

Mixture Models of Categorization

Yves Rosseel
Ghent University

Abstract

Many currently popular models of categorization are either strictly parametric (e.g., prototype models, decision bound models) or strictly nonparametric (e.g., exemplar models) (Ashby & Alfonso-Reese, 1995). In this article, a family of semi-parametric classifiers is investigated where categories are represented by a finite mixture distribution. The advantage of these mixture models of categorization is that they contain several parametric models and nonparametric models as a special case. Specifically, it is shown that both decision bound models (Ashby & Maddox, 1992, 1993) and the generalized context model (Nosofsky, 1986) can be interpreted as two extreme cases of a common mixture model. Furthermore, many other (semi-parametric) models of categorization can be derived from the same generic mixture framework. In this article, several examples are discussed, and a parameter estimation procedure for fitting these models is outlined. To illustrate the approach, several specific models are fitted to a data set collected by McKinley and Nosofsky (1995). The results suggest that semi-parametric models are a promising alternative for future model development.

Formal models of categorization are often closely related to statistical methods of probability density estimation (Ashby & Alfonso-Reese, 1995). In statistics, a distinction is made between *parametric* estimators, that make strong assumptions about the distribution of the sample data, and *nonparametric* estimators that make only weak distributional assumptions. In accord with this distinction, Ashby and Alfonso-Reese defined *parametric classifiers* as those classifiers that make strong assumptions about the functional form of the category distributions, and *nonparametric classifiers* as classifiers that make almost no assumptions about the category form.

Prototype models (Reed, 1972) and decision bound models (Ashby & Maddox, 1992, 1993) are parametric classifiers, because they make strong assumptions about category structure. Decision bound models, for example, assume that the category distributions are multivariate normal (see Ashby, 1992, for a motivation). Despite this strong assumption (and the fact that these models can only predict linear or quadratic decision bounds), Ashby and Maddox (1992, 1993)

Note: this paper is a slightly modified version of the following paper: Rosseel, Y. (in press). Mixture models of categorization. *Journal of Mathematical Psychology*.

The original article is based on a doctoral dissertation submitted to the Ghent University (Belgium). I thank André Vandierendonck as my primary adviser.

The research was supported in part by Grant G.0353.99 from the Belgian National Science Foundation (Fundamental Human Sciences) to G. Storms and P. De Boeck.

Correspondence concerning this paper should be addressed to Yves Rosseel, Department of Data Analysis, Ghent University, Henri Dunantlaan 1, B-9000 Ghent (Belgium). Email: Yves.Rosseel@rug.ac.be

reported that these models provided excellent accounts of the data collected from experiments involving both normally and non-normally distributed categories. Recently, however, several studies suggested that human categorization is nonparametric, rather than parametric. For instance, McKinley and Nosofsky (1995) showed that participants were not constrained to use linear or quadratic decision bounds in a classification task using highly non-normal category distributions. In their study, a (nonparametric) exemplar model provided significantly better accounts of the data than did the (parametric) decision bound models. More recently, Ashby and Waldron (1999) reported the results of two experiments specifically designed to test whether human perceptual categorization is parametric or nonparametric. Again, the results strongly favored a nonparametric account.

Perhaps the most popular nonparametric models of categorization are the exemplar models, including the context model (Medin & Schaffer, 1978), the array model (Estes, 1986) and the generalized context model (Nosofsky, 1986). Ashby and Alfonso-Reese (1995) showed that these models are equivalent to a classifier using a nonparametric estimator (i.e., a kernel estimator) to approximate the category distributions. Importantly, due to their nonparametric nature, exemplar models predict that given enough experience with training exemplars, participants' response patterns should eventually approximate the underlying category distributions, no matter how complex they are. This seems rather unrealistic, since the classification abilities of humans are clearly limited (McKinley & Nosofsky, 1995; Ashby, Waldron, Lee, & Berkman, in press).

It seems that both parametric models and (kernel-based) nonparametric models of categorization are not without problems. Parametric models are not flexible enough to mimic the ability of humans to learn complex category distributions. On the other hand, exemplar models are perhaps too flexible as the kernel estimator can approximate any category distribution. However, a number of authors have proposed models that can be considered as *semi-parametric*. This includes Anderson's (1991) rational model, the covering version of Kruschke's (1992) ALCOVE model (not to be confused with the more popular exemplar-based version), and the striatal pattern classifier (SPC) (Ashby & Waldron, 1999). In all of these models, a category is represented by a number of 'subgroups'. And although the subgroups themselves can be parametric in nature (e.g., they may assume that the distribution of exemplars within a subgroup is multivariate normal), the fact that each category may contain several subgroups gives them much more flexibility than strictly parametric classifiers. However, by using a limited number of subgroups (typically much less than the number of exemplars within a category), the flexibility of these models is clearly limited, unlike strictly nonparametric classifiers.

In this article, I propose a family of semi-parametric classifiers where categories are represented by a *finite mixture distribution*. Finite mixture models are extremely flexible, and provide a generic framework for studying human categorization as a process of probability density estimation. Finite mixture models can mimic both parametric and nonparametric estimators, by varying the number (J) of mixture components included in the mixture model. If $J = 1$ for each category, the model reduces to a parametric estimator; if J equals the number of exemplars, the model becomes essentially an exemplar model. However, by using an intermediate number of mixture components, several semi-parametric models of categorization can be developed. The goal of this article is to explore the properties of these mixture models in the context of large, ill-defined categories with continuous dimensions¹.

The article is organized as follows. First, parametric and nonparametric approaches to probability density estimation are briefly reviewed, followed by the semi-parametric approach

¹When the dimensions are discrete, the use of finite mixture models is often called latent class analysis. For a discussion of latent class analysis in categorization research, see De Soete (1993).

taken by finite mixture models. In the same order, I review a parametric family of categorization models (i.e., decision bound models), a nonparametric model (i.e., the generalized context model), and a semi-parametric approach, where categories are represented by a finite mixture distribution. It is shown that both decision bound models and the generalized context model can be interpreted as two special cases of this generic framework. In the next sections, several specific mixture classifiers are derived from the generic framework. Two approaches are considered. In the first approach, categories are represented by a Gaussian mixture model, containing a relatively small number of mixture components. It will be argued that this approach can be interpreted as an extension of strictly parametric models. The second approach is motivated by exemplar models. I propose a reduced exemplar model, which is closely related to the generalized context model, but where the nonparametric nature of the kernel estimator is replaced by a finite mixture distribution. For both approaches, a parameter estimation procedure is outlined, and several versions of the models are fitted to a data set collected by McKinley and Nosofsky (1995). Finally, the implications of these results are discussed.

Parametric and Nonparametric Density Estimation

Probability density estimation is typically applied to *unlabeled* data (that is data without any class labels). In this case, we are given a finite number of unlabeled data points

$$\mathcal{X} = \{\mathbf{x}_n; n = 1, \dots, N\},$$

where \mathbf{x}_n is a d -dimensional vector. The aim of probability density estimation is to construct a function $\hat{p}(\mathbf{x})$ which approximates the true probability density function $p(\mathbf{x})$ from which the data points in \mathcal{X} were drawn. Another possibility is that the data points are labeled and belong to one of several classes C_k , where $k = 1, \dots, K$. In this case, we can use the same methods for estimating the *class-conditional* densities $p(\mathbf{x}|C_k)$ by considering one class at a time, or by using methods that estimate the unconditional density $p(\mathbf{x})$ and the class-conditional densities $p(\mathbf{x}|C_k)$ simultaneously.

