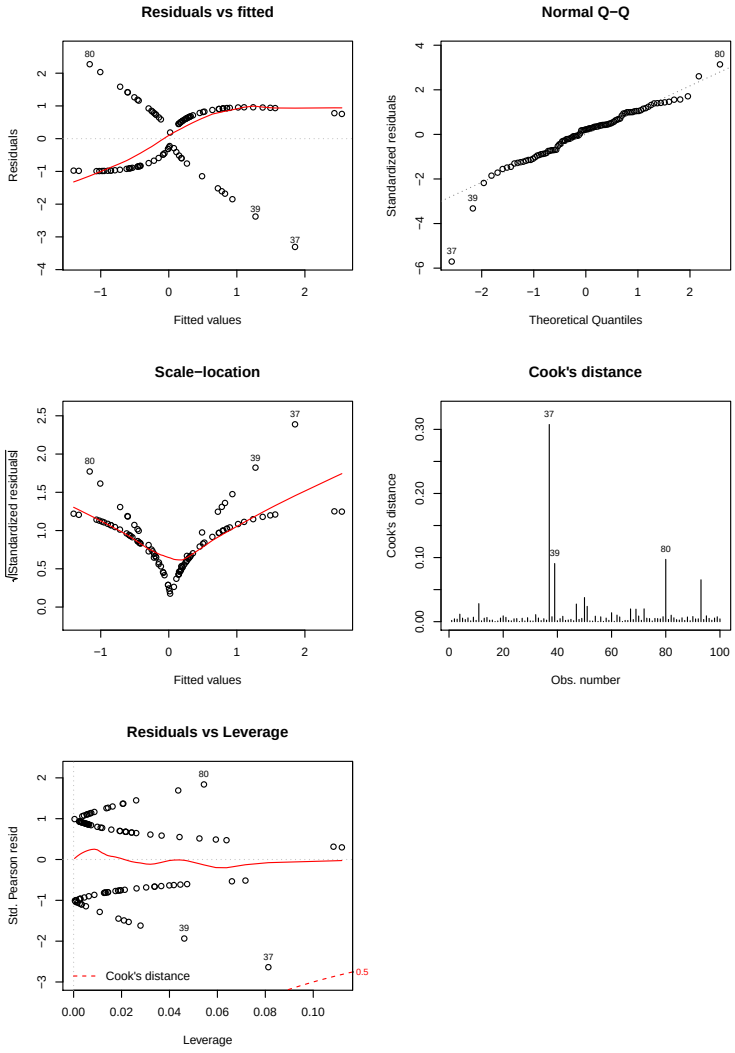

Figura 4.6: Bateria de plots d'un objecte `dbglm` (`dbglm3`)

L'`object` defineix el model `dbglm` ajustat, `newdata` conté la informació dels predictors per a les noves observacions, `type` indica en quina escala es retornaran les previsions (en la del predictor lineal `"link"` o en la resposta `"response"`) i `type_var` determina el format en que s'han introduït les variables predictores al `newdata`. `type_var="Z"` indica que el `dbglm` ajustat s'ha realitzat per la primera de les funcions (en format `formula`) i `newdata` conté els valors de les variables explicatives pels nous individus. S'ha de fixar `type_var="D2"` quan les noves dades són distàncies al quadrat (de classe `D2`) i l'ajust `dbglm` s'ha realitzat per un objecte de classe `dist` o un de classe `D2`. Finalment, `type_var="G"` si s'ha utilitzat la última de les funcions per ajustar el model (el `dbglm` de classe `Gram`) i la informació en `newdata` són les files de la matriu `Gram` pels nous individus.

Continuació Exemple 5.4

Es volen trobar les prediccions dels individus que prenen valors en la variable predictora $z = (-2, -1, 0, 1, 2)$. Es realitzen les prediccions pel cas que la y sigui de la família de Poisson (el primer `dbglm`) i de la família Binomial (els dos següents).

En el `dbglm1`, les prediccions són les següents:

```
> predict(dbglm1,newdata=c(-2,-1,0,1,2),type="response")
           [,1]
[1,]  0.2941066
[2,]  0.8433453
[3,]  2.4182772
[4,]  6.9343656
[5,] 19.8841666
```

Pel cas binomial, les prediccions s'obtenen de forma idèntica. El primer dels dos models ajustat prediu els valors de $z = (-2, -1, 0, 1, 2)$ de la següent manera:

```
> Z_new<-as.matrix(c(-2,-1,0,1,2))
> D_aux<-dist(as.matrix(c(z,Z_new)),"minkowski",p=3)
> D2_aux<-disttoD2(D_aux)
> D2_new <- D2_aux[1:n,(n+1):(n+5)]
> predict(dbglm2,newdata=t(D2_new),type="response",type_var="D2")
          [,1]
[1,] 0.2014264
[2,] 0.3560630
[3,] 0.5479563
[4,] 0.7265774
[5,] 0.8534895
```

Pel que fa referència a l'ajust `dbglm3`, els resultats difereixen respecte el `dbglm2`:

```
> D_aux<-dist(as.matrix(c(z,Z_new)))
> D2_aux<-disttoD2(D_aux)
> D2_new <- D2_aux[1:n,(n+1):(n+5)]
> predict(dbglm3,newdata=t(D2_new),type="response",type_var="D2")
          [,1]
[1,] 0.2153451
[2,] 0.3479463
[3,] 0.5092100
[4,] 0.6685795
[5,] 0.7968449
```

4.6 ldbglm: local distance-based generalized linear models

El `ldbglm` modelitza la versió basada en distàncies de l'ajust no paramètric de versemblança local (s'estudia a l'apartat 2.5). El mètode ordinari (el que no és distance-based) no està programat directament en el paquet base de l'R. Es recomana la utilització del paquet `sm`, que conté una bateria de funcions que apliquen l'ajust local per a models lineals generalitzats. Es tracten de les funcions `sm.binomial`, si la família és Binomial, i `sm.poisson`, pel cas que aquesta sigui Poisson. També cal destacar la funció `mgcv` del paquet `mgcv` que realitza una tasca similar.

En la versió basada en distàncies implementada, el model local generalitzat es pot obtenir, a l'igual que tots els mètodes del `dbstats`, de les següents quatre maneres:

```
## S3 method for class 'formula'
ldbglm(formula,data,...,family=gaussian(),kind.of.kernel=1,
       metric1="euclidean",metric2=metric1,method="GCV",
       weights,user_h=NULL,h.range=NULL,noh=10,k.knn=3,
       rel.gvar=0.95,eff.rank=NULL,maxiter=100,eps1=1e-10,
       eps2=1e-10)

## S3 method for class 'dist'
ldbglm(dist1,dist2=dist1,y,family=gaussian(),kind.of.kernel=1,
       method="GCV",weights,user_h=quantile(dist1,.25),
       h.range=quantile(as.matrix(dist1),c(.05,.5)),noh=10,
       k.knn=3,rel.gvar=0.95,eff.rank=NULL,maxiter=100,
       eps1=1e-10,eps2=1e-10,...)
```

```
## S3 method for class 'D2'
ldbglm(D2_1,D2_2=D2_1,y,family=gaussian(),kind.of.kernel=1,
       method="GCV",weights,user_h=quantile(D2_1,.25)^.5,
       h.range=quantile(as.matrix(D2_1),c(.05,.5))^.5,noh=10,
       k.knn=3,rel.gvar=0.95,eff.rank=NULL,maxiter=100,
       eps1=1e-10,eps2=1e-10,...)

## S3 method for class 'Gram'
ldbglm(G1,G2=G1,y,kind.of.kernel=1,user_h=NULL,
       family=gaussian(),method="GCV",weights,h.range=NULL,
       noh=10,k.knn=3,rel.gvar=0.95,eff.rank=NULL,maxiter=100,
       eps1=1e-10,eps2=1e-10,...)
```

Paràmetres d'entrada

No hi ha cap paràmetre d'entrada nou, tots s'han comentat en apartats anteriors. Recordar que el `ldbglm` conté conceptes del `ldblm`, de fet el `ldblm` és un cas particular del `ldbglm` quan el terme d'error és distribueix normalment i la funció d'enllaç és la identitat. Per tant, hereta conceptes ja vistos com tenir dues mètriques (en el mètode `formula`), distàncies (mètode `dist` i `D2`) o matrius G (mètode `Gram`). La primera matriu s'utilitza pel càlcul dels pesos definits per un nucli (paràmetre `kind.of.kernel`) i la segona per cada iteració `dbglm`. Els altres atributs que comparteixen les dues funcions són per escollir un ample de banda h. El `method` indica quin criteri de selecció automàtica s'usa per escollir la h òptima: `"GCV"` (per defecte), `"OCV"`, `"AIC"`, `"BIC"` i `"user_h"`. Pel cas `"user_h"` l'usuari pot elegir manualment el paràmetre de suavitzat en l'atribut `"user_h"`. `noh` fa referència al número de h's avaluades a dins d'un rang de valors definit en `h.range`. Per últim, `k.knn` indica el número de veïns mínims en l'ajust local per a cada observació.

Per altra banda, el `ldbglm` modelitza el mateix que el `dbglm`. Per això els atributs `family`, que especifica la distribució del model estadístic, el `rel.gvar`, que fixa el rang efectiu tal que s'aconsegueix com a mínim la variabilitat geomètrica relativa `rel.gvar`, el `eff.rank`, que fixa el rang efectiu en cada `dblm` i sobreposa la informació del `rel.gvar`, el `maxiter`, que fixa les iteracions fisher scoring màximes i `eps1`, `eps2` (criteris de convergència), són comuns al `dbglm`.

Valors resultants

Ajustant un model `ldbglm` s'obté un objecte de classe `ldbglm` que conté els següents elements:

`residuals`	La resposta menys els valors ajustats (fitted values).
`fitted.values`	Els valors ajustats per a cada individu del model.
`h_opt`	h òptima si `method!="user_h"`.
`family`	Model estadístic a ajustar.
`y`	La resposta usada per ajustar el model.
`S`	La matriu de suavitzat projectora S: $\hat{y} = S * y$.
`weights`	Els pesos especificats en el model.
`call`	Guarda la crida efectuada de la funció.
`dist1`	Matriu de distància (`dist` o `D2`) usades pel càlcul del pesos kernel.
`dist2`	Matriu de distància (`dist` o `D2`) usades pels ajustos `dbglm`.

Un objecte de la classe `ldbglm` també té disponibles els mètodes genèrics `print`, `summary`, `plot` i `prediction`.

Exemple 5.5

S'ha reproduït, amb dos conjunts de dades diferents, la utilització del `ldbglm` per a les famílies de probabilitat Gamma i Poisson. A més, s'ha comparat amb els resultats que donaria si es realitzés el mateix ajust de forma paramètrica amb el `dbglm`.

Cas Gamma

Primerament, un exemple que fa referència a la distribució Gamma. Les dades s'han simulat de la següent manera:

```
> set.seed(6)
> z <- rnorm(100)
> y <- rgamma(100, 1+z+z^2)
```

Es pretén modelitzar la variable resposta `y` a partir del predictor `z`. S'ajusta un model `dbglm` de família Gamma, distàncies calculades per la mètrica `"euclidean"` i funció d'enllaç canònica (`"inverse"`):

```
> dbglm1<-dbglm(y~z, family=Gamma(link = "inverse"))
```

Alternativament, considerant la versió local del `dbglm`, es modelitza de nou la relació de la y amb la z.

```
> ldbglm1 <- ldbglm(y ~z,noh=5,family=Gamma(link = "log"))
```

La funció d'enllaç del model `ldbglm1` és la `"log"`. S'ha intentat, com en el `dbglm1`, per la funció `inverse` i el programa es para. El motiu és que per alguns amples de banda h, el predictor lineal dóna alguna component del valor esperat μ negativa. El missatge d'error d'R és el següent:

```
> ldbglm1 <- ldbglm(y ~z,noh=5,family=Gamma(link = "inverse"))
Error in dbglm.dist(y, D2, family = family, weights = weights,
maxiter = maxiter,  :
  The linear predictor is negative and some components of 'mu' are < 0.
The Deviance in the iteration 1 is NULL,
     Try to change the link to any one different than: 'inverse',
     'identity', 'logit' or '1/mu^2'
```

Una solució és utilitzar una altra funció d'enllaç (en aquest cas `"log"`). A més el `ldbglm1` ha estat avaluat per cinc h's diferents (`noh=5`) elegint la que minimitza el criteri `GCV`. Les diferències entre el model `dbglm1` i el `ldbglm1` es poden apreciar en la Figura 4.7.

La suma de quadrats residuals, tant del model generalitzat paramètric com del model local, són:

model	**SQR**
dbglm1	259.0436
ldbglm1	159.2437

El model local `ldbglm1` aconsegueix explicar millor la variabilitat de la resposta que el model `dbglm1`.

Cas Poisson

Un segon cas a modelitzar fa referència a la família Poisson. Es simulen les dades a tractar de la següent manera:

```
> n <- 200
> set.seed(29)
> z <- rnorm(200)
> yP <- rpois(200, (1+z+z^2))
```

Cal assenyalar que s'ha configurat la z i la y de tal manera que entre elles hi ha una relació quadràtica. Es poden ajustar de nou tant un model local `ldbglm` com un `dbglm`. S'agafen únicament les primeres 170 observacions (mostra entrenament), les 30 restants es deixen per avaluar la capacitat de predicció dels models (mostra test).

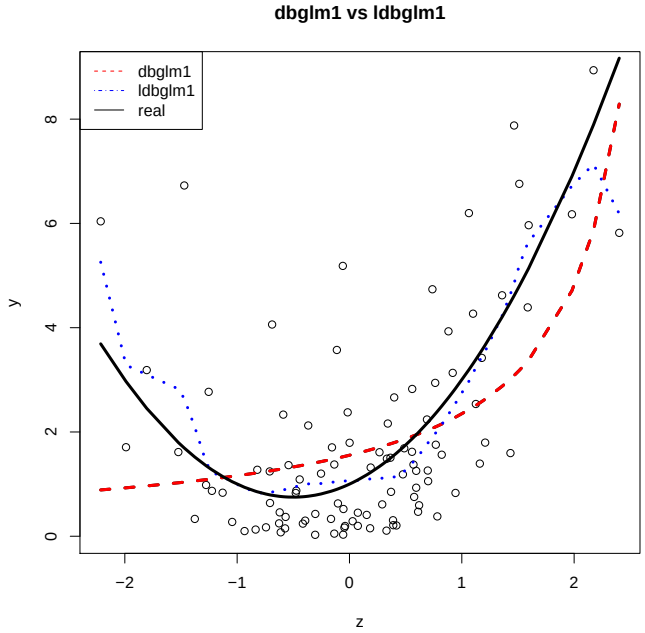

Figura 4.7: Comparació d'ajustos en la família Gamma. Cas `dbglm` i cas `ldbglm`.

```
> dbglm2  <-  dbglm(yP[1:170] ~z[1:170],family=poisson(link = "log"))
> ldbglm2 <- ldbglm(yP[1:170] ~z[1:170],family=poisson(link = "log"))
```

El `ldbglm2` capta la relació no lineal, en canvi, el `dbglm2` no (s'observa en la Figura 4.8). La h òptima del model `ldbglm2` és de 1.049347.

També es pot utilitzar la família de probabilitat Quasi-Poisson. La única diferència respecte la família Poisson es troba en el paràmetre dispersió. No és fixat a 1 com en la Poisson (o també en la binomial), la dispersió és calculada. Per tant, permeten modelar casos de sobredispersió:

```
> ldbglm3<-ldbglm(dist(z[1:170],"manhattan"),y=yP[1:170],
                  family=quasipoisson(link = "log"))
```

La mètrica utilitzada per a calcular els pesos és de `manhattan`. Pel que fa al criteri de selecció de la h és el que minimitza l'`AIC`. La `h` òptima és de 1.042616.

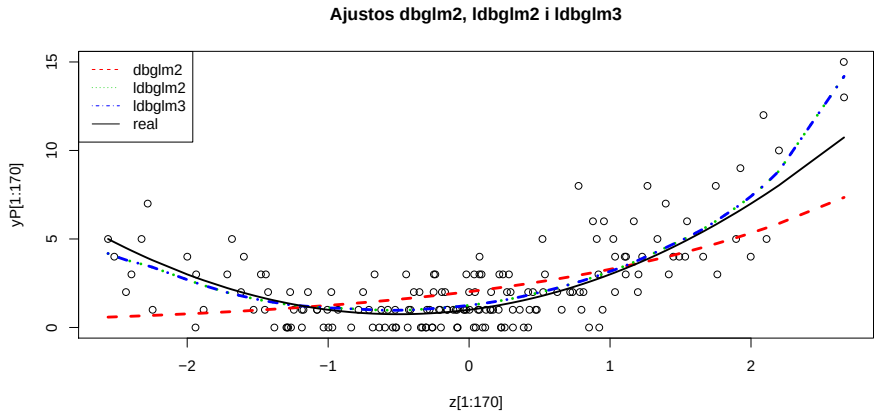

Figura 4.8: Comparació d'ajustos. Cas `dbglm` versus casos `ldbglm`. La relació entre la resposta i la z és quadràtica i el model `dbglm2` no ho capta. El model local `ldbglm1` presenta una funció amb molta variabilitat, indica que el paràmetre de suavitzat ha d'augmentar.

Mètode `print.ldbglm`

La sortida per pantalla quan es crida un objecte de classe `ldbglm` és definit pel mètode `print`.

