Local Classification Methods for Heterogeneous Classes

Julia Schiffner and Claus Weihs

Department of Statistics, Dortmund University of Technology
SFB 475 ‘Complexity Reduction in Multivariate Data Structures’

August 13, 2008

Outline

Introduction – Heterogeneous Classes

package **klaR**:

miscellaneous functions for classification and visualization

- classification into K given classes $c_1, \ldots, c_K$
- underlying assumption for many classification methods: random feature x homogeneous within the classes and heterogeneous across the classes

problem: heterogeneous classes

Introduction – Heterogeneous Classes

package **`klaR`**:

miscellaneous functions for classification and visualization

- classification into K given classes $c_1, \ldots, c_K$
- underlying assumption for many classification methods: random feature x homogeneous within the classes and heterogeneous across the classes

problem: heterogeneous classes

Introduction – Heterogeneous Classes

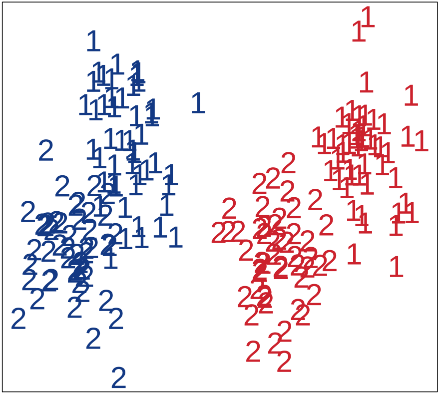
package **`klaR`**:

miscellaneous functions for classification and visualization

- classification into K given classes $c_1, \dots, c_K$
- underlying assumption for many classification methods: random feature x homogeneous within the classes and heterogeneous across the classes

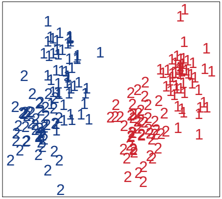
problem: heterogeneous classes

problem: heterogeneous classes



Introduction – Heterogeneous Classes

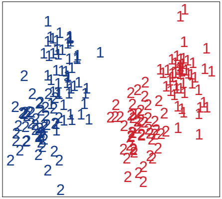
problem: heterogeneous classes



way out: local methods

- **classification methods based on mixture models**, e. g. mixture discriminant analysis (MDA)
- other prototype methods: K-means, learning vector quantization (LVQ)
- k-nearest-neighbor classifier (kNN)
- local likelihood methods: localized logistic regression, localized LDA (LLDA, in `klaR`)
- **local Fisher discriminant analysis (LFDA)**
- tree-based methods: CART, random forests

problem: heterogeneous classes



way out: local methods

- **classification methods based on mixture models**, e. g. mixture discriminant analysis (MDA)
- other prototype methods: K-means, learning vector quantization (LVQ)
- k-nearest-neighbor classifier (kNN)
- local likelihood methods: localized logistic regression, localized LDA (LLDA, in `klaR`)
- **local Fisher discriminant analysis (LFDA)**
- tree-based methods: CART, random forests

Mixture Models in Classification

- marginal density:

$$f(x) = \sum_{k=1}^{K} p_k f(x \mid c_k)$$

- model class conditional densities as mixtures
- data are generated by J sources s_j

- **hierarchical mixture model** (Titsias & Likas, 2002)

- **common components model** (Titsias & Likas, 2001)

Mixture Models in Classification

- marginal density:

$$f(x) = \sum_{k=1}^{K} p_k f(x \mid c_k)$$

- model class conditional densities as mixtures
- data are generated by J sources s_j

- **hierarchical mixture model** (Titsias & Likas, 2002)

- **common components model** (Titsias & Likas, 2001)

Mixture Models in Classification

- marginal density:

$$f(x) = \sum_{k=1}^{K} p_k f(x \mid c_k)$$

- model class conditional densities as mixtures
- data are generated by J sources s_j

- **hierarchical mixture model** (Titsias & Likas, 2002)

$$f(x) = \sum_{k=1}^{K} p_k \sum_{j=1}^{J} \pi_{jk} f(x \mid c_k, s_j)$$

- **common components model** (Titsias & Likas, 2001)

Mixture Models in Classification

- marginal density:

$$f(x) = \sum_{k=1}^{K} p_k f(x \,|\, c_k)$$

- model class conditional densities as mixtures
- data are generated by J sources s_j