Parametric approaches to density estimation assume that the unknown density function $p(\mathbf{x})$ can be represented as a specific functional form which contains several adjustable parameters. Once such a functional form has been chosen, the problem reduces to the estimation of the values of the parameters. The simplest, and most widely used, parametric model is the *normal* or *Gaussian* distribution. In d dimensions, the multivariate normal probability density function can be written in the form

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}, \end{aligned} \quad (1)$$

where the variable is the vector \mathbf{x} and the parameters are (a) the d -dimensional mean vector $\boldsymbol{\mu}$, and (b) the $d \times d$ *covariance matrix* $\boldsymbol{\Sigma}$; $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$. For a given data set \mathcal{X} , only the sample mean $\hat{\boldsymbol{\mu}}$ and sample covariance matrix $\hat{\boldsymbol{\Sigma}}$ need to be estimated from the data to construct the parametric estimator $\hat{p}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$.

Nonparametric methods of density estimation do not specify a particular functional form in advance. Instead, nonparametric methods are data-driven. Examples of nonparametric methods are the K -nearest-neighbor approach, and the method of histograms (Silverman, 1986). But perhaps the most widely used nonparametric estimator is the kernel estimator (Parzen, 1962). The approach taken by this kernel estimator is to define a *kernel function* (also known as a

Parzen window) centered on some point \mathbf{x} , and estimate $\hat{p}(\mathbf{x})$ by summing the value of the kernel function over all the data points in \mathcal{X} . Many different forms for the kernel function can be chosen, but a common choice is the multivariate normal distribution. In this case, the density function $\hat{p}(\mathbf{x})$ can be written as

$$\hat{p}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \mathcal{N}(\mathbf{x}; \mathbf{x}_n, \Sigma), \quad (2)$$

where the sum is taken over all data points in \mathcal{X} . Σ is a fixed $d \times d$ diagonal matrix, with elements σ_i^2 , corresponding to the *width* of the kernel along dimension i . Note that the width of the kernel acts as a *smoothing* parameter: if it is too large, the model density is over-smoothed, and will be a relatively poor estimator of the true density function; if it is too small, the model density will be noisy and very sensitive to the individual data points. Especially with few data points, the bandwidth of the kernel has a dramatic effect on the quality of the estimator (Silverman, 1986; Fukunaga, 1990). Another drawback of this method is that *all* the data points must be stored, in order to evaluate the density of some target data point. An attempt to relax this requirement is made by taking a smaller subset of J data points to construct the estimator. This approach is known as the ‘Reduced kernel estimator’ (Fukunaga, 1990). In this case, the approximating density function is given by

$$\hat{p}(\mathbf{x}) = \frac{1}{J} \sum_{j=1}^J \mathcal{N}(\mathbf{x}; \mathbf{x}_j, \Sigma), \quad (3)$$

where the sum is now taken over only J data points taken from \mathcal{X} . Of course, it is easy to see that this estimator will never be as accurate as the full kernel estimator. However, the quality of our estimator might be increased if we could adapt the positions (and the widths) of the kernels in response to the data. This is the approach taken by finite mixture models, which will be discussed in the next section.

Finite Mixture Models

Parametric and nonparametric approaches to density estimation each have their merits and limitations. Parametric methods assume a specific form (e.g., the multivariate normal distribution) for the density function, which may differ dramatically from the true density. By contrast, the nonparametric approach can approximate any form of density function, but the number of variables in the model grows directly with the number of data points. The kernel estimator, for example, uses a linear superposition of N kernel functions, with one kernel centered on each data point. Especially if N is large, this requires a lot of storage and computational effort for a possibly much simpler problem. Clearly, some intermediate or *semi-parametric* approach could offer a more practical solution.

One such a solution is the use of finite mixture models, where the unknown density function is approximated by a weighted linear combination of component densities $p(\mathbf{x} | j)$:

$$\hat{p}(\mathbf{x}) = \sum_{j=1}^J P(j) p(\mathbf{x} | j), \quad (4)$$

where J is the number of mixture components included in the model (McLachlan & Basford, 1988; Titterton, Smith, & Makov, 1985). The coefficients $P(j)$ are called the *mixing coefficients* or *mixing proportions* and satisfy the constraints $P(j) \geq 0$ and $\sum_{j=1}^J P(j) = 1$. An important property of these mixture models is that they can approximate any continuous density

to arbitrary accuracy, provided the model uses a sufficiently large number of components, and provided the parameters are chosen correctly. Several functional forms for the mixture densities are possible, but again a common choice is the multivariate normal distribution:

$$\hat{p}(\mathbf{x}) = \sum_{j=1}^J P(j) \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \quad (5)$$

This model is called a *Gaussian mixture model*. The model parameters of the Gaussian mixture model can be summarized by the parameter vector $\boldsymbol{\theta} = \{P(j), \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}$ where $j = 1, \dots, J$. A method for estimating the parameters of a Gaussian mixture model, based on the EM algorithm is discussed in appendix A. Note that if $J = 1$, the model reduces to a simple parametric estimator. On the other hand, if $J = N$, the model resembles the properties of the kernel estimator. Obviously, the number of mixture components plays a crucial role in determining the characteristics of the estimator. When fitting a finite mixture model in practice, the number of components is often unknown and must be inferred from the data. Unfortunately, this is a difficult problem as typical model comparison tests (e.g., the likelihood ratio statistic to test for the smallest value of J compatible with the data) can not be used, because for mixture models, certain regularity conditions are not met (McLachlan & Basford, 1988). As an alternative, McLachlan and Basford (1988) use a resampling method. Other approaches have been considered by Banfield and Raftery (1993), and Furman and Lindsay (1994), among others.

Parametric and Nonparametric Models of Categorization

Many models of categorization are equivalent to a process in which the observer estimates the likelihood that a stimulus \mathbf{x} belongs to one of several categories C_k , where $k = 1, \dots, K$ (Ashby & Alfonso-Reese, 1995). Depending on the type of estimator used to estimate these *class-conditional* density functions $p(\mathbf{x} | C_k)$, a distinction can be made between parametric, nonparametric and semi-parametric models. Surprisingly however, Ashby and Alfonso-Reese (1995) showed that many of the currently popular models of categorization can be considered either strictly parametric, or strictly nonparametric. Decision bound models, for example, are strictly parametric while many exemplar models are strictly nonparametric. In this section, the parametric and nonparametric assumptions of these models are briefly reviewed. For the nonparametric approach, we limit our discussion to the generalized context model (Nosofsky, 1986, 1992); for the parametric approach, only decision bound models (Ashby & Maddox, 1993) are discussed.