Continuació Exemple 5.5

En el cas d'exemple, la sortida obtinguda del model ajustat `ldbglm1` és:

```
> ldbglm1

call:   ldbglm.formula(formula = y ~ z,
            family = Gamma(link = "log"), noh = 5)

method= GCV,      kind of kernel= (1) Epanechnikov
metric1: euclidean
metric2: euclidean
optimal bandwidth h : 0.476393
Generalized cross-validation estimate of the prediction error : 1.930496e-02
```

Mètode `summary.ldbglm`

El `summary` d'un objecte de classe `ldbglm` emmagatzema els elements més importants de l'ajust d'un model `ldbglm`. Es crida amb la instrucció

```
summary(object,...)
```

On `object` defineix una instància de classe `ldbglm`. El `summary` retorna una llista de classe `summary.ldbglm` amb els següents elements:

`nobs`	Nombre d'observacions.
`r.squared`	Coeficient de determinació R^2.
`trace.hat`	Traça de la matriu de suavitzat.
`call`	Defineix la crida a la funció `ldbglm`.
`residuals`	Resposta menys valors previstos.
`family`	Model estadístic de la component aleatòria.
`kind.kernel`	Funció Kernel utilitzada pel càlcul dels pesos.
`method`	Criteri seleccionat per optimitzar l'ample de banda.
`h_opt`	Ample de banda òptim i escollit per ajustar el model.
`crit.value`	Valor del criteri d'optimització del bandwidth.

Continuació Exemple 5.5

El `summary` del model local lineal generalitzat basat en distàncies destaca els resultats més importants en el mètode genèric `print.summary`. En el `ldbglm1` la sortida per pantalla resultant és la següent:

```
> summary(ldbglm1)
call:   ldbglm.formula(formula = y ~ z, family = Gamma(link = "log"),
    noh = 5)

Residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-3.0500 -0.7800 -0.3600 -0.0005  0.4650  4.1300

Number of Observations: 100
R-squared : 0.5862
Trace of smoother matrix: 5.99
family: Gamma


kind of kernel= (1) Epanechnikov
optimal bandwidth h : 0.476393
GCV value criterion : 1.930496e-02
```

Mètode `plot.ldbglm`

Els gràfics que es poden obtenir d'un objecte de classe `ldbglm` amb la instrucció `plot` són els mateixos que els especificats en el `plot.ldblm`. Es crida de la següent manera:

```
plot(x,which=c(1,2),id.n=3,main="",...)
```

Hi ha 3 gràfics disponibles, valors previstos envers la resposta, residus envers la resposta i els valors del criteri definit en el paràmetre `method` per les diferents h's avaluades. Només és aplicable si `method =c("OCV","GCV","AIC","BIC")`.

Continuació Exemple 5.5

En el cas d'exemple, els tres plots resultants del model ajustat `ldbglm1` es poden observar en la Figura 4.9.

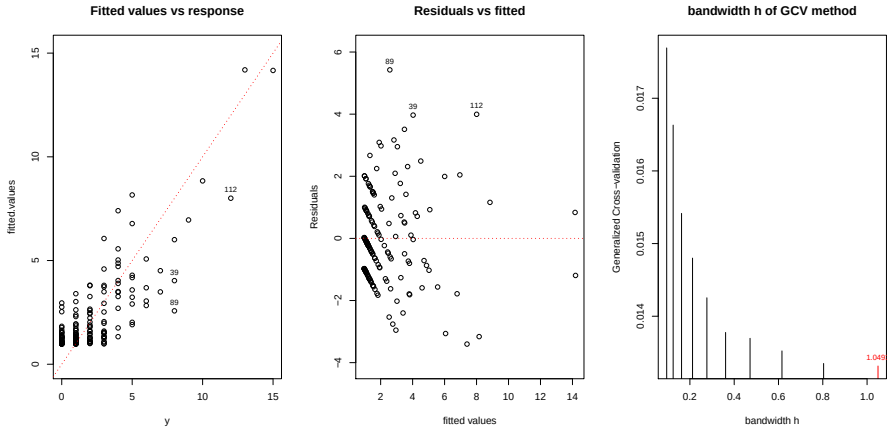

Figura 4.9: Bateria de plots del `ldbglm2`.

Cal adonar-se, en el tercer dels gràfics, com l'ample de banda òptim es troba en l'últim dels valors avaluats. S'hauria d'augmentar el `h.range` i tornar a optimitzar el procés.

Mètode `predict.ldbglm`

L'últim dels mètodes genèrics del `ldbglm` és el `predict`. La crida de la funció és la següent:

```
predict(object,newdata1,newdata2=newdata1,
        new.k.knn=3,type=c("link","response"),
        type_var="Z",...)
```

`object` és la instancia de classe `ldbglm` creada, `newdata1` i `newdata2` defineixen els valors del nous punts en els predictors Z (si s'ha utilitzat `ldbglm` de classe `formula`), les distàncies al quadrat (pel `ldbglm` de classe `dist` i `D2`) i les files de la matriu de productes interiors (pel `ldbglm` de classe `Gram`).

`new.k.knn`, per defecte 3, especifica el nombre d'observacions mínim en la realització de cada ajust local. El `type` indica l'escala que es volen fer les prediccions, predictor lineal (`"link"`) o resposta (`"response"`). Per últim `type_var` juga un paper conjunt amb `newadata`. `type_var` ha de valer `"Z"` si en `newdata` hi ha els regressors, `"D2"` si hi ha les distàncies al quadrat i `"G"` si hi ha els productes interiors.

Continuació Exemple 5.5

Es volen fer prediccions en el cas d'exemple de la família Poisson, és a dir els models `ldbglm2` i `ldbglm3` (també en el cas paramètric `dbglm2`). De les 200 dades inicials, 170 s'han usat per ajustar el model i les 30 restants es guarden per fer prediccions. Les funcions que cal utilitzar són:

```
pr1<-predict(dbglm2,newdata=z[171:200],type="response")
pr2<-predict(ldbglm2,newdata1=z[171:200],type_var="Z",
             type="response")
pr3<-predict(ldbglm3,
             newdata1=as.matrix(dist(z,"canberra")^2)[91:100,1:90],
             type_var="D2",type="response")
```

Les sumes d'errors al quadrat pels tres casos són les següents:

```
cas pr1 (dbglm2):   70.59712
cas pr2 (ldbglm2):  69.33165
cas pr3 (ldbglm3):  69.36913
```

Les sumes d'errors de predicció al quadrat són molt similars. Sembla que un model lineal generalitzat és suficient per estimar la resposta.

4.7 dbplsr: distance-based partial least squares

L'últim dels mètodes basats en distàncies programat en el `dbstats` és el `dbplsr`. Es tracta de la versió basada en distàncies dels mínims quadrats parcials. El package d'R `pls` conté la funció `plsr` que tracta la problemàtica del partial least squares en la modelització d'una variable resposta numèrica a partir de p variables explicatives. Es tracta d'una funció molt potent amb varis algorismes aplicables per ajustar models multivariants.

La versió basada en distàncies implementada segueix l'algorisme estudiat en l'apartat metodològic 2.6. El partial least squares basat en distàncies es pot ajustar amb l'R per una de les següents vies:

```
## S3 method for class 'formula'
dbplsr(formula,data,...,metric="euclidean",
        weights,ncomp,method="ncomp")

## S3 method for class 'dist'
dbplsr(distance,y,...,weights,ncomp=ncomp,method="ncomp")

## S3 method for class 'D2'
dbplsr(D2,y,...,weights,ncomp=ncomp,method="ncomp")

## S3 method for class 'Gram'
dbplsr(G,y,...,weights,ncomp=ncomp,method="ncomp")
```

Paràmetres d'entrada

Els atributs `formula`, `distance`, `D2` i `G` són els que determinen quina de les quatre funcions disponibles utilitzar. Exactament igual als altres mètodes `dbstats`, `dbplsr` permet modelitzar els mínims quadrats parcials quan es codifica la informació de les variables com a distàncies (mètode de la classe `dist`), distàncies al quadrat (classe `D2`), matriu de productes interiors G (classe `Gram`), o pel cas més proper al `plsr` amb les variables predictores Z (classe `formula`).

Pel que fa als altres paràmetres, `metric` indica la mètrica a utilitzar per calcular les inter-distàncies entre individus, només necessari pel cas que s'hagi utilitzat la primera de les funcions. Per defecte s'aplica la mètrica euclidiana, encara que també és possible la utilització de la funció de distàncies per `"manhattan` i per `"gower"`. `weights` permet ajustar un model ponderat segons el vector de pesos descrit en `weights`, per defecte tots els individus tenen el mateix pes. El paràmetre `method` determina el mètode que s'utilitza per decidir quantes components són necessàries per trobar el millor model. Hi ha 5 mètodes disponibles: els criteris de minimització `"AIC","BIC","OCV" i "GCV"`, i elegint manualment el nombre de components `method="ncomp"` (per defecte). El `dbpls`, com ja s'ha vist en el capítol 2.6, és un mètode iteratiu, on en cada iteració troba la component que maximitza la covariança entre els predictors i la resposta. Per tant, realitzarà tantes iteracions com components especificades en `ncomp`.

Valors resultants

A l'ajustar una de les funcions proposades `dbplsr`, s'obté una llista de classe `dbplsr` amb les següents components:

`residuals`	Resposta menys valors previstos en cada iteració.
`fitted.values`	Una llista que conté els valors previstos en cada iteració.
`fk`	Una llista que conté els scores (puntuacions) per a cada iteració.
`bk`	Els coeficient de regressió tal que `fitted.values = fk%*%bk`.
`Pk`	Projector ortogonal en l'espai unidimensional de fk.
`ncomp`	Nombre de components incloses en el model.
`ncomp_opt`	Nombre de components òptimes d'acord al criteri escollit al atribut d'entrada `method`.
`weights`	Vector de pesos especificat.
`method`	Mètode per optimitzar el nombre de components especificat.
`y`	La resposta usada per ajustar el model.
`H`	Matriu projectora (hat matrix).
`G0`	Productes interiors centrats i ponderats de la matriu de distàncies al quadrat inicial.
`Gk`	Productes interiors centrats i ponderats en l'última iteració.
`gvar`	Variabilitat geomètrica total.
`gvec`	Valors de la diagonal de `G0`.
`gvar.iter`	Variabilitat geomètrica en cada component.
`ocv`	Validació creuada ordinària de l'error de predicció.
`gcv`	Validació creuada generalitzada de l'error de predicció.
`aic`:	Valor de la Informació d'Akaike del model.
`bic`:	Valor de la Informació Bayesiana del model.

Exemple 5.6

El PLS esdevé un procediment útil quan hi ha més variables predictores que observacions en el model. Les dades del cas que es proposa en aquest exemple estan descrites en Swierenga et al. (1999) i les recupera l'ajuda de la funció `plsr` per explicar-ne el seu funcionament. Es tracta de les dades `yarn`. Es pretén mesurar el comportament de fils creats a partir d'un plàstic anomenat Tereftalat de Poliestirè, conegut per les sigles PET. PET es un polímer que

s'obté per mitja d'una reacció química i és utilitzat per generar envasos de begudes i tèxtils. `yarn` conté 21 valors de fils PET mesurats per la metodologia NIR spectra (Near Infrared Spectorscopy). L'objectiu és fer prediccions de la densitat global dels fils PET a partir de la informació que proporciona l'espectre mesurat en 268 longituds d'ona diferents. De fet, les dades `yarn` es poden tractar com a dades funcionals on cada individu és una corba o funció de les longituds d'ona especificades (s'observa en la Figura 4.10).

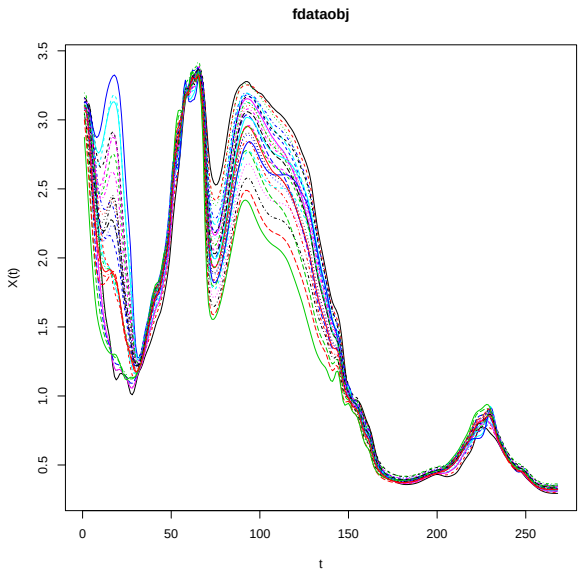

Figura 4.10: NIR Esprectra dels fils PET (Tereftalat de Poliestirè).

En aquest cas d'exemple es pretén explicar el funcionament del `dbplsr`. Al capítol 5 es mostra un cas de modelització d'una variable resposta quan els predictors són dades funcionals. Allà s'entra en detall en què són exactament aquest tipus de dades. De moment es considera que les 268 longituds d'ona conformen els predictors de la resposta. Per obtenir les dades de `yarn` en R s'ha de tenir instal·lat el paquet `pls`:

```
> require(pls)
> data(yarn)
```

L'ajust basat en distàncies del PLS es realitza en una mostra d'entrenament (les 21 primeres observacions) executant les següents instruccions:

```
> dades.entre<-yarn[yarn$train,]
> yarn.dbplsr1 <- dbplsr(density ~ NIR, data = dades.entre, ncomp=6,
                         method="GCV")
```

La mètrica per calcular les distàncies es deixa per defecte (`"euclidean"`), el nombre d'iteracions (i per tant components estimades) és 6, i el mètode per decidir quantes de les sis components són necessàries és el `GCV`.

Alternativament, es podria ajustar el model `dbplsr` a partir de la matriu d'inter-distàncies entre individus, per exemple utilitzant la funció `daisy` i la mètrica de `gower`:

```
> yarn.dbplsr2 <- dbplsr(daisy(dades.entre$NIR,"gower"),
                         dades.entre$density, ncomp=10, method="GCV")
```

A l'hora de determinar el número de components òptimes per ajustar el `dbplsr` s'ha de ser curós en com interpretar la informació que proporciona el criteri seleccionat, en aquest cas el `GCV`. El nombre de components òptim és el que minimitza el criteri. No obstant, a la pràctica, s'acostuma a tallar el nombre de components quan el valor del criteri s'estabilitza. Quan succeeix això, sol indicar que s'ajusta pràcticament el mateix model. En l'estudi del mètode `plot` d'un objecte `dbplsr` es discuteix aquesta circumstància.

Mètode `print.dbplsr`

El mètode `print` d'un objecte de classe `dbplsr` defineix les components que escup per pantalla l'R quan es vol mostrar el contingut de tal objecte.

Continuació Exemple 5.6

En l'exemple, la sortida obtinguda del model ajustat `yarn.dbplsr1` és:

```
> print (yarn.dbplsr1)
dbplsr.formula(formula = density ~ NIR, data = dades.entre,
               method = "GCV", ncomp = 6)

number of components: 6
metric: euclidean

optimal number of components using method GCV: 6
optimal Generalized cross-validation : 6.948088e-03
```

El número de components òptim pel criteri `GCV` és de 6, amb un valor de 6.95e-03.

Mètode `summary.dbplsr`

El mètode genèric `summary.dbplsr` resumeix l'ajust del partial least square basat en distàncies. Retorna una llista de classe `summary.dbplsr` amb les següents components:

`ncomp`:	Nombre de components incloses en el model.
`r.squared`:	Coeficient de determinació R^2.
`adj.r.squared`:	Coeficient de determinació ajustat R^2_{adj}.
`call`:	Crida a la funció `dbplsr`.
`nresiduals`:	Resposta menys valors previstos en cada iteració.
`sigma`:	Desviació estàndard residual.
`gvar`:	Variabilitat geomètrica total.
`gvec`:	Valors de la diagonal de `G0`.
`gvar.iter`:	Variabilitat geomètrica en cada component.
`method`:	Mètode per elegir el nombre de components òptim.
`crit.value`:	Valor del criteri definit en `method`.
`ncomp_opt`:	Nombre de components òptimes pel criteri seleccionat.

Continuació Exemple 5.6

En el cas d'exemple, es pot recuperar el `summary` del `yarn.dbplsr2` fent:

```
> summary(yarn.dbplsr2)
   dbplsr.dist(y = dades.entre$density,
       distance = daisy(dades.entre$NIR,"gower"),
       ncomp = 10, method = "GCV")

Weighted Residuals using 10 components:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-3.99e-03 -1.86e-03 -5.04e-05  2.24e-15  8.94e-04  7.72e-03

Optimal number of components using method GCV: 10
 GCV value criterion : 1.109903e-06

% variance explained:
      1 comp  2 comp  3 comp   4 comp   5 comp    6 comp    7 comp
R2    94.266 98.6459 99.5964 99.91682 99.96927 99.994399 1.000e+02
adjR2 93.964 98.4954 99.5252 99.89603 99.95902 99.991998 1.000e+02
gvar  32.228 48.9436 58.9566 66.60434 72.08546 75.041568 7.792e+01
crit   2.677  0.7005  0.2326  0.05375  0.02242  0.004649 5.659e-04
         8 comp    9 comp   10 comp
R2    1.000e+02 1.000e+02 1.000e+02
adjR2 1.000e+02 1.000e+02 1.000e+02
```

```
gvar  8.106e+01 8.326e+01 8.558e+01
crit  9.981e-05 1.157e-05 1.110e-06
```

A la primera component, l'`R2` i l'`adjR2` ja són del 94% i la variabilitat geomètrica és del 32%. A la segona component, l'`R2` augmenta a quasi el 99% i llavors ja s'estabilitza i no s'explica pràcticament res més de la resposta. La bondat de l'ajust és excel·lent. En canvi, la proporció de variabilitat de les dades explicada creix molt lentament. Però això no és important: el que es pretén es escollir les components necessàries per tal que les futures prediccions siguin bones, i això es pot aconseguir amb un percentatge molt baix de variabilitat geomètrica explicada. A partir de la setena component l'R^2 és del 100% indicant que s'està interpolant les dades.