- **hierarchical mixture model** (Titsias & Likas, 2002)

$$f(x \,|\, \theta) = \sum_{j=1}^{J} \pi_j \sum_{k=1}^{K} p_{kj} f(x \,|\, \mu_{kj}, \Sigma_{kj})$$

- **common components model** (Titsias & Likas, 2001)

Mixture Models in Classification

- marginal density:

$$f(x) = \sum_{k=1}^{K} p_k f(x \mid c_k)$$

- model class conditional densities as mixtures
- data are generated by J sources s_j

- **hierarchical mixture model** (Titsias & Likas, 2002)

$$f(x \mid \theta) = \sum_{j=1}^{J} \pi_j \sum_{k=1}^{K} p_{kj} f(x \mid \mu_{kj}, \Sigma_{kj})$$

- **common components model** (Titsias & Likas, 2001)

$$f(x) = \sum_{k=1}^{K} p_k \sum_{j=1}^{J} \pi_{jk} f(x \mid s_j)$$

Mixture Models in Classification

- marginal density:

$$f(x) = \sum_{k=1}^{K} p_k f(x \,|\, c_k)$$

- model class conditional densities as mixtures
- data are generated by J sources s_j

- **hierarchical mixture model** (Titsias & Likas, 2002)

$$f(x \,|\, \theta) = \sum_{j=1}^{J} \pi_j \sum_{k=1}^{K} p_{kj} f(x \,|\, \mu_{kj}, \Sigma_{kj})$$

- **common components model** (Titsias & Likas, 2001)

$$f(x \,|\, \theta) = \sum_{j=1}^{J} \pi_j \sum_{k=1}^{K} p_{kj} f(x \,|\, \mu_j, \Sigma_j) = \sum_{j=1}^{J} \pi_j f(x \,|\, \mu_j, \Sigma_j)$$

class posterior estimation

step 1: estimate source posteriors assuming a

- simple mixture model (unsupervised, "hm1")

$$f(x\,|\,\varphi) = \sum_{j=1}^{J} \pi_j f(x\,|\,\mu_j, \Sigma_j)$$

  EM algorithm $\Rightarrow P(s_j\,|\,x, \hat{\varphi})$

- common components model (supervised, "hm2")

$$f(x\,|\,\varphi_k) = \sum_{j=1}^{J} \pi_{jk} f(x\,|\,\mu_j, \Sigma_j)$$

  EM algorithm $\Rightarrow P(s_j\,|\,x, c(x), \hat{\varphi}_{c(x)})$

step 2: ML estimation of π_j, p_{kj}, μ_{kj}, and Σ_{kj} depending on x and the source posteriors

class posterior estimation

step 1: estimate source posteriors assuming a

- simple mixture model (unsupervised, "`hm1`")

$$f(x \mid \varphi) = \sum_{j=1}^{J} \pi_j f(x \mid \mu_j, \Sigma_j)$$

  EM algorithm $\Rightarrow P(s_j \mid x, \hat{\varphi})$

- common components model (supervised, "`hm2`")

$$f(x \mid \varphi_k) = \sum_{j=1}^{J} \pi_{jk} f(x \mid \mu_j, \Sigma_j)$$

  EM algorithm $\Rightarrow P(s_j \mid x, c(x), \hat{\varphi}_{c(x)})$

step 2: ML estimation of π_j, p_{kj}, μ_{kj}, and Σ_{kj} depending on x and the source posteriors

Hierarchical Mixture Classifier

class posterior estimation

step 1: estimate source posteriors assuming a

- simple mixture model (unsupervised, "`hm1`")

$$f(x\,|\,\varphi) = \sum_{j=1}^{J} \pi_j f(x\,|\,\mu_j, \Sigma_j)$$

  EM algorithm $\Rightarrow P(s_j\,|\,x, \hat{\varphi})$

- common components model (supervised, "`hm2`")

$$f(x\,|\,\varphi_k) = \sum_{j=1}^{J} \pi_{jk} f(x\,|\,\mu_j, \Sigma_j)$$

  EM algorithm $\Rightarrow P(s_j\,|\,x, c(x), \hat{\varphi}_{c(x)})$

step 2: ML estimation of π_j, p_{kj}, μ_{kj}, and Σ_{kj} depending on x and the source posteriors