General recognition theory and decision bound models

The representation assumptions of decision bound models are based on the general recognition theory (GRT) (Ashby & Townsend, 1986). According to GRT, the repeated presentation of the same stimulus does not always lead to the same perceptual effect, because of inherent *noise* in our sensory and perceptual system. Therefore, the observer's percept of a presented stimulus vector \mathbf{x}_n is modeled by a random vector \mathbf{x}_{pn} in a multidimensional perceptual space by assuming

$$\mathbf{x}_{pn} = \mathbf{x}_n + \mathbf{e}_{pn}, \quad (6)$$

where \mathbf{e}_{pn} is a random vector with zero mean that represents *perceptual noise*. Typically, it is assumed that \mathbf{e}_{pn} is multivariate normal with covariance matrix $\boldsymbol{\Sigma}_{pn}$. Hence, the perceptual effects of each exemplar can be represented by a multivariate normal distribution

$$p(\mathbf{x} | n) = \mathcal{N}(\mathbf{x}; \mathbf{x}_n, \boldsymbol{\Sigma}_{pn}). \quad (7)$$

A category is represented perceptually as a probability mixture of the individual exemplar distributions. When category C_k contains a finite number of N_k exemplars denoted by the set $\mathcal{X}_k = \{\mathbf{x}_n; n = 1, \dots, N_k\}$, the probability density function of the perceptual effects associated with this category is given by

$$p(\mathbf{x} | C_k) = \sum_{n=1}^{N_k} P(\mathbf{x}_n | C_k) \mathcal{N}(\mathbf{x}; \mathbf{x}_n, \boldsymbol{\Sigma}_{pn}), \quad (8)$$

where $P(\mathbf{x}_n | C_k)$ denotes the probability that stimulus \mathbf{x}_n is presented as a member of category C_k . Several special cases of this GRT model can be derived by constraining the covariance matrices $\boldsymbol{\Sigma}_{pn}$. For instance, the stimulus-invariant GRT model assumes $\boldsymbol{\Sigma}_{pn} = \boldsymbol{\Sigma}_p$ for all n . The uncorrelated GRT model assumes $\boldsymbol{\Sigma}_p$ is diagonal, and the simple GRT model assumes that $\boldsymbol{\Sigma}_p = \sigma_p^2 \mathbf{I}$ (Ashby & Maddox, 1993).

In the special case that the category exemplars themselves are multivariate normally distributed in stimulus space, and perceptual noise is stimulus-invariant, then the perceived category distribution reduces to a multivariate normal distribution (Ashby, 1992; Ashby & Lee, 1991):

$$p(\mathbf{x} | C_k) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{pk}, \boldsymbol{\Sigma}_{pk}), \quad (9)$$

where $\boldsymbol{\mu}_{pk}$ and $\boldsymbol{\Sigma}_{pk}$ denote the perceived category C_k mean vector and covariance matrix respectively. Note that since the noise vectors \mathbf{e}_{pn} are assumed to have zero mean, $\boldsymbol{\mu}_{pk} = \boldsymbol{\mu}_k$ and if the stimulus-invariant noise assumption holds, $\boldsymbol{\Sigma}_{pk} = \boldsymbol{\Sigma}_k + \boldsymbol{\Sigma}_p$. Therefore, we can rewrite equation (9) as

$$p(\mathbf{x} | C_k) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k + \boldsymbol{\Sigma}_p). \quad (10)$$

Of course, any classifier that uses the multivariate normal distribution (or some other distributional family) to model the class-conditional density function $p(\mathbf{x} | C_k)$ can be considered a parametric classifier. Hence, both the general linear classifier and the general quadratic classifier (Ashby & Maddox, 1993) are parametric classifiers, since they both assume the category distributions are multivariate normal. (See Ashby & Alfonso-Reese, 1995, for a proof).

The generalized context model

Exemplar models assume that humans represent categories by storing *every* exemplar (together with its category label) in memory. Category decisions are based on a similarity computation between a probe stimulus and stored exemplars. Perhaps the most popular exemplar model is the generalized context model (GCM) (Nosofsky, 1984, 1986, 1992). Let $\mathcal{X}_k = \{\mathbf{x}_n; n = 1, \dots, N_k\}$ be the set of category C_k exemplars stored in memory. According to the GCM, the posterior probability that stimulus \mathbf{x} is classified in category C_k is given by

$$P(C_k | \mathbf{x}) = \frac{b_k h_k(\mathbf{x})^\gamma}{\sum_{l=1}^K b_l h_l(\mathbf{x})^\gamma}, \quad (11)$$

where b_k is a response bias, and $h_k(\mathbf{x})$ is given by the summed similarity between \mathbf{x} and every stored category C_k exemplar:

$$h_k(\mathbf{x}) = \sum_{n=1}^{N_k} \exp\{-d(\mathbf{x}, \mathbf{x}_n)^q\}, \quad (12)$$

where $q = 1$ yields an exponential function, and $q = 2$ yields a Gaussian function; $d(\mathbf{x}, \mathbf{x}_n)$ is a measure of the psychological distance from \mathbf{x} to \mathbf{x}_n . The γ parameter reflects the amount of

determinism in responding (Ashby & Maddox, 1993)². The quantity $h_k(\mathbf{x})$ can be interpreted as a measure of category similarity, and therefore as a measure of evidence that stimulus \mathbf{x} belongs to this category. A slightly more general definition of category similarity was given by Nosofsky (1988), to allow for unequal stimulus presentation frequencies. The resulting model, known as the *frequency sensitive* GCM assumes

$$h_k(\mathbf{x}) = \sum_{n=1}^{N_k} P(\mathbf{x}_n | C_k) \exp\{-d(\mathbf{x}, \mathbf{x}_n)^q\}, \quad (13)$$

where $P(\mathbf{x}_n | C_k)$ is simply given by f_n/N_k (f_n denotes the number of stimulus \mathbf{x}_n presentations). The psychological distance $d(\mathbf{x}, \mathbf{x}_n)$ in (12) and (13) is computed using a weighted Minkowski r -metric in a d -dimensional space:

$$d(\mathbf{x}, \mathbf{x}_n) = c \left[\sum_{i=1}^d w_i |x_i - x_{ni}|^r \right]^{1/r}, \quad (14)$$

where w_i is the proportion of attention allocated to dimension i . The nonnegative parameter c scales the psychological space and can be interpreted as a measure of overall stimulus discriminability (Nosofsky, 1986). The exponent r defines the distance metric: the value $r = 1$ produces the city-block metric, whereas $r = 2$ produces the Euclidean metric. In what follows, I will only consider these cases where $q = r$.

It is easily shown that $h_k(\mathbf{x})$ in (12) and (13) is essentially a nonparametric estimator of the class-conditional density function $p(\mathbf{x} | C_k)$. (See Ashby & Alfonso-Reese, 1995, for a proof). For example, let $q = r = 2$, and using (13) and (14):

$$\begin{aligned} h_k(\mathbf{x}) &= \sum_{n=1}^{N_k} P(\mathbf{x}_n | C_k) \exp \left\{ -c^2 \sum_{i=1}^d w_i (x_i - x_{ni})^2 \right\} \\ &= \sum_{n=1}^{N_k} P(\mathbf{x}_n | C_k) \exp \left\{ -\frac{1}{2} \sum_{i=1}^d 2c^2 w_i (x_i - x_{ni})^2 \right\} \\ &= \sum_{n=1}^{N_k} P(\mathbf{x}_n | C_k) \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{x}_n)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \mathbf{x}_n) \right\}, \end{aligned} \quad (15)$$

where $\boldsymbol{\Sigma}$ is a diagonal $d \times d$ matrix with elements $1/(2c^2 w_i)$ for $i = 1, \dots, d$. Without affecting the posterior probabilities in (11), a class-independent constant term can be added before the exponential in (15):

$$\begin{aligned} p(\mathbf{x} | C_k) &= \sum_{n=1}^{N_k} P(\mathbf{x}_n | C_k) \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{x}_n)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \mathbf{x}_n) \right\} \\ &= \sum_{n=1}^{N_k} P(\mathbf{x}_n | C_k) \mathcal{N}(\mathbf{x}; \mathbf{x}_n, \boldsymbol{\Sigma}). \end{aligned} \quad (16)$$