Mètode `plot.dbplsr`

El mètode genèric `plot` d'un objecte `dbplsr` conté quatre gràfics disponibles que es seleccionen amb el paràmetre `which`. Hi ha un plot de les noves components scores, resposta envers scores, la contribució en el `R2` de cada component i el valor del `"OCV", "GCV", "AIC" o "BIC"` segons el nombre de components escollit. La crida a la funció `plot` és la següent:

```
plot(x,which=c(1:4),main="",scores_comps=1:2,
     component=1,method=c("OCV","GCV","AIC","BIC"),...)
```

On `x` indica l'objecte a dibuixar i `which` es un vector per determinar els gràfics de sortida. `scores_comps` es un paràmetre específic del primer gràfic. S'indica en un vector numèric quines components, en concepte de scores, es volen creuar. Permet creuar més de dues components (per defecte les dues primeres). L'argument `component` és específic pel segon `plot`. Indica la component amb la qual es creua la resposta per realitzar el gràfic. Per últim `method` és un atribut referent a l'últim gràfic. S'il·lustren els valors del criteri escollit en `method` envers les components incloses en el model.

Continuació Exemple 5.6

En el cas d'exemple, es poden observar els quatre plots del `yarn.dbplsr1`, amb el `mètod="GCV"`, a la Figura 4.11. Analitzant el primer dels quatre gràfics, on es creuen els valors del primer i el segon score, s'observa una estructura molt marcada entre aquests dos que fa sospitar que les dades siguin artificials.

És important adonar-se del fet que ja s'indicava al començar aquest exemple. A vegades és millor comptar les components fins que el criteri (`GCV`) s'estabilitza. En el `print` s'ha vist que el `GCV` mínim s'aconseguia en la 6a component. No obstant, els valors a partir de la 4a components són pràcticament iguals (es veu en el quart gràfic de la Figura 4.11). Amb quatre components sembla que n'hi hauria prou per ajustar el model. El `plot` s'obté amb la instrucció

```
> plot(yarn.dbplsr1)
```

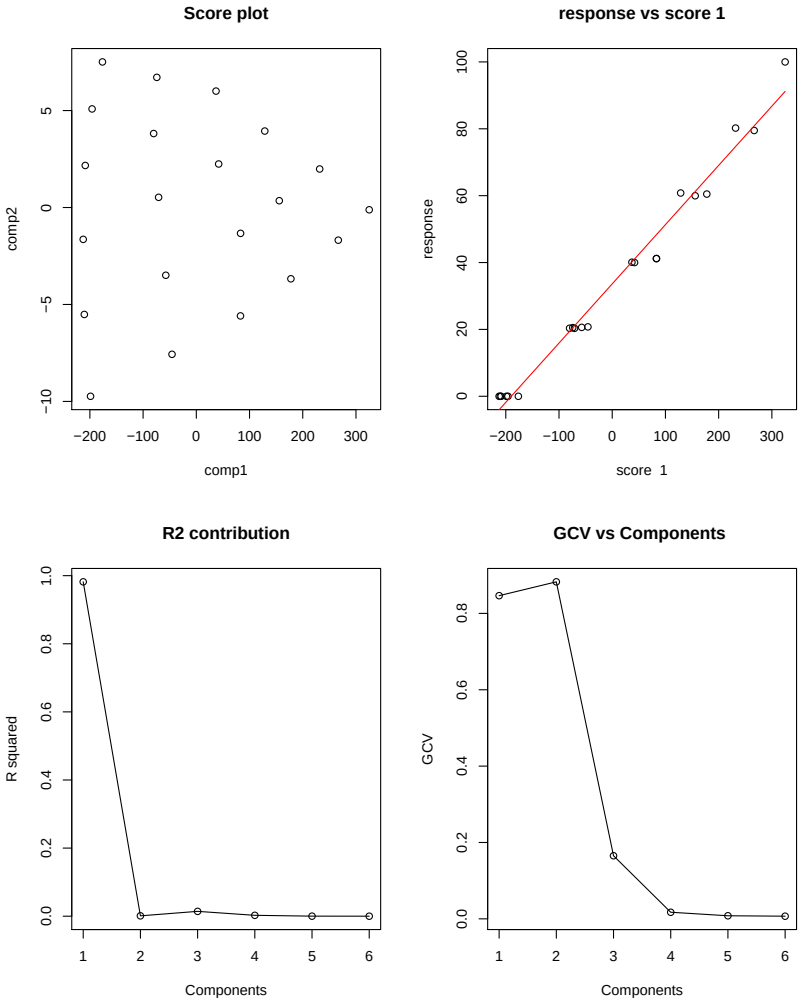

Figura 4.11: Bateria de plots del `yarn.dbplsr`.

Mètode `predict.dbplsr`

El darrer mètode genèric definit per a un objecte `dbplsr` és el `predict`. Defineix la mecànica per realitzar prediccions, el procediment que s'ha de seguir és el mateix que en el `dblm`:

```
predict(object,newdata,type="Z",...)
```

L'atribut `object` és una instància de classe `dbplsr`, `newdata` conté els nous valors de les variables predictives Z (si `type="Z"` i el `dbplsr` ha estat ajustat per la funció `dbplsr` en format `formula`). Les distàncies al quadrat entre els k individus nous i els n originals (si `type="D2"` i el `dbplsr` ha estat cridat per les vies `dbplsr` de classe `dist` o de classe `D2`). Finalment, la matriu de productes interiors G (si `type="G"` i el `dbplsr` s'ha executat per la funció de classe `Gram`).

Val la pena remarcar de nou que s'ha de vigilar que el `type` especificat, la informació que conté `newdata` i com s'ha cridat el procediment `dbplsr` sigui coherent, sinó el programa fallarà.

Continuació Exemple 5.6

Per acabar l'exemple, ja únicament falta realitzar prediccions. Quan s'ha ajustat el model, tant en el `dbplsr1` com el `dbplsr2`, s'han guardat 7 dades per poder mesurar la capacitat de predicció. Es tracta de la mostra test:

```
> dades.test <- yarn[!yarn$train,]
```

Si es prenen 4 components per ajustar el model `yarn.dbplsr1`, argumentant que és a partir de llavors que el criteri GCV s'estabilitza, les prediccions són les següents:

```
> yarn.dbplsr1 <- dbplsr(density ~ NIR, data = dades.entre, ncomp=4,
        method="GCV")
> predict(yarn.dbplsr1,newdata=dades.test$NIR)
          [,1]
[1,] 50.97421
[2,] 51.11887
[3,] 31.78316
[4,] 34.24528
[5,] 30.84914
[6,] 20.63160
[7,] 18.57799
```

Pel que fa al segon model ajustat, el `yarn.dbplsr2`, es podrien agafar únicament les dues primeres components, argumentant que l'aportació de l'R^2 de les components posteriors és pràcticament nul·la. Ara s'obtenen les següents prediccions:

```
> yarn.dbplsr2 <- dbplsr.dist(daisy(dades.entre$NIR,"gower"),
                      dades.entre$density,ncomp=2, method="GCV")
> new.data<-disttoD2(daisy(yarn$NIR,"gower"))[22:28,1:21]
> predict(yarn.dbplsr2,newdata=new.data,type="D2")
         [,1]
[1,] 46.83360
[2,] 52.22487
[3,] 26.72531
[4,] 28.75156
[5,] 30.76070
[6,] 14.31547
[7,] 11.89170
```

Les sumes de quadrats residuals de les prediccions són les següents:

```
SQR yarn.dbplsr1 =  0.31862
SQR yarn.dbplsr2 = 13.51006
```

Per tant, el primer model té una capacitat de predicció millor que el segon. Si s'hagués de triar entre l'un i l'altre, un cop validats, s'escolliria el primer.

Capítol 5

Aplicacions

En aquest capítol s'analitzen exhaustivament dos exemples on els mètodes basats en distància són una alternativa molt vàlida als mètodes estadístics ordinaris. En el primer es pretén trobar un model per fer prediccions d'una variable resposta contínua, a partir de la informació coneguda d'una variable explicativa funcional. Es tracta del conjunt de dades `Wheat` i s'utilitzen les funcions `dblm`, `ldblm` i `dbplsr` de la llibreria `dbstats`. En el segon s'estudia un cas d'identificació de grups de risc per a companyies asseguradores d'automòbils. El conjunt de dades s'anomena `motorins` i s'utilitzen les funcions `dbglm` i `ldbglm` per ajustar un model lineal generalitzat de la família Poisson.

5.1 `Wheat`: aplicació a dades funcionals

Els mètodes estadístics basats en distàncies són mètodes de predicció d'una variable resposta y suposant coneguts els valors en Z (predictors lineals). Aquests no han de ser estrictament numèrics. Això permet la modelització amb variables explicatives categòriques, textuals o, com al cas que es tracta en aquest apartat, dades funcionals.

Introducció a les dades funcionals

Ferraty and Vieu (2006) defineixen una variable funcional com una variable aleatòria Z que pren valors en un espai de dimensió infinita (functional space). En la Figura 5.1 se'n destaca un exemple.

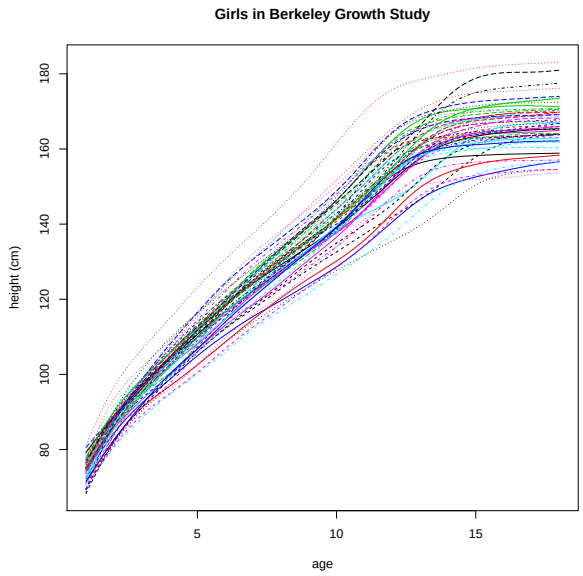

Figura 5.1: Exemple de dades funcionals: mesura de l'alçada de noies de Berkley en funció de la seva edat.

En una variable funcional, cada individu pren uns certs valors en la coordenada y (en el gràfic l'alçada) respecte a la x (l'edat) que resulta suau almenys per l'ull humà. Habitualment es considera que la primera i la segona derivada són contínues. Per alguns conjunts de dades és més informatiu treballar amb les derivades que no amb les funcions originals, per tant s'ha de tenir en compte i computar també les derivades del procés. La y és funció de

x, on x és una variable contínua, per exemple el temps, espectres, longituds d'ona o probabilitats.

L'anàlisi de dades funcionals és un recull de tècniques recents desenvolupades a partir dels anys 90, i un camp d'investigació en l'estadística que està creixent molt en els darrers anys. El motiu principal és que ara és possible mesurar i emmagatzemar dades funcionals. La millora en els instruments de mesura en temps real i la possibilitat d'emmagatzemar grans conjunts de dades fan viable guardar una funció completa per a cada individu d'una experimentació aleatòria.

L'anàlisi de dades funcionals (FDA) descriu i modelitza mostres de dades funcionals. Per exemple, hi ha les versions dels mètodes estadístics més típics adaptats a aquest tipus de dades:

- **Functional descriptive statistics**: mitjana, variança, mediana, moda, etc.
- **Functional linear models**: fregre.lm, fregre.glm, fregre.np, etc. (funcions de la llibreria d'R `fda.usc`).
- **Functional principal component analysis**.

En el cas d'utilitzar `dbstats` es pot replicar el segon punt, tot el que seria la modelització d'una variable resposta a partir de predictors configurats com a dades funcionals.

En R hi ha dues vies principals per aplicar aquestes tècniques. La primera es tracta de la llibreria `fda`, elaborada per Ramsay and Silverman (2003). La segona via és utilitzar el paquet `fda.usc`, realitzat per Febrero-Bande and Oviedo (2011). `fda.usc` és realment potent: conté una llarga llista de mètodes FDA implementats de forma molt eficient. Al llarg de l'anàlisi de l'aplicació `Wheat` es reproduirà tant el tractament per medi del `dbstats` com pels mètodes de `fda.usc`.

Per entendre el problema FDA de regressió com un problema basat en distàncies s'ha de codificar la informació de les dades funcionals en una matriu de distàncies al quadrat. El problema és que ara la dimensió de les dades és infinita, per tant no es poden aplicar les mètriques usuals presentades en l'apartat 4.2, totes elles basades en el sumatori de k termes (amb $k < \infty$). La solució és considerar una semi-mètrica en l'espai funcional. Una semi-mètrica es una funció que satisfà les condicions d'una mètrica euclidiana expressades

en l'apartat 2.1: simetria, diagonal nul·la i desigualtat triangular, però pot trencar la condició no degenerada:

$$\text{si} \quad d(z_1, z_2) = 0 \quad \text{llavors} \quad z_2 = z_1, \forall z_1, z_2 \in Z.$$

A continuació, es detallen algunes possibles semi-mètriques a calcular (Ferraty and Vieu 2006):

- Distància L_2 entre derivades: per a $r = 0, 1, 2, ...$

$$d_r^{deriv}(\chi, \gamma) = \sqrt{\int_a^b \left(\chi^{(r)}(t) - \gamma^{(r)}(t)\right)^2 dt}.$$

- Distància L_2 en l'espai de les q primeres components principals funcionals:

$$d_q^{PCA}(\chi, \gamma) = \sqrt{\sum_{k=1}^{q} (\psi_k^{\chi} - \psi_k^{\gamma})^2}.$$

- Distància L_2 en l'espai de les q primeres components PLS funcionals:

$$d_q^{PLS}(\chi, \gamma) = \sqrt{\sum_{k=1}^{q} (\varphi_k^{\chi} - \varphi_k^{\gamma})^2}.$$

- Semi-mètrica de fourier: aproxima la distància L_2 entre corbes basada en una representació B-spline.

La llibreria `fda.usc` té implementades totes aquestes semi-mètriques (i algunes més) que es criden, un cop carregat el paquet, de la següent manera:

```
semimetric.mplsr(fdata1, fdata2=fdata1, q=2, class1,...)

semimetric.pca(fdata1, fdata2=fdata1, q=1,...)

semimetric.deriv(fdata1, fdata2=fdata1, nderiv=1,
 nknot=ifelse(floor(ncol(DATA1)/3)>floor((ncol(DATA1)-
 nderiv-4)/2),floor((ncol(DATA1)-nderiv-4)/2),
 floor(ncol(DATA1)/3)),...)

semimetric.fourier(fdata1, fdata2=fdata1, nderiv=0,
 nbasis=ifelse(floor(ncol(DATA1)/3)>floor((ncol(DATA1)-
 nderiv-4)/2),floor((ncol(DATA1) - nderiv - 4)/2),
 floor(ncol(DATA1)/3)),period=NULL,...)
```

Fent ús d'una de les semi-mètriques presentades es pot configurar la informació de les dades com una matriu de distàncies i aplicar els mètodes de regressió basats en distàncies implementats al `dbstats`.

Conjunt de dades Wheat

El conjunt de dades `Wheat`, Kalivas (1997), conté 100 mostres de blat mesurades per la metodologia NIR Spectra entre les longituds d'ona 1100nm i 2500nm. L'espectroscòpia NIR (Near-infrared) mesura la longitud d'ona i intensitat d'absorció de llum infraroja en cada mostra. La llum infraroja aplicada a la mostra conté energia per a excitar sobretons i combinacions de vibracions moleculars. Aquesta tècnica s'aplica majoritàriament en camps com la farmacèutica, l'agricultura, l'anàlisi clínic o la química. En tots els casos és molt útil per a la monitorització d'un procés de control.

A més, en el dataset `Wheat` hi ha la mesura, per a cada una de les mostres, del contingut de proteïna i humitat en el blat. L'objectiu és fer prediccions del contingut de la proteïna en noves mostres de blat a partir de la informació funcional que proporciona la metodologia NIR.

De les 100 observacions inicials, 80 són utilitzades per validar el model i les 20 restants per determinar la capacitat de predicció. En la mostra d'entrenament, la que conte els 80 punts per ajustar el model, les dades funcionals del conjunt de funcions originals, les primeres i segones derivades es troben representades en la Figura 5.2 .

Abans de començar a ajustar models, una anàlisi que pot ser d'utilitat fa referència a comparar les correlacions al quadrat entre la resposta i les dades funcionals en cada punt de l'espectre. Aquests valors coincideixen amb l'R^2 d'una regressió lineal on es modelés la resposta a partir d'un únic regressor que prengués els valors de la funció en un punt concret de la longitud d'ona. Això es calcula en un conjunt de valors equiespaiats del argument de les funcions. S'ha realitzat per les dades sense derivar, la primera derivada i la segona. Els resultats són expressats gràficament en la Figura 5.3.