Hierarchical Mixture Classifier

class posterior estimation

step 1: estimate source posteriors assuming a

- simple mixture model (unsupervised, "`hm1`")

$$f(x\,|\,\varphi) = \sum_{j=1}^{J} \pi_j f(x\,|\,\mu_j, \Sigma_j)$$

  EM algorithm $\Rightarrow P(s_j\,|\,x, \hat{\varphi})$

- common components model (supervised, "`hm2`")

$$f(x\,|\,\varphi_k) = \sum_{j=1}^{J} \pi_{jk} f(x\,|\,\mu_j, \Sigma_j)$$

  EM algorithm $\Rightarrow P(s_j\,|\,x, c(x), \hat{\varphi}_{c(x)})$

step 2: ML estimation of π_j, p_{kj}, μ_{kj}, and Σ_{kj} depending on x and the source posteriors

Hierarchical Mixture Classifier

class posterior estimation

step 1: estimate source posteriors assuming a

- simple mixture model (unsupervised, "`hm1`")

$$f(x\,|\,\varphi) = \sum_{j=1}^{J} \pi_j f(x\,|\,\mu_j, \Sigma_j)$$

  EM algorithm $\Rightarrow P(s_j\,|\,x, \hat{\varphi})$

- common components model (supervised, "`hm2`")

$$f(x\,|\,\varphi_k) = \sum_{j=1}^{J} \pi_{jk} f(x\,|\,\mu_j, \Sigma_j)$$

  EM algorithm $\Rightarrow P(s_j\,|\,x, c(x), \hat{\varphi}_{c(x)})$

step 2: ML estimation of π_j, p_{kj}, μ_{kj}, and Σ_{kj} depending on x and the source posteriors

Common Components Classifier

class posterior estimation

estimate π_j, p_{kj}, μ_j, and Σ_j by means of the EM algorithm

some details

- initialization of the EM algorithm: repeated execution of kmeans, posterior deviance
- number of sources J:

  assumed to be known in advance

  choice of J by means of a validation data set

Common Components Classifier

class posterior estimation

estimate π_j, p_{kj}, μ_j, and Σ_j by means of the EM algorithm

some details

- initialization of the EM algorithm: repeated execution of kmeans, posterior deviance
- number of sources J:

  assumed to be known in advance

  choice of J by means of a validation data set

Common Components Classifier

class posterior estimation

estimate π_j, p_{kj}, μ_j, and Σ_j by means of the EM algorithm

some details

- initialization of the EM algorithm: repeated execution of kmeans, posterior deviance
- number of sources J:

  assumed to be known in advance

  choice of J by means of a validation data set

R Functions

- `hm.cc`: generic function with methods for classes `"data.frame"`, `"matrix"`, and `"formula"`
- `hm.cc.start`: initialization of the EM algorithm
- arguments for `hm.cc`:

argument	explanation
`formula`, `data`	for class `"formula"`
`x`, `grouping`	required if no `formula` is given
`J`	number of sources
`method`	`"hm1"`, `"hm2"`, `"cc"`
`tries`, `iter`, `eps`	for `hm.cc.start` and EM algorithm
`threshold`	for subclass pruning in `"hm1"` and `"hm2"`

- `predict`-method for class `"hm.cc"`

R Functions

- `hm.cc`: generic function with methods for classes `"data.frame"`, `"matrix"`, and `"formula"`
- `hm.cc.start`: initialization of the EM algorithm
- arguments for `hm.cc`:

argument	explanation
`formula`, `data`	for class `"formula"`
`x`, `grouping`	required if no `formula` is given
`J`	number of sources
`method`	`"hm1"`, `"hm2"`, `"cc"`
`tries`, `iter`, `eps`	for `hm.cc.start` and EM algorithm
`threshold`	for subclass pruning in `"hm1"` and `"hm2"`

- `predict`-method for class `"hm.cc"`

R Functions

- `hm.cc`: generic function with methods for classes `"data.frame"`, `"matrix"`, and `"formula"`
- `hm.cc.start`: initialization of the EM algorithm
- arguments for `hm.cc`:

argument	explanation
`formula`, `data`	for class `"formula"`
`x`, `grouping`	required if no `formula` is given
`J`	number of sources
`method`	`"hm1"`, `"hm2"`, `"cc"`
`tries`, `iter`, `eps`	for `hm.cc.start` and EM algorithm
`threshold`	for subclass pruning in `"hm1"` and `"hm2"`

- `predict`-method for class `"hm.cc"`

Fisher Discriminant Analysis (FDA)

- supervised linear dimensionality reduction and classification
- FDA transformation matrix:

$$T_{FDA} = \arg\max_{T} \left(tr \left(T' S_w T \right)^{-1} T' S_b T \right)$$

- FDA projection: sample pairs in the same class are made close and sample pairs in different classes are separated from each other
- reduced dimension at most $K - 1$