Note the similarity with the Gaussian kernel estimator in (2). A similar derivation can be given for the case where $q = r = 1$, but where a Laplacian distribution is used instead of the normal distribution, and where the diagonal elements of the covariance matrix $\boldsymbol{\Sigma}$ are given by $1/(2c w_i)$

²The original response rule of the GCM used $\gamma = 1$ (Nosofsky, 1984, 1986).

for $i = 1, \dots, d$ (Ashby & Alfonso-Reese, 1995). Apart from the fact that this illustrates the nonparametric nature of the generalized context model, equation (16) also allows us to interpret each exemplar \mathbf{x}_n as a random vector having a multivariate normal distribution³ centered on \mathbf{x}_n , and with a *diagonal* covariance matrix Σ . Therefore, the representational assumptions of the general context model are closely related to the GRT model, as stated in the next theorem:

Theorem 1 *The representational assumptions of the generalized context model as defined in (13) and (14) are equivalent to the stimulus-invariant, uncorrelated GRT model, if $q = p = 2$.*

It is clear from (16) that a category in the generalized context model is modeled as a (weighted) sum of N_k exemplar distributions $p(\mathbf{x}|n)$. By contrast, decision bound models, which are based on the normal GRT model as defined in (10), use only 1 (normal) distribution to model a category. As a result, these models can be regarded as two extreme cases on a continuum where the number (J) of components that make up a category distribution is allowed to range in the interval $[1, N_k]$. Clearly, many alternative models can be developed where $1 < J < N_k$. A generic framework that incorporates these models is presented in the next section.

Finite mixture models of categorization

In this section, finite mixture models are used to construct a broad family of semi-parametric classifiers, all based on a common framework. First, the (representational) assumptions of this generic framework are defined, and it is shown how by further specifying several aspects, more specific (and hence testable) models can be derived. Next, the relationship between mixture models and several existing semi-parametric models of categorization is discussed. Finally, a procedure for estimating the parameters of fully specified mixture classifiers is described. This procedure is then used to fit several mixture classifiers to a published data set collected by McKinley and Nosofsky (1995).

A generic mixture model of categorization

The representational assumptions of the generic mixture model of categorization are described in the following definition:

Definition 1 *The generic mixture model of categorization makes three assumptions:*

1. *A specific exemplar \mathbf{x}_n is represented perceptually by a random vector having a multivariate distribution $p(\mathbf{x}|n)$ centered on \mathbf{x}_n and with a covariance matrix Σ_{pn} .*
2. *The unconditional probability density function of a set of exemplars (belonging to K categories) is modeled as a finite mixture distribution:*

$$p(\mathbf{x}) = \sum_{j=1}^J P(j) p(\mathbf{x}|j), \quad (17)$$

where J is the number of mixture components used in the mixture model; $P(j)$ denotes the unconditional mixture proportions, and $p(\mathbf{x}|j)$ are the individual component densities.

3. *The probability density function of a category C_k is modeled as a finite mixture distribution sharing the same mixture components $p(\mathbf{x}|j)$ of the unconditional mixture distribution $p(\mathbf{x})$:*

$$p(\mathbf{x}|C_k) = \sum_{j=1}^J P(j|C_k) p(\mathbf{x}|j), \quad (18)$$

where $P(j|C_k)$ denotes the class-conditional mixture proportions.

³Again, for the case where $q = r = 1$, the Laplacian distribution should be used instead.

The first assumption in Definition 1 is based on GRT, and reflects the effect of noise in our sensory and perceptual system. Another interpretation is that the storage of individual exemplars in memory is fuzzy, and when a particular exemplar \mathbf{x}_n is recalled, no distinction can be made between the original exemplar, and any exemplar drawn from $p(\mathbf{x} | n)$. The true probability density function of these exemplar distributions (ignoring their category labels) can be expressed by

$$p(\mathbf{x}) = \sum_{n=1}^N P(n) p(\mathbf{x} | n), \quad (19)$$

where $P(n)$ denotes the probability that \mathbf{x}_n is presented as a stimulus. However, according to the generic mixture model, this unconditional density function $p(\mathbf{x})$ is approximated by a finite mixture model using J mixture components $p(\mathbf{x} | j)$, where $1 \leq J \leq N$ and $J \geq K$. A specific category C_k is represented as a finite mixture distribution sharing the same mixture components $p(\mathbf{x} | j)$ of the unconditional distribution. The mixture proportions $P(j | C_k)$ determine the degree to which each of the J mixture components belongs to a particular category. In other words, the mixture proportions express the strength of association between mixture component j , and category C_k . This sharing of mixture components has several advantages. First, it allows for overlapping categories in a natural way. Second, the unconditional mixture proportions $P(j)$ and the class-conditional proportions $P(j | C_k)$ are independent. In other words, the importance of a mixture component j within a category distribution is determined solely by the proportion $P(j | C_k)$ and not by $P(j)$ or $P(j | C_l)$ for $l \neq k$. The only constraint is that $\sum_{j=1}^J P(j | C_k) = 1$. Third, by using a common pool of mixture components, this representation suggests that an observer might have (partially) learned the unconditional mixture distribution in (17) without any knowledge of the category labels (i.e., unsupervised). And perhaps later, when the observer receives information about the category labels of some exemplars (e.g., in a supervised learning task), the class-conditional mixture proportions $P(j | C_k)$ are learned (while the common mixture components are further adjusted). This opens the possibility to construct models of category learning that adopt a combination of both supervised and unsupervised learning mechanisms (Rosseel, 1998).

Specific mixture models of categorization

In the definition of the generic mixture model of categorization, many aspects have been left open. Further specifications and constraints are needed in order to develop a testable and falsifiable model. First of all, a response rule is needed to predict the posterior probabilities $P(C_k | \mathbf{x})$. The response rule used throughout this article is similar to the one used by the (deterministic) generalized context model (see Eq. 11):

$$P(C_k | \mathbf{x}) = \frac{b_k p(\mathbf{x} | C_k)^\gamma}{\sum_{l=1}^K b_l p(\mathbf{x} | C_l)^\gamma}, \quad (20)$$

where b_k is a response bias ($b_k > 0$ and $\sum_{k=1}^K b_k = 1$), and γ determines the amount of determinism in responding. Both the response biases b_k and γ are free parameters of the model.