S'hi destaquen certes diferències entre les tres maneres d'expressar les dades funcionals. En les longituds d'ona més baixes sembla que derivant es perd informació de la resposta i és amb les dades sense derivar que es troben les correlacions més altes. Tot i això, a partir de la longitud de 1400nm aproximadament, la diferenciació d'ordre 1 dóna els millor coeficients. La segona derivada no pren valors tant alts si bé té el pic més alt amb una R^2 de pràcticament 0.8. Més endevant es confirmarà quina de les transformació proporciona una millor capacitat de predicció.

Per avaluar la capacitat de predicció de cada un dels models que s'aniran

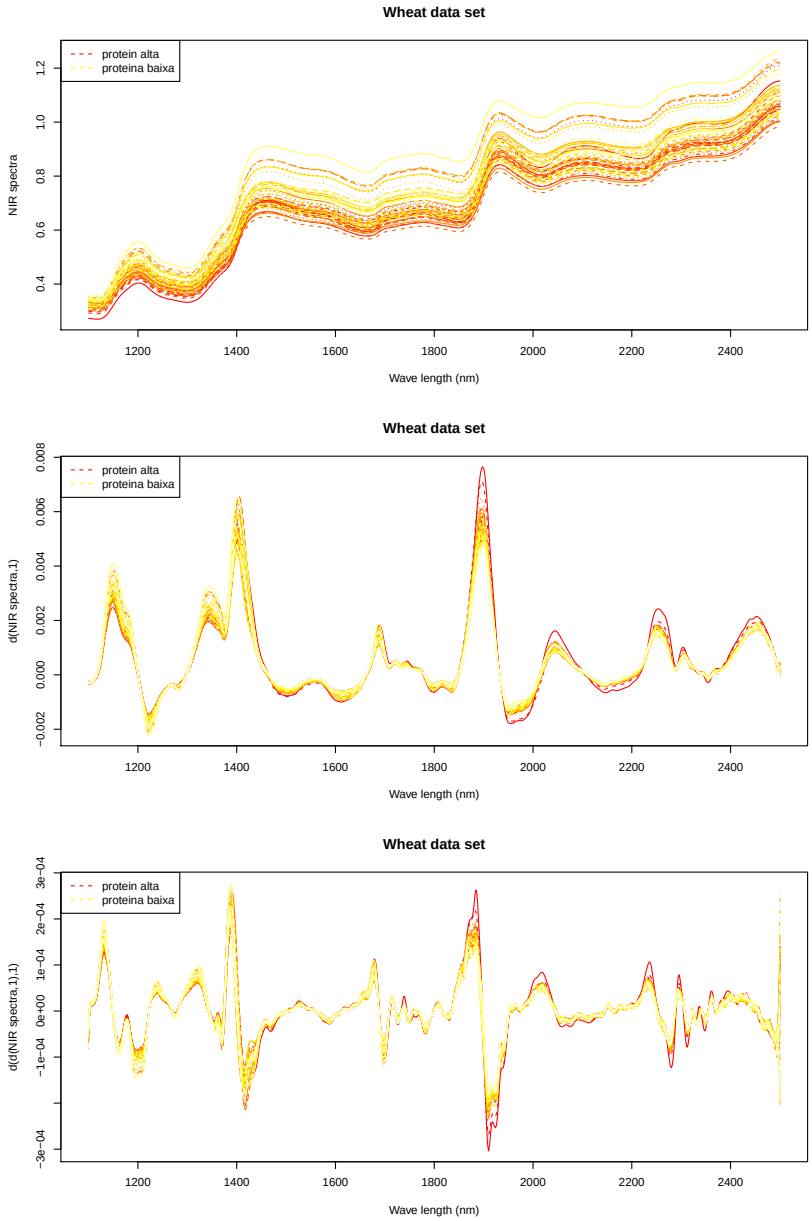

Figura 5.2: Conjunt de dades Wheat: mesura el NIR spectra de mostres de blat. Es distingeixen cada una de les corbes per un color. Els colors formen una paleta que va des del vermell fins al groc de manera que, com més pròxim al vermell, el contingut de proteïna és més alt. Ja a simple vista s'aprecia una diferència marcada entre observacions amb alt i baix contingut de proteïna. En les dades sense derivar la majoria de corbes de proteïna alta prenen valors del NIR més baixos. Pel que fa a la primera i la segona derivada hi ha algun pic on queden marcades, també, les diferències entre el contingut de proteïna.

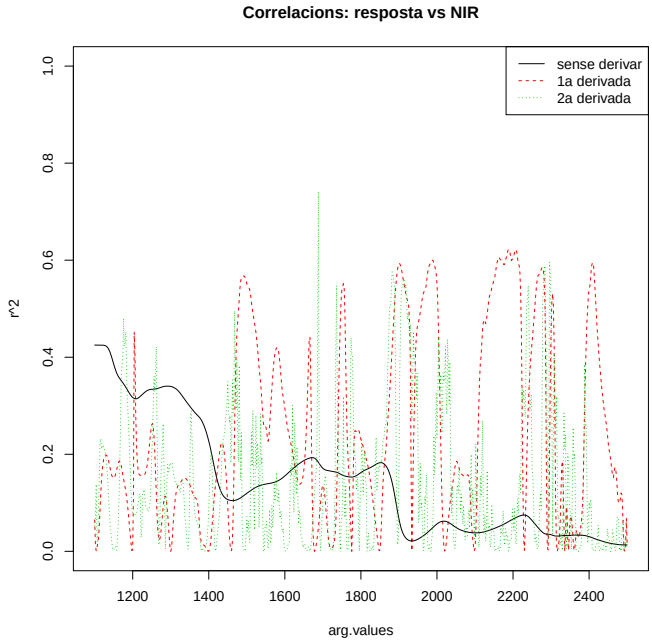

Figura 5.3: Correlacions al quadrat entre la resposta i cada punt de la funció NIR.

ajustant s'ha establert la següent mesura:

$$SQRr = \frac{\sum_{i=1}^{k} (\tilde{y}_i - y_i)^2}{\sum_{i=1}^{k} (y_i - \bar{y}_i)^2}$$

El $SQRr$ (Suma de Quadrats Residuals relativa) dóna la proporció de variabilitat de la resposta en les noves dades no explicada pel model. Si es pren $1 - SQRr$ s'està donant el valor del coeficient de determinació R^2 pels nous casos. Un $SQRr$ alt indica una mala capacitat per fer prediccions del model. En canvi, si aquest és baix, el model serà més propens a fer bones prediccions. Es podria establir un llindar orientatiu de 0.25, tal que models que tinguin un $SQRr$ inferior a 0.25 es considerin models amb una bona capacitat de predicció.

Resolució del cas pels mètodes usuals

Es pretén fer prediccions d'una variable resposta contínua. Començarem per ajustar un model lineal amb variables predictores funcionals. En aquest cas el coeficient β no té dimensió finita. Ramsay and Silverman (2005) modelitzen

la relació entre la resposta i les covariables funcionals mitjançant l'expressió

$$Y_i = \int_T Z_i(t)\beta(t)dt + e_i,$$

Per ajustar el model amb R, s'utilitzarà la variable funcional Z, així com la seva primera i segona derivada. Mitjançant la funció `fdata.deriv` del paquet `fda.usc` es pot derivar el conjunt de dades funcionals:

```
fdata.deriv(fdataobj,nderiv=1,method="bspline",
            class.out='fdata',nbasis=NULL,...)
```

El model lineal funcional s'ajusta amb la funció `fregre.lm`:

```
fregre.lm(formula,data,basis.x=NULL,basis.b=NULL,...)
```

En l'argument `formula` es descriu el model a ajustar. En l'exemple es pretén modelitzar la relació entre la variable resposta `protein` i les dades funcionals `wheat` (sense derivar, primera i segona derivada) en el conjunt de la mostra d'entrenament amb 80 casos. `data` ha de contenir les dades definides en `formula`. Pel que fa a `basis.x` i `basis.b` determinen les bases per estimar la representació de les dades funcionals Z i el coeficient β respectivament. La generació d'aquestes en l'aplicació `wheat` es defineixen per la funció `create.bspline.basis` tal com es mostra en la Figura 5.4.

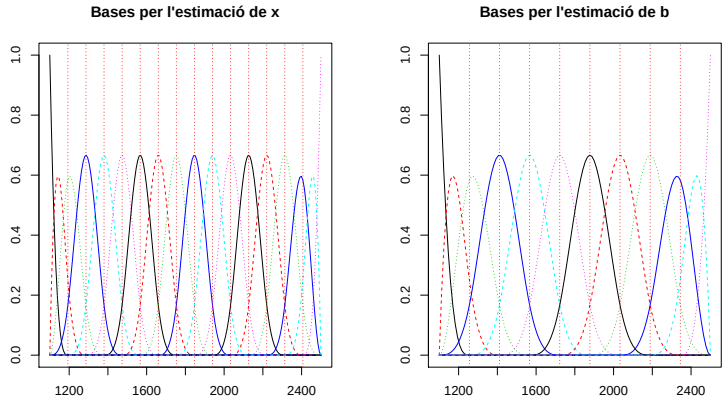

Figura 5.4: Bases de l'estimació de Z i de β per ajustar el model de regressió: 18 bases en Z i 12 en β

Aquí hi ha el codi R del tractament de les dades que s'ha realitzat inicialment:

```
nt <- 80
set.seed(2)
ind<-sample(1:100)[1:nt]

y=protein[ind]
newy=protein[-ind]
dataf=as.data.frame(protein[ind])
names(dataf)="proteinaux"
newdataf=as.data.frame(protein[-ind])

tt=x[["argvals"]]

x.d2=fdata.deriv(x[ind,],nderiv=2,nbasis=18)
newx=x[-ind,]
newx.d2=fdata.deriv(newx,nderiv=2,nbasis=18)

x.d1=fdata.deriv(x[ind,],nderiv=1,nbasis=18)
newx.d1=fdata.deriv(newx,nderiv=1,nbasis=18)

ldata=list("df"=dataf,"x"=x[ind,],"x.d2"=x.d2,"x.d1"=x.d1)
newldata=list("df"=newdataf,"x"=newx,"x.d2"=newx.d2,"x.d1"=newx.d1)

basis1=create.bspline.basis(rangeval=range(tt),nbasis=18)
basis2=create.bspline.basis(rangeval=range(tt),nbasis=18)
basis.x=list("x"=basis1)
basis.b=list("x"=basis2)
```

i s'estima el model de la següent forma

```
> f0 <- proteinaux~x
> res.lm <- fregre.lm(f0,ldata,basis.x=basis.x,basis.b=basis.b)
```

L'R^2 val 0.8749. Fent prediccions de la mostra test

```
> pred.lm <- predict.fregre.lm(res.lm,newldata)
```

el $SQRr$ obtingut per aquest primer ajust és de 0.2503177. Per tant les prediccions ja són força bones. Realitzant el mateix per la primera derivada i la segona:

```
# 1a derivada
f1 <- m1aux~x.d1
res.lm.1 <- fregre.lm(f1,ldata,basis.x=basis.x,basis.b=basis.b)
pred.lm.1 <- predict.fregre.lm(res.lm.1,newldata)

# 2a derivada
f2 <- m1aux~x.d2
res.lm.2 <- fregre.lm(f2,ldata,basis.x=basis.x,basis.b=basis.b)
pred.lm.2 <- predict.fregre.lm(res.lm.2,newldata)
```

els R^2 obtinguts són 0.7681 (primera derivada) i 0.7398 (segona derivada). Els errors de predicció es troben a la Taula 5.1.

fregre.lm	**SQRr**
sense derivar	0.250
primera derivada	0.292
segona derivada	0.250

Taula 5.1: Valor del $SQRr$ de les prediccions per diferents codificacions de les dades: sense derivar, primera derivada (el pitjor) i segona derivada.

Amb aquesta modelització les millors prediccions s'aconsegueixen amb les funcions sense derivar i amb la segona derivada.

Resolució del cas pel `dblm`

El primer pas que s'ha de realitzar és codificar les covariables funcionals com a inter-distàncies al quadrat entre individus. S'utilitza la funció `semimetric.basis` del paquet `fda.usc` (norma L_2). Es calculen les distàncies per les funcions sense derivar, primera i segona derivada:

```
> D2.0 <- semimetric.basis(x,x,nderiv=0,nbasis1=18,nbasis2=18)^2
> D2.1 <- semimetric.basis(x,x,nderiv=1,nbasis1=18,nbasis2=18)^2
> D2.2 <- semimetric.basis(x,x,nderiv=2,nbasis1=18,nbasis2=18)^2
```

La representació de les dades funcionals és donada a partir de la creació de 18 bases per B-splines i `nderiv` derivades. Per ajustar el model, només es guarden les inter-distàncies entre individus al quadrat de la mostra d'entrenament:

```
> D2.0.trai <- D2.0[ind,ind];class(D2.0.trai)<-"D2"
> D2.1.trai <- D2.1[ind,ind];class(D2.1.trai)<-"D2"
> D2.2.trai <- D2.2[ind,ind];class(D2.2.trai)<-"D2"
```

Modelització per les dades funcionals sense derivar

S'ajusta el model `dblmFDA.0` utilitzant la matriu de distàncies al quadrat `D2.0.trai`:

```
> dblmFDA.0 <- dblm(D2.0.trai,y,method="GCV",full_search=TRUE)
```

El mètode per triar la dimensió efectiva és el `GCV`. L'`eff.rank` òptim és de 17 (s'observa en l'últim gràfic de la Figura 5.5). A més s'ha provat pels altres criteris disponibles (`OCV`, `AIC` i `BIC`) i la dimensió òptima obtinguda és la mateixa (17 coordenades euclidianes). El model ajustat `dblmFDA.0` té un R^2 de 0.871309 i un R^2 ajustat de 0.836022 (es poden obtenir en el `summary(dblmFDA.0)`).

Per validar el model es miren els gràfics que proporciona la funció `plot` d'un objecte `dblm`. Es troben en la Figura 5.5. En el primer gràfic es verifica la variança constant en els residus. A més, no s'hi aprecien observacions extremes. Els residus més alts fan referència als individus 25, 71 i 73. En el segon gràfic s'observa normalitat en els residus: no hi ha cap observació que s'allunyi de la recta de normalitat. En el quart gràfic s'hi destaquen les observacions més influents. Aquestes són la 25, 65 i 71. Sobretot destaca la 71 amb una distància de Cook força més elevada a la resta. No obstant, no es troba fora de les franges vermelles del cinquè dels gràfics, que indicaria una influència excessiva de tal observació en el model. El model, per tant, queda validat.

Modelització per a la primera derivada de les dades funcionals
S'ajusta el model `dblmFDA.1` utilitzant la matriu de distàncies al quadrat `D2.1.trai`:

```
> dblmFDA.1 <- dblm(D2.1.trai,y,method="GCV",full_search=TRUE)
```

Pel mètode `GCV` la dimensió efectiva òptima és d'11 (s'observa en l'últim gràfic de la Figura 5.6). Pels altres criteris disponibles la dimensió obtinguda és la mateixa (11). El model ajustat `dblmFDA.1` té un R^2 de 0.866472 i un R^2 ajustat de 0.84487.

Mirant els gràfics de la Figura 5.6 es pot validar el model. En aquest ajust, hi ha una observació (la 71) que és molt influent en el càlcul del model. S'hauria d'analitzar el seu motiu, i, si s'escau, treure-la de l'ajust. Pel què fa a les condicions de variança constant, independència i normalitats dels residus, queden de nou validades.

Modelització per a la segona derivada de les dades funcionals
S'ajusta el model `dblmFDA.2` utilitzant la matriu de distàncies al quadrat `D2.2.trai`:

```
> dblmFDA.2 <- dblm(D2.2.trai,y,method="GCV",full_search=TRUE)
```

Pel mètode `GCV` la dimensió efectiva és de 13 (s'observa en l'últim gràfic de la Figura 5.7). El model ajustat `dblmFDA.2` té un R^2 de 0.859047 i un R^2 ajustat de 0.831284, més baixos respecte la primera derivada.

En la Figura 5.7 hi ha els gràfics corresponents per validar el model. De nou s'observa com l'observació 71 és molt influent en el càlcul del model. La variança constant, independència i normalitat dels residus queden validades.

Prediccions pels models `dblm`
Un cop validats els models es pretén avaluar la seva capacitat de predicció. S'han guardat 20 observacions, que conformen la mostra test, per tal de quantificar quin dels models és el que fa les millor prediccions de la resposta (proteïna) per a les noves dades. Per mitjà de les següents comandes s'estimen les prediccions pels tres models validats:

```
> pred.db.glm.0 <- predict(dbglmFDA.0,D2.0[-ind,ind],type_var="D2",
                   type="response")
> pred.db.glm.1 <- predict(dbglmFDA.1,D2.1[-ind,ind],type_var="D2",
                   type="response")
> pred.db.glm.2 <- predict(dbglmFDA.2,D2.2[-ind,ind],type_var="D2",
                   type="response")
```

Els errors de predicció, com pel cas usual, es quantifiquen per la mesura $SQRr$. Es troben en la taula 5.2.

dblm	**SQRr**
sense derivar	0.269
primera derivada	0.207
segona derivada	0.232

Taula 5.2: Valor del $SQRr$ de les prediccions dels models `dblm` per diferents codificacions de les dades: sense derivar, primera derivada (el que dóna millors resultats) i segona derivada.