Fisher Discriminant Analysis (FDA)

- supervised linear dimensionality reduction and classification
- FDA transformation matrix:

$$T_{FDA} = \arg\max_{T} \left(tr \left(T' S_w T \right)^{-1} T' S_b T \right)$$

- FDA projection: sample pairs in the same class are made close and sample pairs in different classes are separated from each other
- reduced dimension at most $K - 1$

Fisher Discriminant Analysis (FDA)

- supervised linear dimensionality reduction and classification

- FDA transformation matrix:

$$T_{FDA} = \arg\max_{T} \left(tr \left(T' S_w T \right)^{-1} T' S_b T \right)$$

- FDA projection: sample pairs in the same class are made close and sample pairs in different classes are separated from each other

- reduced dimension at most $K - 1$

Fisher Discriminant Analysis (FDA)

- supervised linear dimensionality reduction and classification
- FDA transformation matrix:

$$T_{FDA} = \arg\max_{T} \left(tr \left(T' S_w T \right)^{-1} T' S_b T \right)$$

- FDA projection: sample pairs in the same class are made close and sample pairs in different classes are separated from each other
- reduced dimension at most $K - 1$

Local FDA (LFDA) – Dimensionality Reduction

- supervised linear dimensionality reduction (Sugiyama, 2007) into *arbitrary* dimensional spaces
- heterogeneous classes: preserve the within-class local structure by introducing an affinity matrix A into the calculation of S_w and S_b (A_{ij}: affinity between x_i and x_j)
  $\Rightarrow$ downweight influence of far apart sample pairs in the same class
- LFDA transformation matrix:

$$T_{LFDA} = \arg\max_{T} \left(tr \left(T' S_w^A T \right)^{-1} T' S_b^A T \right)$$

- LFDA projection: **only nearby** sample pairs in the same class are made close and sample pairs in different classes are separated from each other

Local FDA (LFDA) – Dimensionality Reduction

- supervised linear dimensionality reduction (Sugiyama, 2007) into *arbitrary* dimensional spaces
- heterogeneous classes: preserve the within-class local structure by introducing an affinity matrix A into the calculation of S_w and S_b (A_{ij}: affinity between x_i and x_j)
  $\Rightarrow$ downweight influence of far apart sample pairs in the same class
- LFDA transformation matrix:

$$T_{LFDA} = \arg\max_{T} \left(tr \left(T' S_w^A T \right)^{-1} T' S_b^A T \right)$$

- LFDA projection: **only nearby** sample pairs in the same class are made close and sample pairs in different classes are separated from each other

Local FDA (LFDA) – Dimensionality Reduction

- supervised linear dimensionality reduction (Sugiyama, 2007) into *arbitrary* dimensional spaces
- heterogeneous classes: preserve the within-class local structure by introducing an affinity matrix A into the calculation of S_w and S_b (A_{ij}: affinity between x_i and x_j) $\Rightarrow$ downweight influence of far apart sample pairs in the same class
- LFDA transformation matrix:

$$T_{LFDA} = \arg\max_{T} \left(tr \left(T' S_w^A T \right)^{-1} T' S_b^A T \right)$$

- LFDA projection: **only nearby** sample pairs in the same class are made close and sample pairs in different classes are separated from each other

Local FDA (LFDA) – Dimensionality Reduction

- supervised linear dimensionality reduction (Sugiyama, 2007) into *arbitrary* dimensional spaces
- heterogeneous classes: preserve the within-class local structure by introducing an affinity matrix A into the calculation of S_w and S_b (A_{ij}: affinity between x_i and x_j)
  $\Rightarrow$ downweight influence of far apart sample pairs in the same class
- LFDA transformation matrix:

$$T_{LFDA} = \arg\max_{T} \left(tr \left(T' S_w^A T \right)^{-1} T' S_b^A T \right)$$

- LFDA projection: **only nearby** sample pairs in the same class are made close and sample pairs in different classes are separated from each other

LFDA – Classification

assumption: classes are composed from subclasses c_{km}

classification rule:

$$\hat{c}(x) = \arg\min_k \min_m \left\| T'_{LFDA} x - T'_{LFDA} \bar{x}_{km} \right\|$$

supervised case: subclasses are known

unsupervised case: subclasses are unknown

- spectral clustering within the K classes
- advantages: number of clusters is determined automatically, affinity matrix is used
- two methods: eigenvalues, eigenvectors