To compute the posterior probabilities in (20), the functional form of the component distributions $p(\mathbf{x} | j)$ must be specified. Several such distributions can be used, but interesting candidates are the normal distribution, the logistic distribution, and the Laplacian distribution. All these distributions are unimodal, symmetric, and completely described by a mean vector and a covariance matrix. If the normal distribution is chosen, the exemplar distributions can be written as:

$$p(\mathbf{x} | n) = \mathcal{N}(\mathbf{x}; \mathbf{x}_n, \boldsymbol{\Sigma}_{pn}), \quad (21)$$

as in (7). The density function of the mixture components can be written as:

$$p(\mathbf{x}|j) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_{pj}), \quad (22)$$

where $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_{pj}$ denote the perceived mean vector and covariance matrix of the j -th component distribution respectively. This special case is called the normal or Gaussian mixture classifier. In a similar way, a logistic or Laplacian version can be defined. Just like in (10), the covariance matrix $\boldsymbol{\Sigma}_{pj}$ can be decomposed as a sum of two terms: $\boldsymbol{\Sigma}_{pj} = \boldsymbol{\Sigma}_j + \boldsymbol{\Sigma}_p$. The first term depends on the specific component, and reflects the local covariance structure around the mean vector $\boldsymbol{\mu}_j$. The second term $\boldsymbol{\Sigma}_p$, however, is common for all mixture components, and acts as a smoothing parameter.

Once a specific functional form has been specified, there are several ways to further constrain the model. Just like in GRT, the covariance matrix of the exemplar distributions in (21) can be constrained to be stimulus-invariant ($\boldsymbol{\Sigma}_{pn} = \boldsymbol{\Sigma}_p$ for all n), uncorrelated ($\boldsymbol{\Sigma}_p$ is diagonal), or simple ($\boldsymbol{\Sigma}_p = \sigma_p^2 \mathbf{I}$). In the same way, the covariance matrices $\boldsymbol{\Sigma}_j$ can be constrained to be diagonal (i.e., axis-aligned), or $\boldsymbol{\Sigma}_j = \sigma_j^2 \mathbf{I}$. Furthermore, it is possible to use the same covariance matrix for all the mixture components: $\boldsymbol{\Sigma}_j = \boldsymbol{\Sigma}$ for all j . Note that the impact of these constraints on the covariance matrices is highly dependent on the number (J) of mixture components used in the model. For example, a normal distribution with high correlations between the dimensions can often be equally well described by either a single density function with a full covariance matrix, or by a mixture distribution containing a (possibly large) number of components with highly constrained covariance matrices. In other words, there is a trade-off between the number of mixture components, and the constraints on the covariance matrices (and hence the number of model parameters).

A different set of constraints is related to the class-conditional mixture proportions $p(\mathbf{x} | C_k)$. For example, it is possible to restrict these proportions so that any mixture component belongs to one category only. This *class-specific* version of the generic mixture model is defined as follows:

Definition 2 *The class-specific mixture model of categorization is a special case of the generic model as defined in Definition 1 where the following holds: if $P(j | C_k) > 0.0$ for some k , then $P(j | C_l) = 0.0$ for all $l \neq k$.*

In this case, the class-conditional density function of a category C_k can be written as:

$$p(\mathbf{x} | C_k) = \sum_{j=1}^{J_k} P(j | C_k) p(\mathbf{x} | j), \quad (23)$$

where J_k denotes the number of mixture components used for the category C_k distribution only. Two special cases of the class-specific mixture model are particularly interesting: if $J_k = 1$ and if $J_k = N_k$. The former is closely related to decision bound models, while the latter is related to the generalized context model, as stated in the following theorems:

Theorem 2 *The representational assumptions of decision bound models as defined in (10) are equivalent to a class-specific Gaussian mixture model if $J_k = 1$.*

As a result, both the general quadratic classifier, and the general linear classifier can be considered as two special cases of a class-specific Gaussian mixture classifier.

Theorem 3 *The category representation of the (frequency-sensitive) generalized context model as defined in (13) and (14) where $q = r$, is equivalent to a class-specific mixture model if $J_k = N_k$, and the class-conditional density function can be written as*

$$p(\mathbf{x} | C_k) = \sum_{n=1}^{N_k} P(\mathbf{x}_n | C_k) p(\mathbf{x} | n), \quad (24)$$

where the exemplar distributions $p(\mathbf{x} | n)$ are multivariate distributions with mean vector \mathbf{x}_n , and a common diagonal covariance matrix Σ . If $q = r = 2$, then $p(\mathbf{x} | n)$ is multivariate normal and the diagonal elements of Σ are given by

$$\frac{1}{2 c^2 w_i},$$

for $i = 1, \dots, d$. If $q = r = 1$, then $p(\mathbf{x} | n)$ is multivariate Laplacian and the diagonal elements of Σ are given by

$$\frac{1}{2 c w_i},$$

for $i = 1, \dots, d$.

These theorems are an illustration of how a (class-specific) mixture classifier can mimic both parametric and nonparametric classifiers, by setting the number of mixture components to some extreme number. The theorems can also be seen as an illustration of how *specific* (and hence testable) models can be derived from the generic mixture model.

Other semi-parametric models of categorization

Recently, a number of authors have proposed models that can also be considered as semi-parametric. In this section, I briefly consider the rational model (Anderson, 1991), the covering version of ALCOVE (Kruschke, 1992), and the striatal pattern classifier (SPC) (Ashby & Waldron, 1999; Ashby et al., in press), and discuss their relationship with mixture models.

The Rational model. In the rational model, categories are represented in terms of multiple clusters (or subcategories). The number of clusters depends on a *coupling* parameter, which is a free parameter of the model. If the number of clusters is smaller than the number of exemplars, but greater than the number of categories, the rational model can be regarded as a semi-parametric classifier. In fact, although there are many formal differences, the rational model is closely related to the mixture classifiers considered in this article. At least conceptually, the clusters of the rational model seem to correspond to the mixture components of the mixture classifiers. In particular, if all exemplar features are continuous, the categories in the rational model can be represented by a mixture of (multivariate) t distributions⁴.

⁴In the rational model, category labels are regarded as just another feature to be predicted. This feature is typically not continuous (and hence, the mixture components can not be described by a multivariate t distribution). In general, exemplars may have both discrete and continuous features. However, because features are assumed to be independent (within a cluster), the marginal distributions of the continuous features are still (univariate) t distributions, while the marginal distributions of the discrete features follow a Dirichlet distribution. (See Anderson, 1991, for further details).

ALCOVE. Although the most widely known version of ALCOVE is exemplar-based, its name still refers to the original covering version where, instead of using all exemplars, categories are represented by a smaller number (J) of hidden nodes covering the input space. In this version of ALCOVE, the evidence that stimulus \mathbf{x} belongs to category C_k is given by:

$$h_k(\mathbf{x}) = \sum_{j=1}^J w_{kj} \exp\{-d(\mathbf{x}, \boldsymbol{\mu}_j)^q\}, \quad (25)$$

where the distance $d(\mathbf{x}, \boldsymbol{\mu}_j)$ is defined as in (14); $\boldsymbol{\mu}_j$ is the position of the j -th hidden node in the psychological space; the weights w_{kj} denote the strength of association between category C_k and hidden node j . Using the same steps as in (15) and (16), it is straightforward to rewrite (25) in terms of a mixture distribution involving Gaussian (if $q = p = 2$) or Laplacian (if $q = p = 1$) densities.