Per la via distance-based els errors de predicció són relativament similars als obtinguts per la funció `fregre.lm`. Tot i això, contràriament al cas lineal usual, és utilitzant la informació de les primeres derivades quan la capacitat de predicció en els models `dblm` és més gran. De fet, quan s'analitzava el gràfic

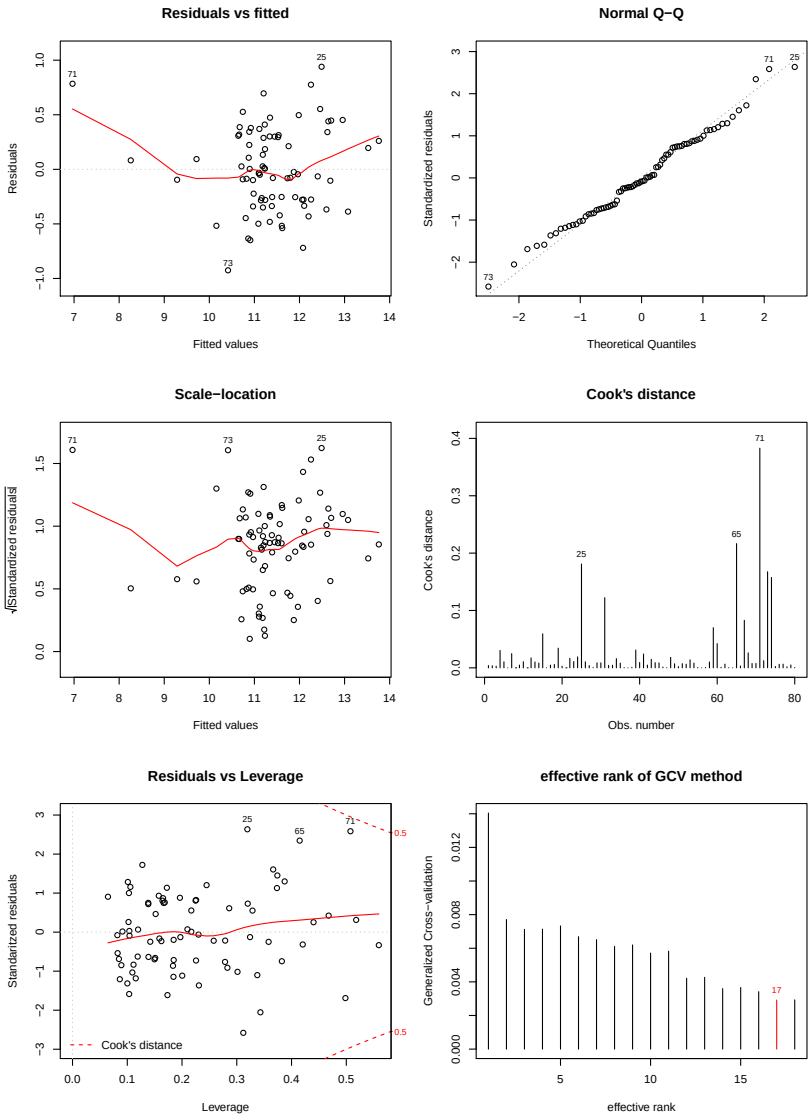

Figura 5.5: Bateria de gràfics per a la validació del model `dblmFDA.0`. El model quedaria validat: variança constant, independència i normalitat en els residus. El rang efectiu òptim pel criteri GCV és de 17.

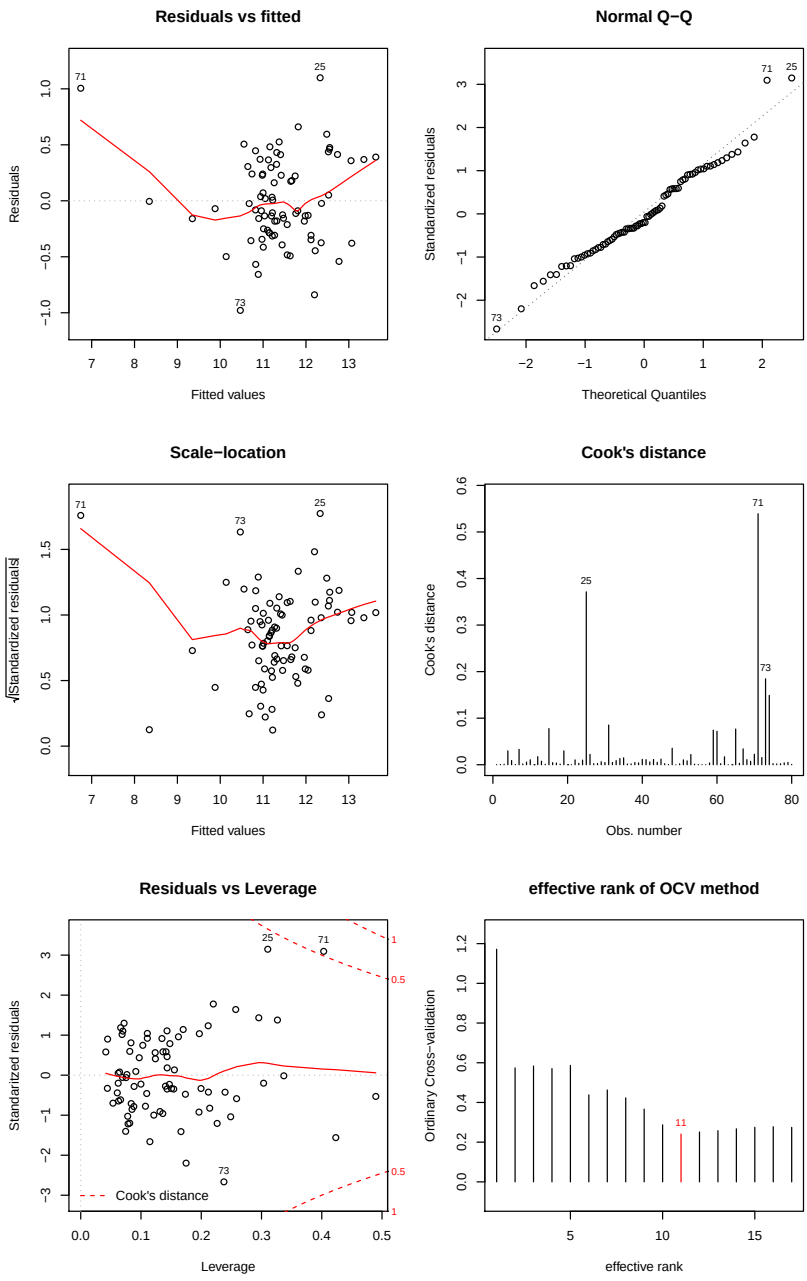

Figura 5.6: Bateria de gràfics per a la validació del model `dblmFDA.1`. El model quedaria validat encara que s'hauria d'estudiar si treure o no l'observació 71 del model. El rang efectiu òptim pel criteri GCV és d'11.

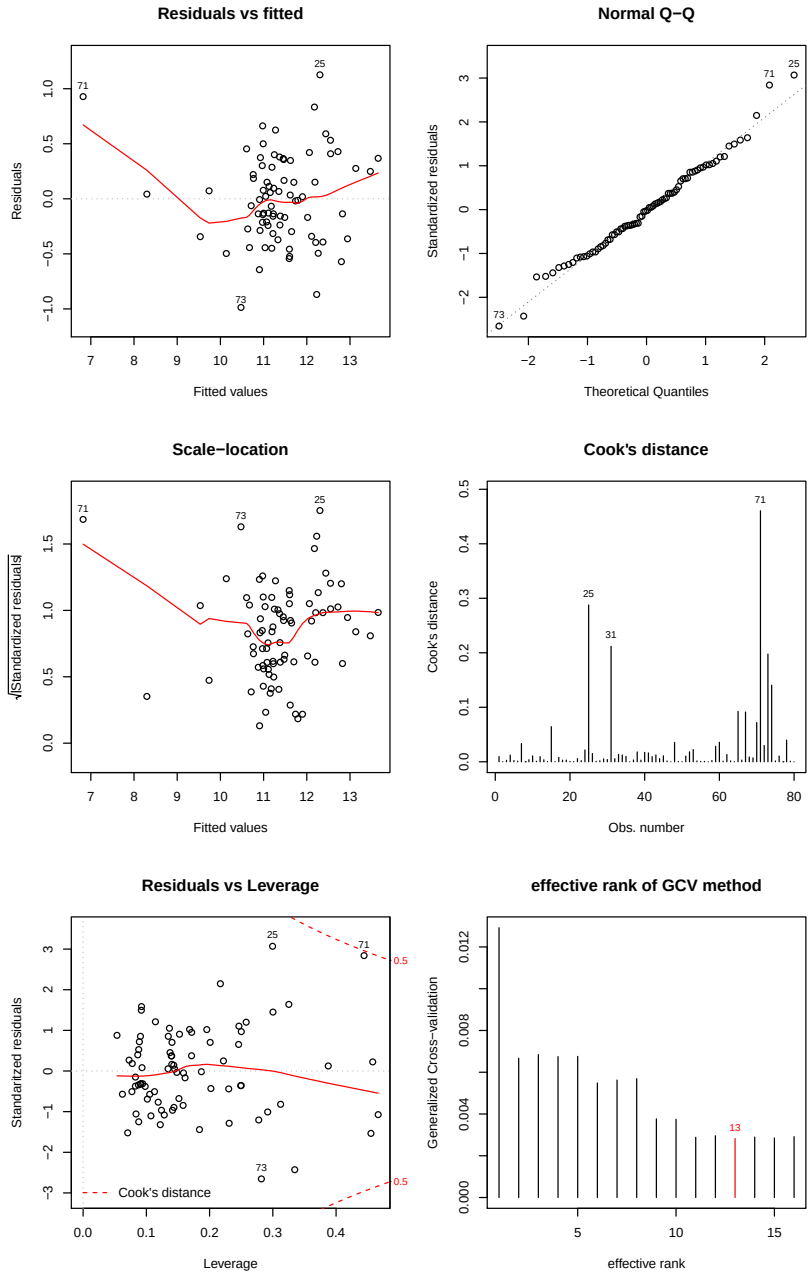

Figura 5.7: Bateria de gràfics per a la validació del model `dblmFDA.2`. L'observació 71 té una influència en el model força elevada, s'hauria de mirar bé si és bo estimar el model sense tal punt. Es valida independència, variança constant i normalitat en els residus. El rang efectiu òptim pel criteri GCV és de 13.

de correlacions 5.3, ja s'havia comentat que era amb la primera derivada on les correlacions puntuals eren més elevades. S'ha confirmat aquesta hipòtesi. Tant en bondat de l'ajust (el que té l'R^2 ajustat més elevat) com en capacitat de predicció (el que té el $SQRr$ més baix) el model `dblmFDA.1` és el millor dels models lineals basats en distàncies.

Resolució del cas pel `ldblm`

Amb les mateixes matrius de distàncies, `dbstats` permet ajustar la versió local del model lineal definit pel conjunt de dades `wheat`.

Modelització per les dades funcionals sense derivar
El primer ajust es realitza amb la matriu de distàncies calculada a partir de les corbes inicials. Tant pel càlcul dels pesos Kernel (`Epanechnikov`) com pels ajustos local (`dblm`) s'utilitza la mateixa semi-mètrica (`D2.0.trai`). L'argument `rel.gvar`, fixat a 0.9, indica que cada ajust lineal local prendrà el rang efectiu tal que com a mínim s'expliqui el 90% de variabilitat geomètrica de les dades. El mètode d'elecció del paràmetre de suavitzat és el `"GCV"` avaluat en 10 amples de banda en un rang de valors entre 1 i 4. La crida és la següent:

```
> ldblmFDA.0 <- ldblm(D2.0.trai,y=y,method="GCV",h.range=c(1,4),rel.gvar=0.9)
```

L'ample de banda òptim val 2.16 (s'observa el comportament del criteri GCV en la Figura 5.8). L'R^2 obtingut és de 0.606.

Modelització per a la primera derivada de les dades funcionals
Fent ús de la primera diferenciació de les dades s'ajusta el model `ldblmFDA.1`

```
> ldblmFDA.1<- ldblm(D2.1.trai,y=y,method="GCV",rel.gvar=0.9)
```

L'ample de banda òptim pel criteri `GCV` és de 0.0029. Aquest no és comparable al trobat en `ldbglmFDA.0` per estar expressats en escales diferents: distàncies entre funcions o distàncies entre les derivades de les funcions. En la Figura 5.9 hi ha el gràfic que mostra el comportament del criteri GCV per les h avaluades en el model `ldblmFDA.1`. L'R^2 és de 0.901, més elevada que en el `ldblmFDA.0`.

Modelització per a la segona derivada de les dades funcionals
Si es pren la segona derivada, el model ajustat s'obté amb R així:

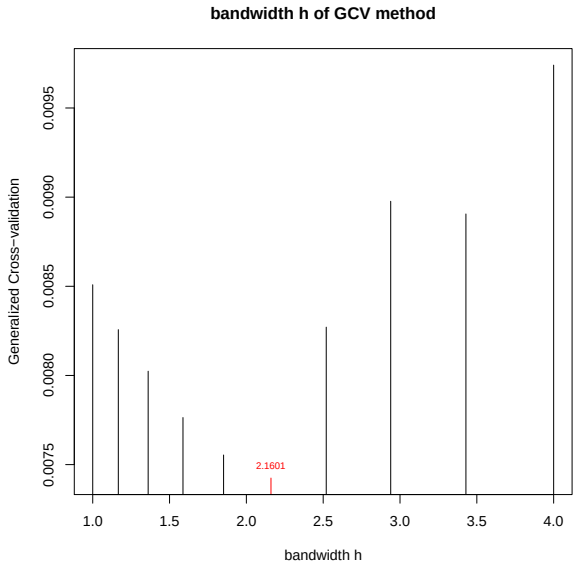

Figura 5.8: Criteri `GCV` per l'elecció del bandwidth òptim en l'estimació del model `ldbglmFDA.0`: la h que minimitza el `GCV` val 2.160.

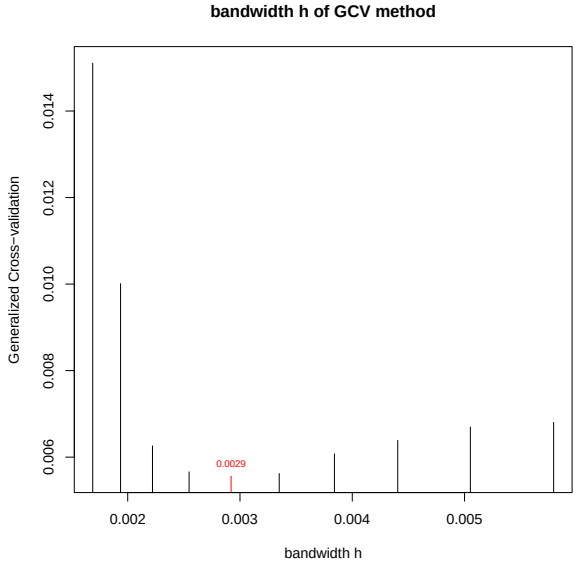

Figura 5.9: Criteri `GCV` per l'elecció del bandwidth òptim en l'estimació del model `ldbglmFDA.1`: la h que minimitza el `GCV` val 0.0029.

```
> ldblmFDA.2 <- ldblm(D2.2.trai,y=y,method="GCV",rel.gvar=0.9,
                        h.range=c(1e-4,4e-4))
```

L'ample de banda òptim és de 0.00018. En la Figura 5.10 hi ha el gràfic que mostra el comportament del criteri GCV per les h avaluades en el model `ldblmFDA.2`. L'R^2 és de 0.681, inferior al corresponent al `ldblmFDA.1` .

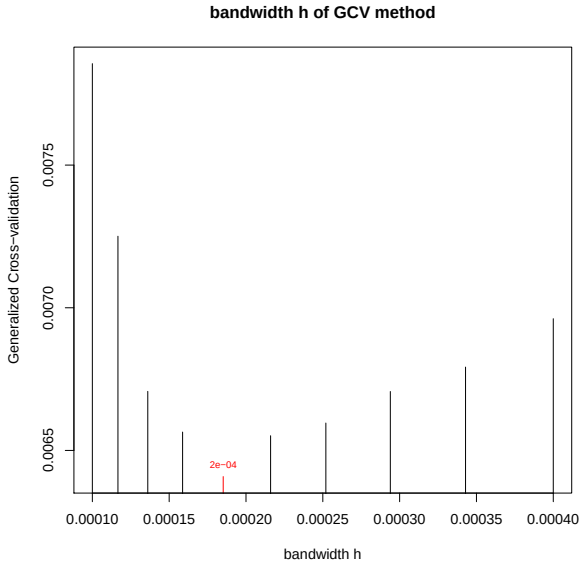

Figura 5.10: Criteri `GCV` per l'elecció del bandwidth òptim en l'estimació del model `ldbglmFDA.2`: la h que minimitza el `GCV` val 0.00018.