LFDA – Classification

assumption: classes are composed from subclasses c_{km}

classification rule:

$$\hat{c}(x) = \arg\min_k \min_m \left\| T'_{LFDA} x - T'_{LFDA} \bar{x}_{km} \right\|$$

supervised case: subclasses are known

unsupervised case: subclasses are unknown

- spectral clustering within the K classes
- advantages: number of clusters is determined automatically, affinity matrix is used
- two methods: eigenvalues, eigenvectors

LFDA – Classification

assumption: classes are composed from subclasses c_{km}

classification rule:

$$\hat{c}(x) = \arg\min_k \min_m \left\| T'_{LFDA} x - T'_{LFDA} \bar{x}_{km} \right\|$$

supervised case: subclasses are known

unsupervised case: subclasses are unknown

- spectral clustering within the K classes
- advantages: number of clusters is determined automatically, affinity matrix is used
- two methods: eigenvalues, eigenvectors

R Functions

- `lfda`: generic function with methods for classes `"data.frame"`, `"matrix"`, and `"formula"`
- arguments for `lfda`:

argument	explanation
`formula`, `data`	for class `"formula"`
`x`, `grouping`	required if no `formula` is given
`subgrouping`	subclass membership
`dimension`	desired dimensionality reduction
`norm.method`	method for normalizing the transformation matrix
`aff.method`	method for calculation of the affinity matrix
`cluster.method`	method for calculation of the subclass centers

- `predict`-method for class `"lfda"`

R Functions

- `lfda`: generic function with methods for classes `"data.frame"`, `"matrix"`, and `"formula"`
- arguments for `lfda`:

argument	explanation
`formula`, `data`	for class `"formula"`
`x`, `grouping`	required if no `formula` is given
`subgrouping`	subclass membership
`dimension`	desired dimensionality reduction
`norm.method`	method for normalizing the transformation matrix
`aff.method`	method for calculation of the affinity matrix
`cluster.method`	method for calculation of the subclass centers

- `predict`-method for class `"lfda"`

R Functions

- `lfda`: generic function with methods for classes `"data.frame"`, `"matrix"`, and `"formula"`
- arguments for `lfda`:

argument	explanation
`formula`, `data`	for class `"formula"`
`x`, `grouping`	required if no `formula` is given
`subgrouping`	subclass membership
`dimension`	desired dimensionality reduction
`norm.method`	method for normalizing the transformation matrix
`aff.method`	method for calculation of the affinity matrix
`cluster.method`	method for calculation of the subclass centers

- `predict`-method for class `"lfda"`

Summary & Outlook

hierarchical mixture and common components classifiers

- singularities in EM: variable selection, dimensionality reduction
- automatic determination of the number of clusters
- mixtures of other distributions
- ML estimation of parameters: criteria better suited for classification
- documentation of the fitting process (`trace`)

LFDA

- metric for classification rule
- kernel LFDA

Summary & Outlook

hierarchical mixture and common components classifiers

- singularities in EM: variable selection, dimensionality reduction
- automatic determination of the number of clusters
- mixtures of other distributions
- ML estimation of parameters: criteria better suited for classification
- documentation of the fitting process (`trace`)

LFDA

- metric for classification rule
- kernel LFDA

References

I. Czogiel, K. Luebke, M. Zentgraf, and C. Weihs.
Localized Linear Discriminant Analysis.
In R. Decker and H.-J. Lenz, editors, *Advances in Data Analysis*, volume 33, pages 133–140, Heidelberg, 2007. Springer.

T. Hastie and R. Tibshirani.
Discriminant Analysis by Gaussian Mixtures.
Journal of the Royal Statistical Society B, 58(1):155–176, 1996.

M. Sugiyama.
Dimensionality Reduction of Multimodal Labeled Data by Local Fisher Discriminant Analysis.
Journal of Machine Learning Research, 8:1027–1061, 2007.

M. K. Titsias and A. C. Likas.
Shared Kernel Models for Class Conditional Density Estimation.
IEEE Transactions on Neural Networks, 12(5):987–997, September 2001.

M. K. Titsias and A. C. Likas.
Mixture of Experts Classification Using a Hierarchical Mixture Model.
Neural Computation, 14:2221–2244, 2002.

L. Zelnik-Manor and P. Perona.
Self-Tuning Spectral Clustering.
In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 1601–1608. Cambridge, MA, 2005. MIT Press.