The striatal pattern classifier. Ashby and Waldron (1999) have proposed a model called the striatal pattern classifier (SPC). According to this model, categories are represented by a number of *striatal units*, forming a low-resolution map of the perceptual space. Using the vocabulary used in this article, the striatal pattern classifier can be described as a class-specific Gaussian mixture classifier where the mixture components share a common covariance matrix: $\boldsymbol{\Sigma}_j = \boldsymbol{\Sigma}$ for all j . Moreover, the covariance matrix is assumed to be spherical: $\boldsymbol{\Sigma} = \lambda \mathbf{I}$ where λ reflects the perceived variance of the exemplars, and depends on the perceptual noise σ^2 .

At least at the representational level, it seems that the properties of all these semi-parametric classifiers are well captured within the framework of mixture classifiers. Sometimes, expressing an existing model in terms of the generic mixture framework is merely a matter of reparameterization. However, apart from the ability to describe a large number of categorization models within a common framework, the aim of the mixture framework is also to provide a generic parameter estimation procedure that can be used for a large range of different models. Furthermore, the procedure should be able to deal with situations where more than two categories are involved. A procedure that works well when the number of mixture components is fairly small is described in the next section.

Fitting a specific mixture model to a data set

As a concrete example, consider a series of Gaussian mixture classifiers with common mixture components and full covariance matrices. Furthermore, assume that the number of mixture components is rather small (say, $J \leq 10$). Let M- x J denote a mixture model containing x mixture components. For example, M-4J and M-6J contain 4 and 6 mixture components respectively. In this section, we outline a procedure for fitting these models to a data set, collected from a typical human categorization experiment (involving continuous dimensions).

In such an experiment, it is often convenient to make a distinction between a training phase, and a test phase. The former can be characterized by a labeled data set $\mathcal{T} = \{\mathbf{x}_n, k_n; n = 1 \dots, N\}$ where N is the total number of stimuli presented during the training phase, and k_n denotes the *true* category label of stimulus \mathbf{x}_n . The test phase can be summarized by a labeled data set $\mathcal{R} = \{\mathbf{x}_t, r_t; t = 1 \dots, T\}$ where T is the number of test stimuli, and the r_t now denote the responses (category labels) given by the participant. For each of the T test stimuli, a particular model predicts the posterior probabilities $P(C_k | \mathbf{x}_t)$ for each of the K categories. The likelihood of observing the responses in \mathcal{R} is given by

$$\mathcal{L}_r(r_1, r_2, \dots, r_T) = \prod_{t=1}^T P(r_t | \mathbf{x}_t). \quad (26)$$

The problem of fitting a Gaussian mixture classifier to a data set is to find those values for the unknown parameters that maximize the likelihood \mathcal{L}_r in (26). For mixture classifiers, a distinction is made between model parameters and free parameters. Model parameters are associated with the mixture model itself (e.g., the mean vectors and covariance matrices of the mixture components). The free parameters are (a) the biases b_k , (b) the γ parameter used in the response rule in (20), and (c) the smoothing term Σ_p . If Σ_p is diagonal, the diagonal elements are denoted by β_i for $i = 1, \dots, d$. If $\Sigma_p = \beta \mathbf{I}$, only one smoothing factor β needs to be estimated. All these parameters can be freely estimated to maximize the likelihood \mathcal{L}_r in (26). For a fully specified mixture classifier, the complete parameter estimation procedure involves the following steps:

1. Set the free parameters (b_k , γ and Σ_p) to some initial values. For example: $b_k = 1/K$ for all k , $\gamma = 1.0$, and $\Sigma_p = \mathbf{0}$.

2. Estimate the mixture model. For Gaussian mixtures, the parameter vector θ can be written as $\theta = \{P(j|C_k), \mu_j, \Sigma_j\}$ for $j = 1, \dots, J$ and $k = 1, \dots, K$. Using the training data \mathcal{T} , estimates of the parameters in θ can be found by maximizing the likelihood function

$$\begin{aligned} \mathcal{L}_t(\mathcal{T} | \theta, \Sigma_p) &= p(\mathcal{T} | \theta, \Sigma_p) \\ &= \prod_{k=1}^K \left\{ \prod_{n=1}^{N_k} p(\mathbf{x}_n | \theta, \Sigma_p) \right\}. \end{aligned} \quad (27)$$

A method for estimating these parameters, based on the EM algorithm is described in appendix B.

3. Using the current mixture model, calculate the posteriors $P(C_k | \mathbf{x}_t)$ for each of the stimuli in \mathcal{R} . Compute the (log of the) likelihood \mathcal{L}_r in (26).

4. Using a hill-climbing algorithm, adjust the free parameters (b_k , γ and Σ_p), and repeat step 2 and 3 until the likelihood \mathcal{L}_r is maximized.

This procedure can be repeated for several mixture classifiers containing a different number of mixture components. For model selection, we simply choose the model containing the smallest number of mixture components, but still fitting the data reasonably well.

It is important to note that the goal of the parameter estimation procedure is to maximize the likelihood \mathcal{L}_r in (26), and not the likelihood \mathcal{L}_t in (27). This has a couple of important implications. For example, suppose a specific mixture model is fitted to the data of different participants who all saw the same stimuli during the training phase. Typically, the responses (r_t) given by these participants during the test phase will differ. As a result of the fitting procedure, the estimated model parameters will not be identical for all participants. This is due to the smoothing term Σ_p , a free parameter which enters the likelihood function \mathcal{L}_t in (27). Furthermore, the best model (having the smallest number of mixture components) may also differ between participants. Because of these reasons, the estimated model parameters may differ dramatically between participants. A second implication is that adding additional mixture components does not necessarily lead to a better fit in terms of the likelihood \mathcal{L}_r . This avoids a common problem with mixture models: finding the best number of mixture components for a given data set.

A concrete example

To illustrate the parameter estimation procedure, several versions of the Gaussian mixture model were fitted to a data set collected from a human categorization experiment reported by McKinley and Nosofsky (1995)⁵. In particular, the models were fitted to data sets from five

⁵I am indebted to R. Nosofsky for supplying the data sets from the McKinley and Nosofsky (1995) study.

individual participants who participated in Experiment 1 (Condition 1) in the McKinley and Nosofsky (1995) study. The category structures used in this condition are shown in Figure 1. Each category was generated from a mixture composed of two bivariate normal densities with equal mixing proportions. In Figure 1, the two mixture components in the lower-right region belong to category C_A , while the mixture components in the upper-left region belong to category C_B . The solid lines are the ‘optimal’ classification boundaries separating the two category mixture densities. The 2-dimensional stimuli used in this experiment were circles varying in radius, with an embedded radial line varying in angle. 4000 stimuli were presented to the participants of the experiment (in a total of five sessions over a contiguous work week). (See McKinley & Nosofsky, 1995, for further details on the experimental procedure). McKinley