Modelització per a la combinació de dos distàncies

Un últim model que es vol ajustar combina la informació de la primera derivada i de les dades originals. Amb les distàncies de la primera derivada es defineixen els pesos amb una funció nucli i amb les distàncies de les dades originals es realitzen els ajustos locals. La crida en R és la següent:

```
> ldblmFDA.10 <- ldblm(D2.1.trai,D2.0.trai,y,method="GCV",rel.gvar=0.9)
```

L'ample de banda òptim és de 0.0025. En la Figura 5.11 hi ha el gràfic que mostra el comportament del criteri GCV per les h avaluades en el model

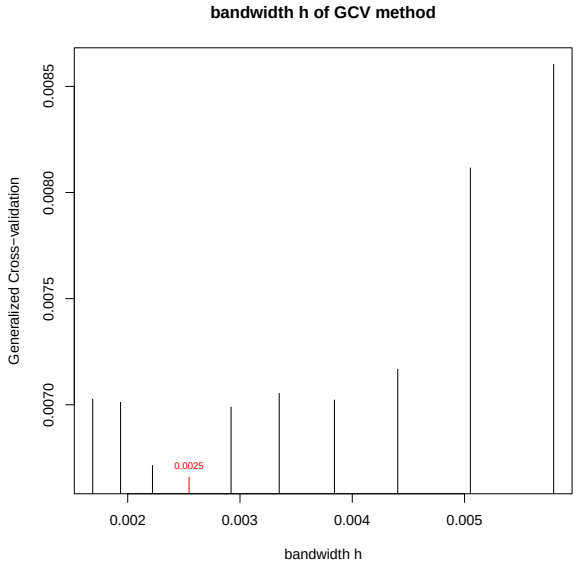

Figura 5.11: Criteri `GCV` per l'elecció del bandwidth òptim en l'estimació del model `ldbglmFDA.10`: la h que minimitza el `GCV` val 0.0025.

`ldblmFDA.10`. L'R^2 és de 0.853, inferior al corresponent a la primera derivada.

Prediccions pels models `ldblm`

Les prediccions dels quatre models estimats es troben amb les comandes:

```
> pred.ldb.glm.0 <- predict(ldbglmFDA.0,D2.0[-ind,ind],type_var="D2",
                    type="response")$fit
> pred.ldb.glm.1 <- predict(ldbglmFDA.1,D2.1[-ind,ind],type_var="D2",
                    type="response")$fit
> pred.ldb.glm.2 <- predict(ldbglmFDA.2,D2.2[-ind,ind],type_var="D2",
                    type="response")$fit
> pred.ldb.glm.12 <- predict(ldbglmFDA.10,D2.1[-ind,ind],D2.0[-ind,ind],
                    type_var="D2",type="response")$fit
```

i els errors de predicció ($SQRr$) es mostren en la Taula 5.3. Els models locals tenen una capacitat de predicció molt similar. En destaca sobretot l'últim dels models ajustats: el `dblmFDA.10` que amb un $SQRr$ de 0.0742 és el model amb un error de predicció menor. A més té l'R^2 certament alt (de 0.853). Aquest és un cas on combinar la informació de les matrius de distàncies (distàncies entre primera derivada per estimar els pesos i distàncies entre observacions originals pels `dblm` local) pot proporcionar bons resultat en l'estimació d'un model.

ldblm	SQRr
sense derivar	0.1418375
primera derivada	0.1304473
segona derivada	0.1927517
priemra/zero derivada	0.0741718

Taula 5.3: Valor del *SQRr* de les prediccions pels models `ldblm` per diferents codificacions de les dades: sense derivar, primera derivada, segona derivada i la combinació de la primera derivada i les corbes inicials (el que dóna millors resultats).

Resolució del cas pel `dbplsr`

L'última metodologia basada en distàncies que es pot emprar en aquest exemple és el `dbplsr`. Es modelitza, com amb les altres metodologies, la relació entre la resposta (`protein`) i les covariables funcionals (sense derivar, primera i segona derivada) transformades en matrius de distàncies al quadrat. S'avalua per a cada un dels models ajustats la capacitat de predicció.

Modelització per les dades funcionals sense derivar
L'ajust d'un `dbplsr` per la matriu de distàncies calculada a partir de les dades originals es realitza amb la següent comanda:

```
dbplsFDA.0 <- dbplsr(D2.0.trai,y,method="GCV",ncomp=20)
```

El model `dbplsFDA.0` té 20 components i mitjançant el criteri `GCV` es decideix quantes d'aquestes són necessàries. Es un cas complicat, mirant la Figura 5.12 s'aprecia que el `GCV` disminueix a mesura que es van afegint components al model. No obstant, sembla que a partir de la vuitena component el `GCV` s'estabilitza i aleshores les diferències en el valor del criteri esdevenen insignificants. Consegüentment, s'ajusta el model amb 8 components:

```
> dbplsFDA.0 <- dbplsr(D2.0.trai,y,method="ncomp",ncomp=8)
```

Realitzant el `summary` del `dbplsFDA.0` s'obtenen les següents caracteritzacions de la variança explicada del model:

```
% variance explained:
      1comp  2comp  3comp  4comp  5comp  6comp  7comp  8comp
R2     7.09  48.57  54.34  55.22  66.78  68.02  73.50  80.60
adjR2  5.90  47.24  52.54  52.84  64.54  65.40  71.03  78.41
```

```
gvar  93.68  99.63  99.84  99.97  99.97  99.99      1      1
```

La segona component és la que té un impacte més gran en l'R^2. Amb les 8 components l'R^2 és de 0.806, una mica inferior al que es tenia amb la modelització `dblm` (`dblmFDA.0`). A més, l'R^2 ajustat val 0.784. La variabilitat geomètrica explicada a partir de la setena component ja és del 100%. Per tant, no es perd informació de les dades inicials.

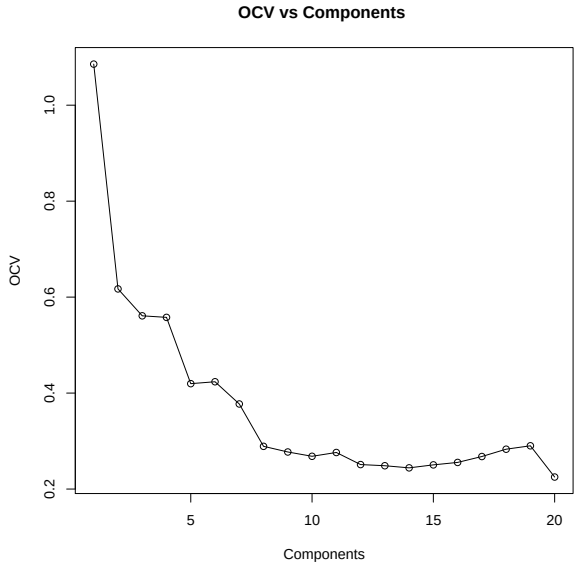

Figura 5.12: Criteri `GCV` per triar el nombre de components òptim en l'estimació del model `dbplsFDA.0`: amb 8 components el criteri `GCV` ja s'estabilitza.

Modelització per a la primera derivada de les dades funcionals

Utilitzant la primera derivada s'ajusta el model `dbplsFDA.1` amb la comanda:

```
> dbplsFDA.1 <- dbplsr(D2.1.trai,y,method="GCV",ncomp=18)
```

El nombre de components òptim és 8 (s'observa en la Figura 5.13). Per tant, una altra vegada es prenen 8 components per ajustar el model, en aquest cas el `dbplsFDA.1`:

```
> dbplsFDA.1 <- dbplsr(D2.1.trai,y,method="ncomp",ncomp=8)
```

Realitzant el `summary` del `dbplsFDA.1` s'obtenen les següents expressions per establir la bondat del ajust:

```
% variance explained:
      1 comp 2 comp 3 comp 4 comp 5 comp 6 comp 7 comp 8 comp
R2     52.87  54.13  56.17  58.81  77.10  80.43  85.40  86.66
adjR2  52.26  52.94  54.44  56.61  75.55  78.81  83.98  85.15
gvar   29.69  71.83  95.90  99.20  99.49  99.72  99.81  99.84
```

La primera component, contràriament al cas anterior, és la que té un impacte més gran en l'R^2. Amb les 8 components, l'R^2 és de 0.867, similar al valor en la modelització `dblm` (`dblmFDA.1`). L'R^2 ajustat amb 8 components val 0.852. La variabilitat geomètrica explicada a partir de la quarta s'estabilitza al 99% i creix molt lentament en les posteriors 4 components (amb 8 és del 99.84%).

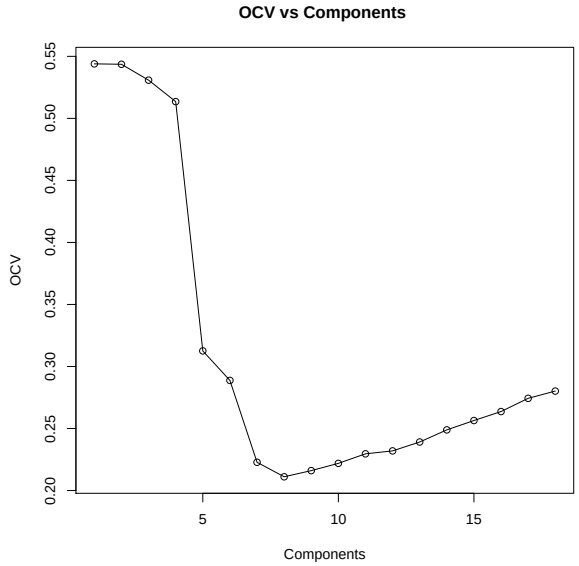

Figura 5.13: Criteri `GCV` per triar el nombre de components òptim en l'estimació del model `dbplsFDA.1`: amb 8 components el criteri `GCV` és mínim.

Modelització per a la segona derivada de les dades funcionals

L'última codificació de les dades que s'ha calculat és la matriu de distàncies entre les segones derivades de les funcions inicials. El model `dbplsFDA.2` s'ajusta:

```
> dbplsFDA.2 <- dbplsr(D2.2.trai,y,method="GCV",ncomp=20)
```

El nombre de components òptim és 9 (s'observa en la Figura 5.14). S'ajusta, consegüentment, el model amb 9 components:

```
> dbplsFDA.2 <- dbplsr(D2.2.trai,y,method="ncomp",ncomp=9)
```

El `summary` del `dbplsFDA.2` proporciona les següents expressions que quantifiquen la variança explicada del model:

```
% variance explained:
      1 comp 2 comp 3 comp 4 comp 5 comp 6 comp 7 comp 8 comp 9 comp
R2     53.58  55.18  58.62  61.65  73.34  78.86  82.13  84.14  85.19
adjR2  52.99  54.02  56.98  59.60  71.54  77.12  80.39  82.36  83.28
gvar   44.61  95.13  96.39  99.17  99.79  99.86  99.91  99.93  99.96
```

La primera component, de la mateixa manera que en el `dbplsFDA.1`, és la que té un impacte més gran en l'R^2. Amb les 9 components l'R^2 és de 0.852 i l'R^2 ajustat val 0.833. La variabilitat geomètrica explicada a partir de la quarta s'estabilitza al 99% i creix molt lentament en les posteriors 5 components (amb 9 és del 99.96%).

Predicció per a la mostra test
Les prediccions pels tres models ajustats es troben amb les següents comandes:

```
> pred.dbpls.0 <- predict(dbplsFDA.0,D2.0[-ind,ind],type="D2")
> pred.dbpls.1 <- predict(dbplsFDA.1,D2.1[-ind,ind],type="D2")
> pred.dbpls.2 <- predict(dbplsFDA.2,D2.2[-ind,ind],type="D2")
```

Utilitzant la mateix mesura per quantifica l'error (el $SQRr$), en la Taula 5.4 es troben els $SQRr$ pels tres models `dbplsr` ajustats.

Amb la primera derivada, una altra vegada, és quan s'aconsegueix un error de predicció menor. Tot i això, les diferències amb la segona són inapreciables. Els dos models tenen una capacitat de predicció molt bona.

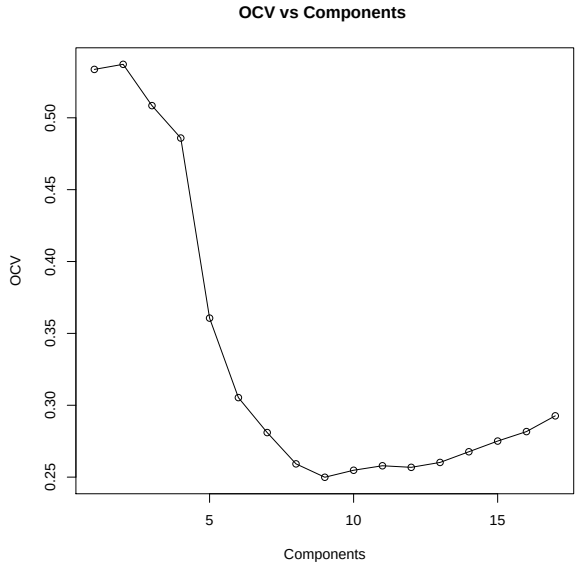

Figura 5.14: Criteri `GCV` per triar el nombre de components òptim en l'estimació del model `dbplsFDA.2`: amb 9 components el criteri `GCV` és mínim.

dbplsr	**SQRr**
sense derivar	0.315613
primera derivada	0.197662
segona derivada	0.207374

Taula 5.4: Valor del *SQRr* de les prediccions pels models `dbplsr` per diferents codificacions de les dades: sense derivar, primera derivada (el que dóna millors resultats) i segona derivada.

Comparativa entre totes les modelitzacions

Hi ha dos aspectes que cal considerar a l'hora de triar un model. El primer és com de bo és l'ajust de les dades utilitzades per a la modelització. L'R^2 quantifica aquest aspecte: la bondat de l'ajust. En la Taula 5.5 es destaquen els valors de l'R^2 per a tots els models basats en distàncies que s'han ajustat al llarg de l'exemple. Tant els ajustos `dblm` com els `dbplsr` tenen l'R^2 molt elevat, entre 0.8 i 0.9

El segon aspecte per triar entre un o altre model és la capacitat de predicció. L'objectiu és realitzar prediccions per futures dades, per tant mesurar quina capacitat té cada model per predir nous punts de forma precisa té una

Bondat de l'ajust: R^2	**fregre.lm**	**dblm**	**ldblm**	**dbpls**
sense derivar	0.8749	0.8713	0.6060	0.8060
primera derivada	0.7681	0.8449	0.9010	0.8666
segona derivada	0.7398	0.8590	0.6810	0.8519
primera/zero derivada	-	-	0.8530	-

Taula 5.5: Representació de la bondat de l'ajust dels models de regressió basats en distàncies de les dades `Wheat`: l'R^2. El model amb un R^2 major és el `ldblmFDA.1` que utilitza la primera derivada per estimar els pesos Kernel i els ajustos locals. Entre el `dblm` i el `dbpls` no s'hi destaquen diferències importants i els R^2 són comparables.

importància fundamental en l'elecció del model. És més important que l'R^2 de fet, ja que models amb R^2 alts no indiquen de forma segura que l'error de predicció sigui més baix. La comparativa entre els errors de predicció per les tres metodologies emprades utilitzant la mesura $SQRr$ es troba en la Taula 5.6.

Capacitat predicció: $SQRr$	**fregre.lm**	**dblm**	**ldblm**	**dbplsr**
sense derivar	0.250	0.269	0.142	0.316
primera derivada	0.292	0.207	0.130	0.198
segona derivada	0.250	0.232	0.193	0.207
primera/zero derivada	-	-	0.074	-

Taula 5.6: Avaluació de la capacitat de predicció pels diferents models ajustats al llarg de l'exemple. Els models local, en pràcticament tots els casos, proporcionen un menor valor del $SQRr$ fent èmfasi en la gran potència que tenen per realitzar prediccions. Els `dbplsr` donen unes bones prediccions i són una alternativa als models lineals clàssics.

Val la pena recalcar la potència dels mecanismes no paramètrics en la predicció. En tots els tractaments realitzats a les dades `Wheat`: sense derivar, primera derivada i segona derivada, l'ajust local per `ldblm` millora l'obtingut pel LM paramètric o el LM basat en distàncies. A més, és en la combinació de dues maneres de codificar les dades (1era derivada i dades originals) quan les prediccions són més bones: amb un $SQRr$ de 0.074 aconsegueix explicar pràcticament tota la variabilitat de l'error en la resposta per a les noves dades.

5.2 `motorins`: aplicació a assegurances d'automòbils

Les dades que es tracten en aquest exemple descriuen el comportament de les assegurances d'automòbils amb pòlissa a tercers a Suècia l'any 1977. A Suècia, totes les companyies d'assegurances d'automòbils apliquen els mateixos arguments de risc per classificar als clients. Per tant, les seves carteres (portfolios) i les seves estadístiques de reclamacions (claims statistics) es poden tractar globalment. Les dades van ser donades al comitè suec: Swedish Committee on the Analysis of Risk Premium in Motor Insurance, per tal de ser analitzada la influència real de l'argument de risc respecte les reclamacions i comparar aquesta estructura amb la tarifa corresponent.