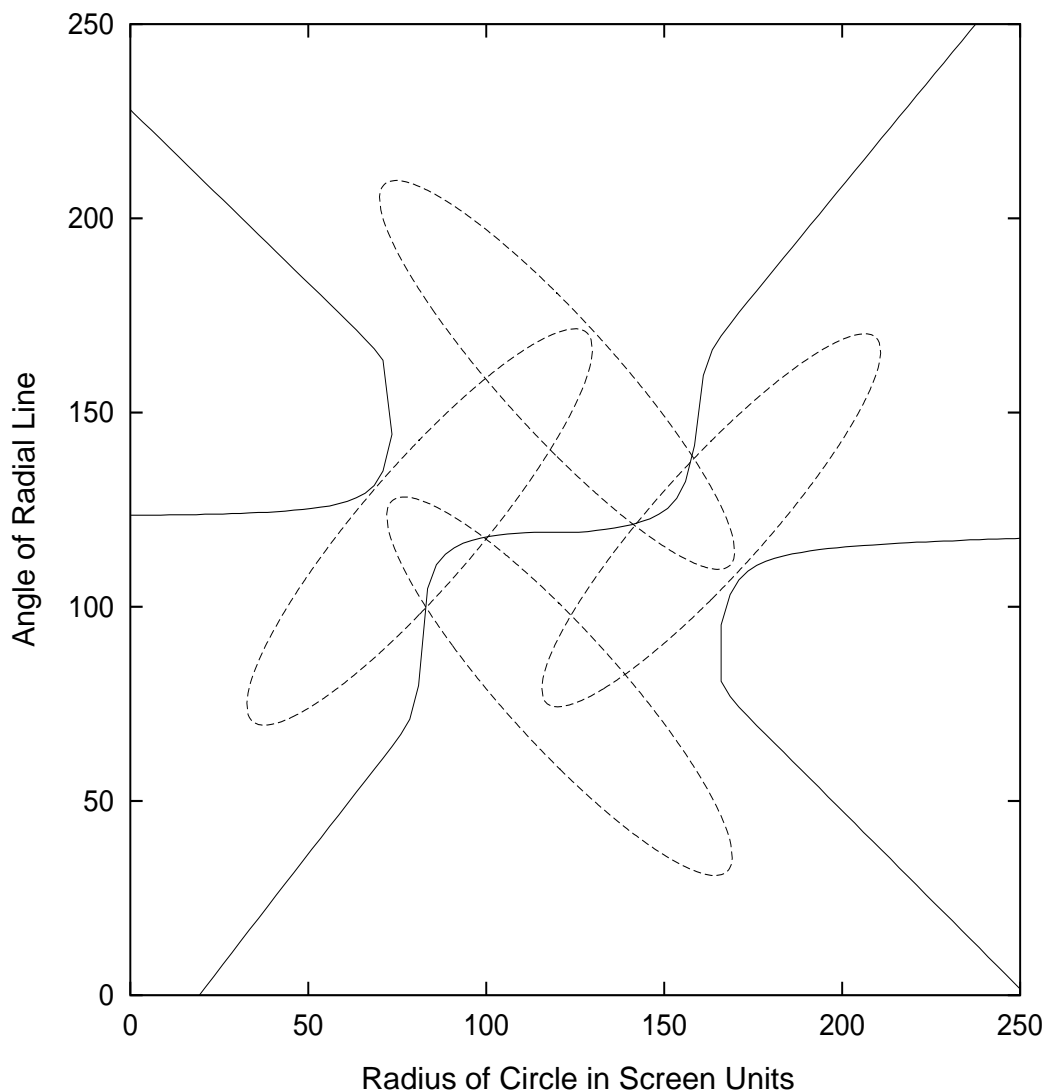


Figure 1. The category structures used in Experiment 1 (Condition 1) in the McKinley and Nosofsky (1995) study. The ellipses correspond to the equiprobability contours for the density functions that compose the two categories. The solid lines are the ‘optimal’ classification boundaries separating the two category mixture densities.

and Nosofsky (1995) have fitted several models to these data sets, among them, (a) the general linear classifier (GLIN), (b) the quadratic classifier (GQC), and (c) a deterministic version of the generalized context model (GCM-D). All these models were fitted to *the last 300 trials* of the experiment for each participant. Specifically, the fits of the models were computed using a hill-climbing algorithm that searched for parameter values that maximized the likelihood of observing the set of responses made by a particular participant during the last 300 trials of the experiment. The same likelihood function was used as in (26), where $T = 300$. To allow for model comparisons, the fits were reported by using a statistical criterion known as Akaike’s information criterion (AIC). The AIC is a modification of the method of maximum likelihood that penalizes models for extra free parameters (Akaike, 1974). The AIC for a model \mathcal{M} is defined as

$$\text{AIC}(\mathcal{M}) = -2 \ln \mathcal{L} + 2\mathcal{P}, \quad (28)$$

where \mathcal{L} denotes the likelihood of the model, and \mathcal{P} denotes the number of free parameters in the model. The best fitting AIC values for three different models are listed in Table 1. The numbers in parentheses refer to the number of free parameters.

Table 1: Model fits of the general linear classifier (GLIN), the general quadratic classifier (GQC) and the deterministic generalized context model (GCM-D) to the last 300 experimental trials. The numbers in this table are taken from Table 4 in McKinley and Nosofsky (1995).

Model	Participants				
	1	2	3	4	5
GLIN(3)	170.5	201.3	268.6	210.7	217.3
GQC(7)	149.9	204.0	265.9	192.8	214.9
GCM-D(6)	118.2	188.5	253.1	114.9	181.9

The following Gaussian mixture classifiers were fitted to the same data set: M-2J, M-3J, M-4J and M-6J. The first 3700 trials were considered as the training set \mathcal{T} , and the last 300 trials as the test set \mathcal{R} . All models had four free parameters: a bias parameter b , a response determinism parameter γ , and the two diagonal elements (β_1, β_2) of the smoothing term Σ_p . The AIC scores for the different models are listed in Table 2. The values for the free parameters for the best fitting models are given in Table 3. To compare the different models, consider

Table 2: Model fits of several Gaussian mixture classifiers to the last 300 trials of Experiment 1 (Condition 1), reported in McKinley and Nosofsky (1995).

Model	Participants				
	1	2	3	4	5
M-2J(4)	163.9	199.6	264.7	204.8	218.4
M-3J(4)	116.2	196.0	256.7	123.5	192.2
M-4J(4)	116.2	194.0	249.3	115.6	192.8
M-6J(4)	116.2	194.0	249.6	115.6	192.9

Figure 2. This figure shows the responses made during the last 300 trials of the best performing participant in this experiment (participant 4) and a graphical representation of the different mixture classifiers. The solid lines are the ‘emergent’ decision bounds (i.e., the points for which $P(C_k | \mathbf{x}) = P(C_l | \mathbf{x})$ for all $k \neq l$). The ellipses are the equidensity contours for the mixture

Table 3: The values of the free parameters for the best fitting mixture classifiers with the smallest number of mixture components.

Parameters	Participants				
	1	2	3	4	5
J	3	4	4	4	3
b_1	0.34	0.62	0.50	0.41	0.62
β_1	328	672	601	76	962
β_2	0	800	655	273	56
γ	2.38	2.33	1.38	2.10	2.03

components. (Note that these equidensity contours do *not* reflect the mixture proportions.) It is clear from Table 2 that mixture classifiers with more than two mixture components fit the data much better than the M-2J model. However, the M-6J model with 6 mixture components shows no advantage over the M-4J model. In fact, the fit of the M-4J model could not be further improved, no matter how many components were added to the mixture model. The best fitting model with the smallest number of mixture components are model M-3J for participant 1 and 5, and model M-4J for all other participants.