Aquest conjunt de dades s'estudia en Hallin and Ingenbleek (1983) i està inclòs en la llibreria d'R Faraway (2012) amb el nom `motorins`. `data(motorins)` és un *data frame* amb 1797 observacions que conté les següents 8 variables:

- `Kilometres`: factor ordinal que representa els quilòmetres per any de l'automòbil:
  1: $<$ 1000, 2: 1000-15000, 3: 15000-20000, 4: 20000-25000, 5: $>$ 25000
- `Zone`: factor que representa la zona geogràfica de Suècia al qual pertany l'automòbil: 1: Estocolm, Göteborg i Malmö; 2: Altres grans ciutats i rodalies; 3: Petites ciutats i rodalies del sud de Suècia; 4: Àrees rurals al sud de Suècia; 5: Petites ciutats i rodalies del nord de Suècia; 6: Àrees rurals al nord de Suècia; 7: La illa de Gotland.
- `Bonus`: sense prima per reclamació. Es igual al número d'anys, més u, des de l'última reclamació.
- `Make`: factor que diferencia entre vuit models de cotxes diferents. Hi ha un novè nivell que engloba totes les altres classes d'automòbils.
- `Insured`: número d'assegurances total el primer any que l'automòbil té pòlissa d'assegurança.
- `Claims`: número de reclamacions. Es defineix reclamació, en l'àmbit de les assegurances, com una sol·licitud formal destinada a una companyia asseguradora demanant un pagament basat en els termes de la pòlissa d'assegurança corresponent.
- `Payment`: Valor total dels pagaments amb moneda: corones sueques (Skr).

- `perd`: taxa de pagaments per reclamació.

Es tractaràn únicament les observacions de la primera zona descrita en `Zone`. Hi ha 295 casos corresponents a diferents grups de risc en les ciutats d'Estocolm, Göteborg i Malmö. La variable resposta Y que es pretén modelitzar és el número de reclamacions donades en cada automòbil assegurat ponderats segons el nombre d'assegurats en els anys de pòlissa d'assegurança `Insured` (aquest determina els pesos ω). Hi ha tres factors que es consideren importants i que entraran al model: `Distance`, `Bonus` i `Make`. Els dos primers es consideren predictors numèrics i continus. `Distance` és codificada numèricament a partir de la mitjana per a cada un del 5 grups de la variable `Kilometres`:

`Kilometres==1` (<1000 Km per any):	750 quilòmetres per any.
`Kilometres==2` (1000-15000 Km per any):	8000 quilòmetres per any.
`Kilometres==3` (15000-20000 Km per any):	17500 quilòmetres per any.
`Kilometres==4` (20000-25000 Km per any):	22500 quilòmetres per any.
`Kilometres==5` (>25000 Km per any):	40000 quilòmetres per any.

`Bonus` és codificat numèricament d'1 fins un màxim de 7 anys (número d'anys + 1 des de l'última reclamació). Pel que fa referència a la tercera variable del model, `Make`, es considera que és una variable categòrica numèrica amb nou nivells. El tractament de les dades que s'ha realitzat és el següent:

```
require(faraway)
data(motorins)
Motor1 <- subset(motorins, Zone == 1)
Motor1$frequency <- Motor1$Claims / Motor1$Insured
y <- Motor1$frequency
w <- Motor1$Insured
Motor1$KmC <- rep(0,nrow(Motor1))
Motor1$KmC[Motor1$Kilometres == "1"] <- 750
Motor1$KmC[Motor1$Kilometres == "2"] <- 8000
Motor1$KmC[Motor1$Kilometres == "3"] <- 17500
Motor1$KmC[Motor1$Kilometres == "4"] <- 22500
Motor1$KmC[Motor1$Kilometres == "5"] <- 40000
Motor1$BonC <- as.numeric(Motor1$Bonus)
Motor1$MakeC <- as.numeric(Motor1$Make)
```

Es pretén modelitzar la Y a partir dels tres predictors indicats i els pesos ω definits per la variable `Insured` mitjançant un Model Lineal Generalitzat de la família Poisson.

Descriptiva del conjunt de dades a tractar

Primer de tot, es realitza un anàlisi descriptiu de les variables que entraran al model. Un primer aspecte és comparar el número d'observacions en cada grup per a les 3 covariables predictores:

```
> tapply(y,Motor1$KmC,length)
  750  8000 17500 22500 40000
   61    63    60    57    54

> tapply(y,Motor1$BonC,length)
   1   2   3   4   5   6   7
  42  41  43  38  43  43  45

> tapply(y,Motor1$MakeC,length)
   1   2   3   4   5   6   7   8   9
  35  35  34  27  33  35  32  29  35
```

Hi ha un nombre considerable d'observacions en tots els grups i per a cada una de les variables. A més, no hi ha gaires diferències de longituds entre grups. Per tant, les estimacions en termes de mitjana o variança a dins de cada grup seran raonablement fiables.

En la Figura 5.15 es presenten els gràfics boxplot que creuen la resposta amb les 3 variables predictores. Són visibles les diferències entre grups en cada una de les variables. En la variable `Distance` l'últim dels nivells (el que té més quilometratge) pren valors en la resposta més elevats. En la variable `Bonus`, en canvi, és en els nivells més baixos quan el risc de reclamacions augmenta. L'última covariable, `Make`, no té sentit veure si hi ha una tendència al llarg dels grups. Els nivells són etiquetes i no té importància l'ordre que estan col·locats. No obstant, si que es mostren diferències en la resposta pels diferents nivells. Pel que fa a la detecció d'*outiliers*, s'hi observa algun valor que s'allunya dels valors esperats, però pot ser degut a l'aleatorietat del procés (no hi ha cap *outlier* que s'hagi d'eliminar per ajustar el model).

No es contempla cap interacció entre les 3 covariables per ajustar el model. En la Figura 5.16 hi ha els gràfics referents a les tres interaccions entre parelles de variables que es poden computar. Visualment no s'hi aprecien gaires diferències i es suposa que no té sentit incloure-les al model.

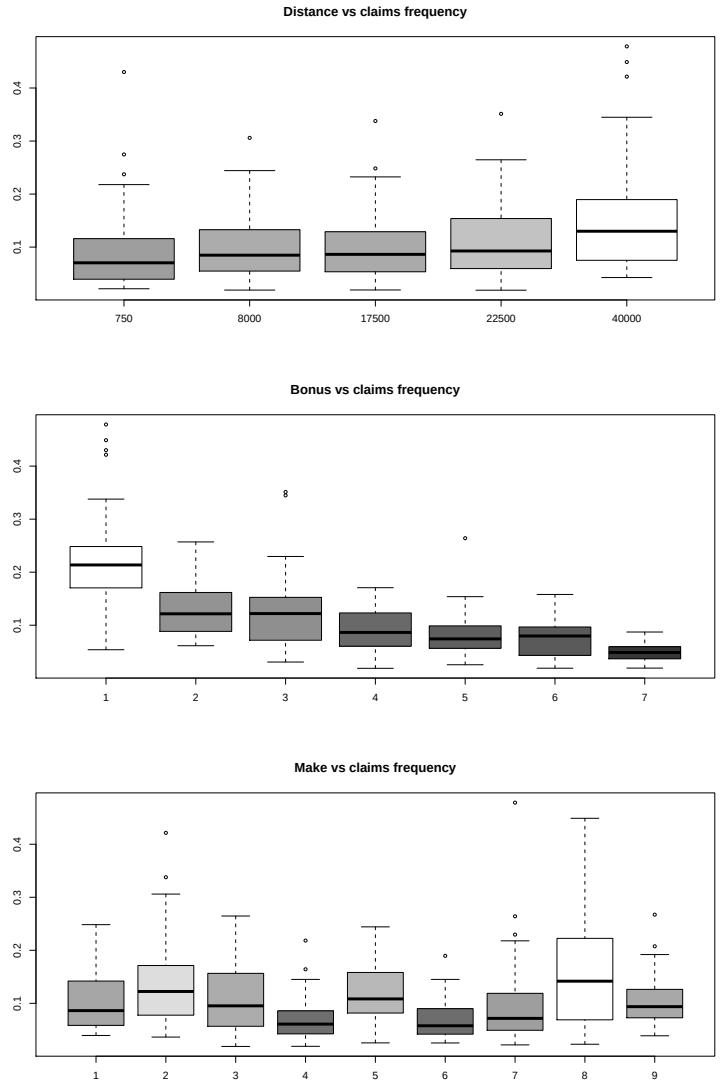

Figura 5.15: Gràfics boxplots entre les variables predictores i la resposta. S'aprecien diferències entre grups en totes les variables. Per tant té sentit que entrin en el model per fer prediccions de la resposta.

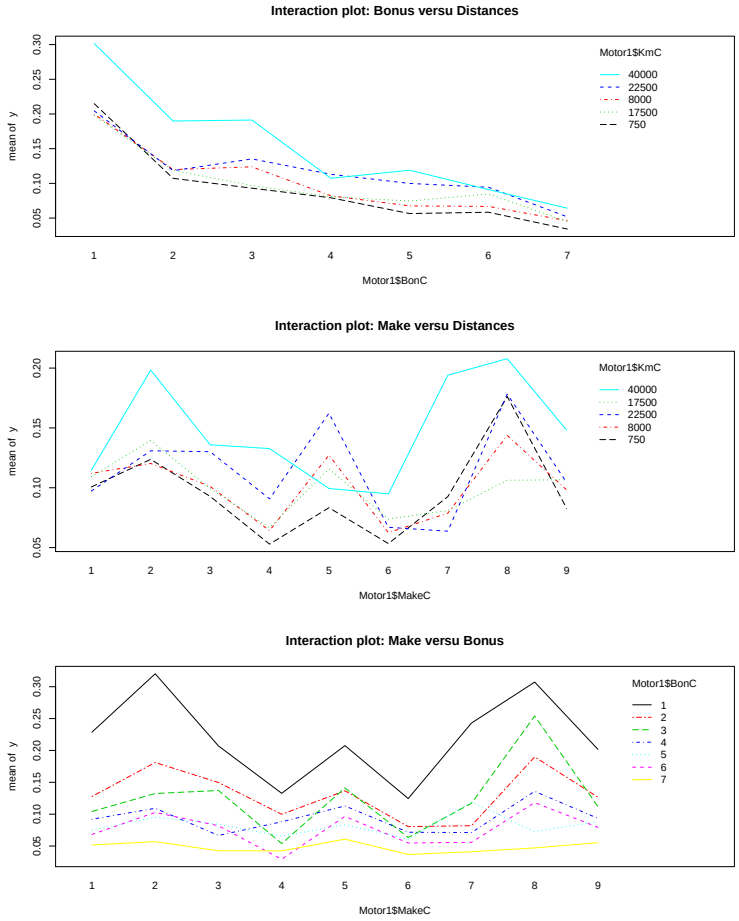

Figura 5.16: Gràfics d'interaccions dos a dos entre les variables predictores i la resposta. No s'hi destaquen grans diferències. És a dir, el comportament de la resposta al llarg dels grups de cada variable no sembla que depengui de les altres variables.

Ajust d'un Model Lineal Generalitzat amb la funció `glm`

Els mètodes basats en distàncies són un complement als mètodes usuals. Es tracten de tècniques de predicció, així que per determinar quines variables són necessàries en el model per fer prediccions de la resposta és vàlid ajudar-se dels mètodes ordinaris, en aquest cas el GLM.

El model `glm1` de la família Poisson i funció d'enllaç `"log"` s'ajusta amb R de la següent manera:

```
> glm1 <- glm(y ~ KmC + BonC + factor(MakeC), data=Motor1,
              family =poisson (link = "log"),weights = w)
```

i el seu `summary` permet visualitzar la significació de cada variable

```
> summary(glm1)

Call:
glm(formula = y ~ KmC + BonC + factor(MakeC), family = poisson(link = "log"),
    data = Motor1, weights = w)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.5134  -0.8980  -0.0643   0.8076  10.0902

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -1.640e+00  2.637e-02 -62.181  < 2e-16 ***
KmC             1.431e-05  6.381e-07  22.424  < 2e-16 ***
BonC           -2.165e-01  2.762e-03 -78.387  < 2e-16 ***
factor(MakeC)2  1.282e-01  4.598e-02   2.788  0.00531 **
factor(MakeC)3 -2.140e-01  5.162e-02  -4.146 3.38e-05 ***
factor(MakeC)4 -5.162e-01  4.987e-02 -10.352  < 2e-16 ***
factor(MakeC)5  1.270e-01  4.850e-02   2.618  0.00883 **
factor(MakeC)6 -3.976e-01  4.467e-02  -8.900  < 2e-16 ***
factor(MakeC)7 -1.320e-01  5.891e-02  -2.240  0.02508 *
factor(MakeC)8  1.396e-01  8.673e-02   1.609  0.10762
factor(MakeC)9 -3.079e-02  2.276e-02  -1.353  0.17618
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 6977.53  on 294  degrees of freedom
Residual deviance:  779.36  on 284  degrees of freedom
AIC: Inf
Number of Fisher Scoring iterations: 5
```

Les tres variables són significatives amb uns p-valors molt baixos.

Ajust d'un GLM basat en distàncies amb la funció `dbglm`

Utilitzant les mateixes variables es realitzen diversos ajustos DBGLM. Canviant la mètrica pel càlcul de les distàncies, així com modificant la manera de fixar el rang efectiu en cada iteració DBLM, es poden computar models alternatius al `glm1`.

El primer model que s'ajusta pretén demostrar que el DBGLM és una extensió del mètode clàssic GLM. Utilitzant la mètrica euclidiana per codificar els regressors com una matriu de distàncies entre individus i aplicant la funció del `dbstats`: `dbglm`, el model basat en distàncies i el clàssic són els mateixos:

```
> dbglm1 <- dbglm(y ~ KmC + BonC + factor(MakeC), data=Motor1,
                family = poisson (link = "log"),weights = w,rel.gvar=1)
> max(dbglm1$fitt-glm1$fitt)
[1] 1.524401e-08
```

La diferència més gran en els valors ajustats es troba en el vuitè decimal, pràcticament negligible.

No obstant, aquest és un cas on hi ha una variable categòrica en el model. Ja s'ha vist en l'apartat 4.2 que per un conjunt de dades amb variables predictores contínues, categòriques i binàries una de les mètriques més populars i utilitzada és la mètrica de Gower. Únicament especificant que l'argument `metric` és igual a `"gower"` s'ajusta el `dbglm2`

```
> dbglm2 <- dbglm(y ~ KmC + BonC + factor(MakeC), data=Motor1,
                  metric="gower",family = poisson (link = "log"),
                  weights = w,rel.gvar=1)
```

El `summary` del model `dbglm2` és el següent:

```
> summary(dbglm2)

Call:  dbglm.formula(formula = y ~ KmC + BonC + factor(MakeC), data = Motor1,
    family = poisson(link = "log"), metric = "gower", weights = w,
    rel.gvar = 1)

Deviance Residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-6.361000 -0.671100  0.043700 -0.000172  0.764000  5.987000

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 6977.53  on 294 degrees of freedom
```

```
Residual deviance: 454.05  on 276 degrees of freedom
AIC: Inf

Number of Fisher Scoring iterations: 6
Convergence criterion: DevStat
```

El `summary` vist amb el `glm1` i el que ara s'ha expressat pel `dbglm2` permet observar com les desviàncies residuals són diferents. En aquest últim ajust, la desviància ha baixat respecte el primer model (on la mètrica es considerava euclidiana). Els gràfics resultants del mètode genèric `plot` per l'objecte `dbglm2` es troben en la Figura 5.17 i són els mateixos que proporciona el mètode `plot` d'un objecte `glm`.

En el tercer i últim model que s'ajusta amb la funció `dbglm` es canvia l'argument `rel.gvar` (anteriorment a 1) per 0.9. Cal recordar que l'atribut `rel.gvar` determina el rang efectiu que s'usa en cada iteració DBLM. En aquest cas s'agafa el menor `eff.rank` tal que com a mínim s'expliqui el 90% de la variabilitat geomètrica de les dades:

```
> dbglm3 <- dbglm(y ~ KmC + BonC + factor(MakeC), data=Motor1,
                  metric="gower",family = poisson (link = "log"),
                  weights = w,rel.gvar=0.9)
```

La desviància residual ha pujat una mica respecte el model `dbglm2` (539.54), encara que segueix sent menor a la del `dbglm1`.

Els resultats que es deriven dels ajustos realitzats es troben en la Taula 5.7. En els dos models DBGLM (amb la mètrica de Gower) s'obté una desviància residual menor al cas del GLM usual. A més, utilitzant els mateixos paràmetres per estimar el model (`eff.rank=10`) la desviància residual del DBGLM és menor que la del GLM.

Poisson / Logarithmic	Residual Deviance	Eff.rank
DBGLM (`rel.gvar = 1`)	454.05	18
DBGLM (`rel.gvar = 0.90`)	539.54	10
GLM	779.36	10

Taula 5.7: Resultats pels ajustos del model Poisson amb funció d'enllaç logarítmica

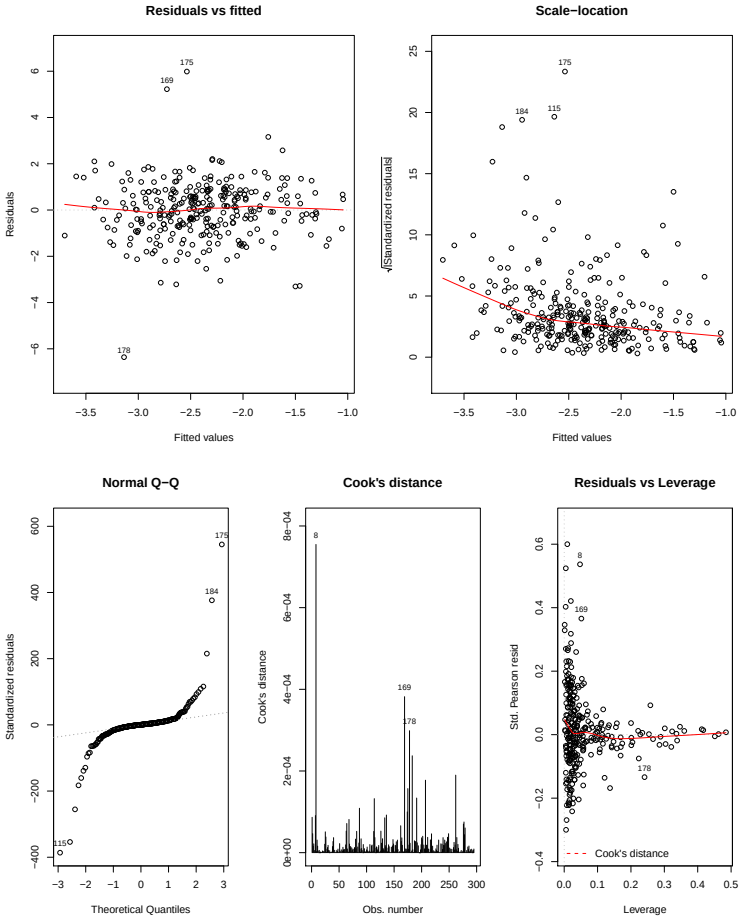

Figura 5.17: Gràfics que es desprenen del mètode genèric `plot` per un objecte de classe `dbglm`. En aquest cas, el `dbglm2`. S'hi aprecien tres observacions amb un residu certament alt: la 169 i 175 (positiu) i la 178 (negatiu).

GLM local basat en distàncies amb la funció `ldbglm`

En aquest apartat es considera la versió local del DBGLM per ajustar els models de predicció. Es realitza l'ajust pel `ldbglm` triant el paràmetre de suavitzat h que faci mínim el criteri d'optimització `GCV`. A més, la mètrica utilitzada és, igual que en `dbglm2` i `dbglm3`, la mètrica de Gower:

```
> ldbglm2 <- ldbglm(y ~ KmC + BonC + factor(MakeC), data=Motor1,
          metric="gower", family =poisson (link = "log"),
          weights = w,rel.gvar=1,method="GCV",h.range=c(2,15))
```

S'avalua el criteri `GCV` en 10 h's diferents en un rang de valors entre 2 i 15. La h òptima obtinguda és de 3.13. En la Figura 5.18 s'hi mostren els valors del GCV per les diferents h.

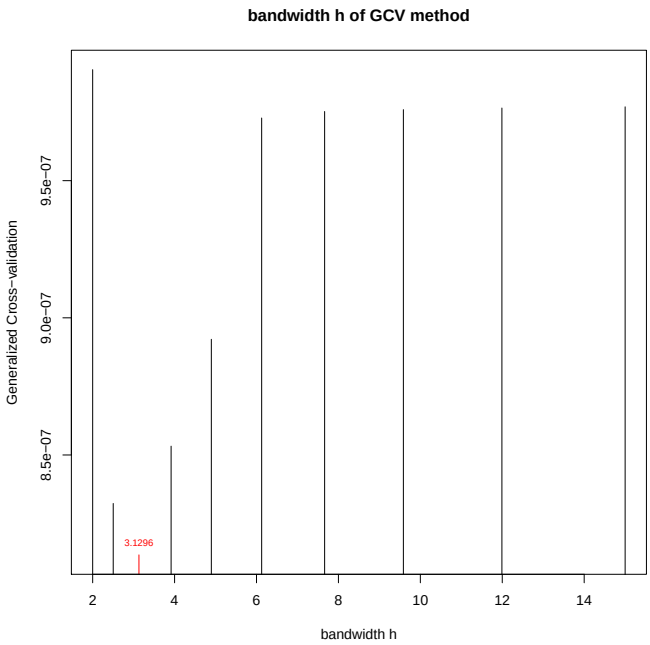

Figura 5.18: Criteri d'elecció de l'ample de banda òptim pel model `ldbglm2`. El GCV es minimitza per una h de 3.1296.

El `summary` del `ldbglm2` resumeix els resultats més importants que se'n deriven de l'ajust:

```
> summary(ldbglm2)

call:   ldbglm.formula(formula = y ~ KmC + BonC + factor(MakeC),
    data = Motor1, ... = list(metric = "gower"),
    family = poisson(link = "log"), method = "GCV", weights = w,
    h.range = c(2, 15), rel.gvar = 1)

Residuals:
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-0.1400000 -0.0200000  0.0000000  0.0001356  0.0100000  0.1600000

Number of Observations: 295
R-squared : 0.9474
Trace of smoother matrix: 51.89
family: poisson

kind of kernel= (1) Epanechnikov
optimal bandwidth h : 3.129594
GCV value criterion : 8.135821e-07
```

El `ldbglm2` té un R^2 d'aproximadament 0.95. Per tant, la variabilitat de les dades queda pràcticament tota explicada pel model.

Prediccions per a noves observacions

Es volen fer prediccions per una nova observació amb els nivells en les covariables: Distance = 750; Bonus = 1 i Make = 1. La traducció a l'R és immediata:

```
> newdata <- data.frame(KmC = 750, BonC = 1, MakeC = 1)
```

Les prediccions pels quatre models diferents que s'han ajustat: el `glm1`, el `dbglm2`, el `dbglm3` i el `ldbglm2` s'obtenen amb les següents comandes:

```
> pred.glm1 <- predict(glm1, newdata, type = "response")
> pred.dbglm2 <- predict(dbglm2, newdata, type = "response", type_var = "Z")
> pred.dbglm3 <- predict(dbglm3, newdata, type = "response", type_var = "Z")
> pred.ldbglm2 <- predict(ldbglm2, newdata, type = "response",
                          type_var = "Z")$fit
> result <- data.frame(pred.glm1=pred.glm1,pred.dbglm2=pred.dbglm2,
                       pred.dbglm3=pred.dbglm3,pred.ldbglm2=pred.ldbglm2)
> rownames(result)<-"new.data";
```

Els resultats difereixen en els quatre casos:

```
          pred.glm1 pred.dbglm2 pred.dbglm3 pred.ldbglm2
new.data  0.157937     0.17226   0.1709732     0.229962
```

Aquestes prediccions estimen la freqüència de reclamacions esperades per cotxes amb 750 km, l'última reclamació fa menys d'un any, i el primer tipus de cotxe que defineix la variable `make`.

Deixant el mateix quilometratge, però augmentat el Bonus al nivell 6 (és a dir més de 5 anys sense reclamar) i el setè tipus de cotxe, s'obté la predicció pel nou cas pels 4 models:

```
> newdata <- data.frame(KmC = 750, BonC = 6, MakeC = 7)
> pred.glm1 <- predict(glm1, newdata, type = "response")
> pred.dbglm2 <- predict(dbglm2, newdata, type = "response", type_var = "Z")
> pred.dbglm3 <- predict(dbglm3, newdata, type = "response", type_var = "Z")
> pred.ldbglm2 <- predict(ldbglm2, newdata, type = "response",
                          type_var = "Z")$fit
```

Aquestes contrasten amb les trobades anteriorment:

```
          pred.glm1 pred.dbglm2 pred.dbglm3 pred.ldbglm2
new.data  0.044634    0.043024   0.0484675    0.0426291
```

Ara el número esperat de reclamacions és molt més baix. Aquest fet era d'esperar. Mirant la Figura 5.15, ja s'observa que per `Bonus` alts la resposta disminueix. A més, el setè tipus de cotxe i poca distància realitzada (primer nivell) també solen prendre valors de la resposta baixos. Aquestes circumstàncies queden reflectides en les prediccions obtingudes (de nou diferents pels quatre models ajustats).

Capítol 6

Conclusions i qüestions pendents

En aquest treball s'ha pogut explicar de forma detallada en què consisteixen els mètodes estadístics basats en distàncies. S'han estudiat les versions basades en distàncies dels següents mètodes de regressió:

- DBLM: model lineal basat en distàncies.
- LDBLM: model local lineal basat en distàncies.
- DBGLM: model lineal generalitzat basat en distàncies.
- LDBGLM: model local lineal generalitzat basat en distàncies.
- DBPLS: mínims quadrats parcials basats en distàncies.

S'ha demostrat que únicament amb la matriu de distàncies entre individus al quadrat es poden aplicar les versions basades en distàncies dels mètodes estadístics més usuals. Aplicant multidimensional scaling a partir d'una matriu de dissimilituds s'obté una configuració euclidiana de les dades. Les coordenades euclidianes resultants, que s'expressen com a X, són les noves variables explicatives per a la resposta.

Els mètodes basats en distància són mètodes de predicció d'una variable resposta univariant un cop coneguda la matriu de distàncies al quadrat entre individus. Aquesta pot ser obtinguda directament, si les dades ja són distàncies, o calculades a partir de les variables explicatives mitjançant una certa funció de distàncies (mètrica). Destaquen les proposades en les funcions d'R `dist` (paquet `stats`), `dist` (paquet `proxy`) i `daisy` (paquet `cluster`). Recomanem la utilització de la mètrica de Gower pel cas que els predictors

estiguin formats per dades mixtes, és a dir, que n'hi hagi numèrics, categòrics o binaris. Aquests mètodes són vàlids per altres tipus de dades, com ara textuals o fins i tot dades funcionals (s'ha estudiat una aplicació on els predictors eren funcions). Els mètodes basats en distància són una bona alternativa als mètodes usuals al tenir un ampli ventall de possibilitats de configurar les dades com a inter-distàncies. A més, pel cas que s'utilitzi la mètrica euclidiana, el resultat d'una regressió usual i una basada en distàncies és equivalent.

En la llibreria d'R `dbstats` hi ha la codificació de tots els mètodes basats en distàncies explicitats i està disponible al servidor CRAN. S'ha intentat en tot moment que les funcions que hi ha al `dbstats` siguin el màxim de similars possible als mètodes clàssics ja implementats i contrastats en R. Això queda visible en els mètodes genèrics de cada funció: `print`, `summary`, `plot` i `predict`. La sortida que generen és molt similar a les funcions usuals. També permet la configuració de les distàncies per diferents vies. Per compatibilitat és pot processar un mètode `dbstats` introduint les variables explicatives Z en format `formula` (és la mateixa manera d'introduir les dades que la funció `lm`). Una altra manera és donar directament la matriu de distàncies, de classe `dist`, o la matriu de distàncies al quadrat, de classe `D2`. Per últim, també és possible a partir de la matriu de productes interiors G de classe `Gram`.

Aquestes tècniques s'han aplicat a dos exemples per tal d'il·lustrar el seu funcionament i assenyalar que són una alternativa molt apta per realitzar prediccions. La primera aplicació fa referència a la modelització d'una variable resposta contínua on els regressors són dades funcionals. Es realitza l'ajust per models lineals basats en distàncies (`dblm`), ajustos locals basats en distàncies (`ldblm`) i per mínims quadrats parcials (`dbplsr`). S'ha utilitzat la llibreria d'R `fda.usc`, on hi ha implementades diverses semimètriques per configurar les dades funcionals com una matriu de distàncies entre individus. A més, s'ha comparat la metodologia basada en distàncies envers la usual i donen resultats raonablement similars. Cal recalcar la potència que tenen els ajustos locals per fer prediccions. Els errors de predicció fent ús de la funció `ldblm` han baixat per tots els casos (dades sense derivar, primera i segona derivada) en comparació amb els mètodes paramètrics. Pel que fa referència a la funció `dbplsr`, dóna uns molt bons resultats i són comparables als obtinguts a l'ajustar un model lineal basat en distàncies (`dblm`).

La segona aplicació modelitza la prima de risc d'assegurances d'automòbil a Suècia segons al grup de risc al qual pertany cada vehicle. S'han utilitzat les funcions `dbglm` i `ldbglm` pel fet que el model és de la família Poisson. A més s'ha comparat amb el que s'obtindria si s'ajustés un model lineal

generalitzat per la funció d'R `glm`. La bondat de l'ajust per la metodologia `dbglm`, utilitzant la mètrica de Gower per codificar la matriu de distàncies entre individus, és millor que l'obtinguda pel mètode usual `glm`. Al contenir no només variables predictores continues, la dissimilitud de Gower és una bona mesura per quantificar les semblances entre observacions i això queda en evidència en aquest exemple.

Qüestions pendents

Tot i que el paquet `dbstats` ja conté una bona llista de mètodes estadístics basats en distàncies, cal indicar que encara se'n poden desenvolupar d'altres. A més, hi ha alguns aspectes que es podrien millorar en les tècniques ja implementades. Totes aquestes qüestions pendents, o possibles treballs en un futur es resumeixen en els següents punts:

- El cost computacional dels mètodes basats en distàncies és elevat per conjunts de dades grans: s'hauria de refinar el codi.
- Implementar la versió local del partial least squares.
- Implementar l'ANOVA basada en distàncies.
- Implementar el K-means basat en distàncies.
- Implementar l'anàlisi discriminant basat en distàncies.

Referències

Arenas, C. and C. Cuadras (2002). Recent statistical methods based on distances. *Contributions to Science 2*, 183–191.

Boj, E., A. Caballé, P. Delicado, and J. Fortiana (2012). `dbstats`*: distance-based statistics*. R package version 1.0.2.

Boj, E., P. Delicado, and J. Fortiana (2010). Distance-based local linear regression for functional predictors. *Computational Statistics and Data analysis 54*, 429–437.

Boj, E., A. Grané, J. Fortiana, and M. Claramunt (2007). Implementing pls for distance-based regression: computational issues. *Computational Statistics 22*, 237–248.

Borg, I. and P. Groenen (2005). *Modern Multidimensional Scaling*. New York: New York: Springer.

Carroll, J. and J. Chang (1970). Analysis of individual differences in multidimencional scaling via an nway generalitzation of eckart-young descomposition. *Psychometrika 35*, 283–319.

Coombs, C. (1964). A theory of data. *New York: John Wiley and Sons*, 80–180.

Cuadras, C. (1989). Distance analysis in discrimination and classification using both continuous and categorical variables. In Y. Dodge (Ed.), *Recent Developments in Statistical Data Analysis and Inference*, pp. 459–474. Elsevier Sciencie Publisher.

Cuadras, C. and C. Arenas (1990). A distance based regression model for prediction with mixed data. *Communications in Statistics A. Theory and Methods 19*, 2261–2279.

Everitt, B. (1993). *Cluster Analysis* (Third Edition ed.). London.

Faraway, J. (2012). `faraway`*: Functions and datasets for books by Julian Faraway*. R package version 1.0.5.

Febrero-Bande, M. and M. Oviedo (2011). *fda.usc: Functional Data Analysis and Utilities for Statistical Computing (fda.usc)*. R package version 0.9.5.

Ferraty, F. and P. Vieu (2006). *Non parametric functional data analysis. Theory and practice*. Springer.

Genolini, C. (2008). *A (Not So) Short Introduction to S4*.

Gower, J. (1968). Adding a point to vector diagrams in multivariate analysis. *Biometrika 55*, 582–585.

Gower, J. (1971). A general coefficient of similarity and some of its properties. *Biometrics 27*, 857–871.

Gower, J. and W. J. Krzanowski (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika 53*, 325–338.

Gower, J. C. and P. Legendre (1986). Metric and euclidean properties of dissimilarity coefficients. *Journal of classification*, 5–48.

Hallin, M. and J. F. Ingenbleek (1983). The Swedish automobile portfolio in 1977. a statistical study. *Skandinavisk Aktuarietidskrift (Scandinavian Actuarial Journal) 83*, 49–64.

Kalivas, J. (1997). Two data sets of near infrared spectra. *Chemometrics and Intelligent Laboratory Systems 37*, 255–259.

Kruskal, J. (1964, March). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *PSYCHOMETRIKA 29*, 1–27.

Leisch, F. (2009). *Creating R packages: A Tuturial*. R Development Core Team.

Mardia, K. V., J. T. Kent, and J. M. Bibby (1979). Multivariate analysis. *Academic Press*.

McCullagh, P. and J. Nelder (1989). *Generalized linear models*. United States of America: Chapman and Hall.

Nelder, J. and R. Wedderburn (1972). Generalized linear models. *Journal of the Royal Statistical Society 135*, 370–384.

Penrose, R. (1955). A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society 51*, 406–413.

Ramsay, J. (1977). Maximum likelihood estimation in multidimensional scaling. *Psychometrika 42*, 241–266.

Ramsay, J. and B. Silverman (2003). *fda: Functional Data Analysis*. R package version 2.2.6.

Ramsay, J. and B. Silverman (2005). *Functional Data Analysis* (Second ed.). New York: Springer.

Sammon, J. (1969). A nonlinear mapping for data structure analysis. *IEEE Trans. Comput. 18*, 401–409.

Shepard, R. (1977). The analysis of proximities: multidimensional scaling with an unknown distance function. *Psychometrika 27*, 219–246.

Swierenga, H., A. Weijer, R. Wijk, and L. Buydens (1999). Strategy for constructing robust multivariate calibration models. *Chemometrics and Intelligent Laboratory Systems*, 1–17.

Torgerson, W. S. (1958). Theory and methods of scaling. *New York: Wiley*.

Wold, H. (1966). Estimation of principal components and related models by iterative least squares. *Multivariate Analysis*, 391–420.