When comparing the fits of the mixture classifiers to the fits given in Table 1, several remarks are needed. First, it may seem surprising that the fit of M-2J is worse than the GQC (and more similar to the GLIN) since the representational assumptions of these models are very similar. However, the mixture parameters of the M-2J model (i.e., the mean vectors and covariance matrices) are highly constrained by the stimuli of the training set. This is due to the parameter estimation procedure where the training stimuli (and their true category labels) play an important role in maximizing \mathcal{L}_t in (27). By contrast, the GQC is free to predict any decision boundary that separates the responses observed in the test set, while any information contained in the training set is simply ignored. Due to this estimation method, the GQC has much more flexibility than the M-2J model. Note that fitting a mixture classifier to the test data *only* would be trivial, since a perfect fit would be obtained by setting $J = T$.

The fits of the best fitting mixture classifiers are fairly similar to those of the generalized context model in Table 1, although for participant 5, the context model provides a better fit. Note that in the version of the GCM considered here, similarity computations were based on *all* previously seen exemplars, including those belonging to the test set. In addition, the GCM included a recency parameter, reflecting the possibility that more recently presented exemplars received greater weight in computing summed similarity. The combination of these two factors made it possible for the test set to have a large influence on the category representations. By contrast, the category representations of the mixture classifiers are solely based on the training set. For all these reasons, a direct comparison between the fits of the two models is not possible.

The reduced exemplar model

The Gaussian mixture models presented in the previous section all had two aspects in common: the mixture components had full covariance matrices, and only a relatively small number of mixture components. In a sense, one could say that these models are very similar in spirit to decision bound models or even prototype models. But instead of being strictly parametric, and using one single (normal) distribution to represent a category, the best fitting Gaussian mixture models typically used three or four (common) component distributions to

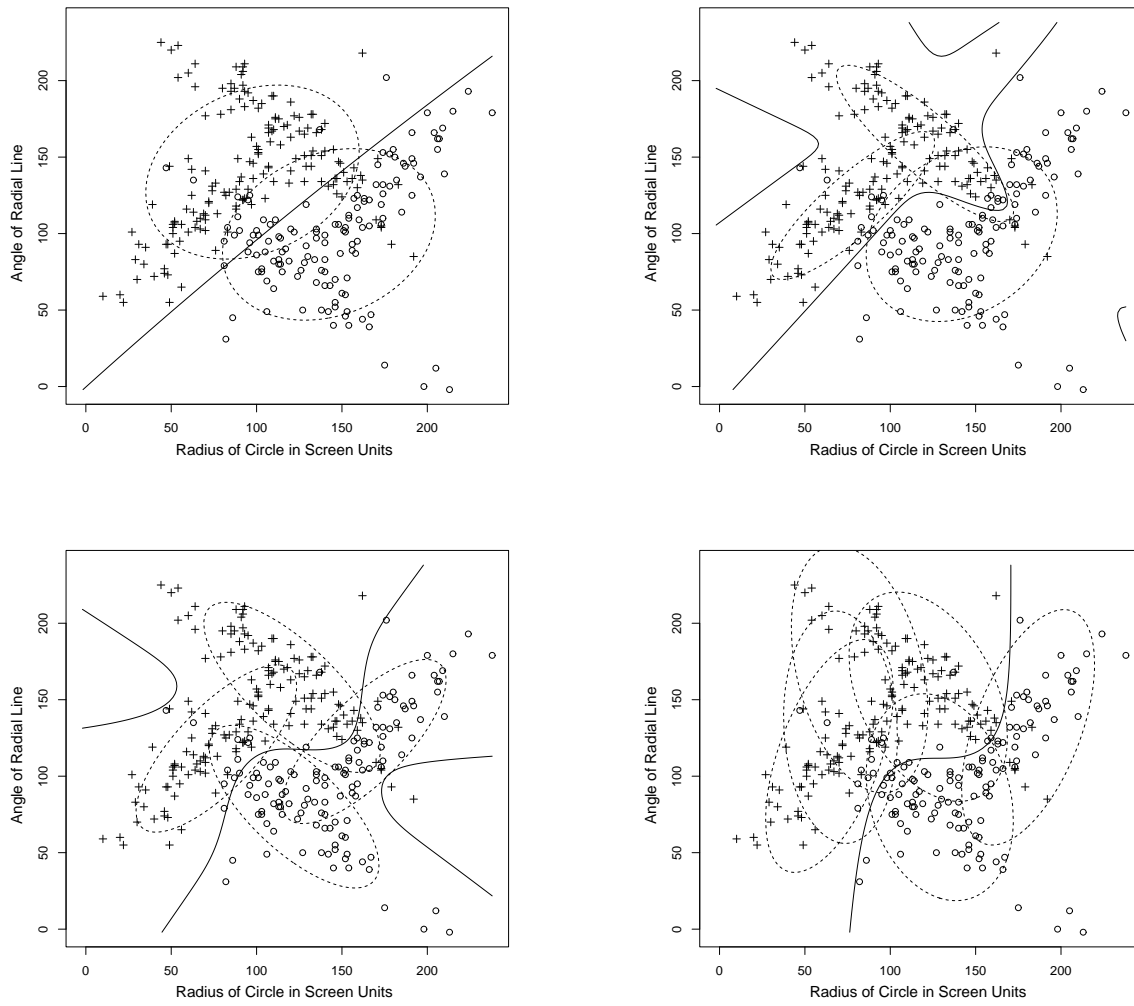


Figure 2. The last 300 responses made by participant 4 together with the ‘emergent’ decision bounds and the equiprobability contours for the mixture components for four different Gaussian mixture classifiers: M-2J (top-left), M-3J (top-right), M-4J (bottom-left) and M-6J (bottom-right). The circles refer to category C_A responses; the crosses refer to category C_B responses.

represent the categories. At the other end of the continuum, exemplar models use N component distributions (see Eq. 16), but their covariance matrices are highly constrained: $\Sigma_n = \Sigma$ for $n = 1, \dots, N$, and Σ is diagonal. Obviously, the high number of mixture components compensates for these constraints. It seems interesting though, to explore the capabilities of a mixture classifier using the same constraints for the covariance matrices as the exemplar models, but where the number of mixture components is smaller than N .

Full versus reduced exemplar storage

As discussed before, the storage requirements of (kernel-based) exemplar models are extremely high: exemplar models need to store every training exemplar together with its category label in memory. Although this may be a reasonable assumption in a laboratory categorization task containing only a limited set of exemplars, it seems a very unrealistic assumption for the representation of natural categories. To cite Myung (1994): “it is hard to imagine that a 70-year-old fisherman would remember every instance of fish that he has seen when attempting to categorize an object as a fish”. Furthermore, categorizing a new stimulus requires a lot of computational effort, since N similarity computations are needed. Even if this computation happens in parallel –as is often suggested by exemplar theorists– the number of computations needed grows directly with the number of stored exemplars. By contrast, for mixture classifiers, the number of computations needed to evaluate new stimuli grows only with the complexity of the mixture model.

However, the real psychological problem with storing a large number of exemplars is not only the possible limited storage size, or the high computational cost for evaluating new stimuli, but rather the strong similarities between many exemplars. To use the above example of the fisherman again, it is easy to imagine that many specific instances of fish are *perceived* as identical to each other, perhaps due to inherent noise in our sensory and perceptual system, or because the observer simply fails to notice any perceptual differences. Even if they are not perceived as identical, one could imagine that many specific instances of fish are *remembered* as identical to each other. Therefore, when a new stimulus needs to be categorized, the actual set of recalled exemplars may not be complete, but *reduced* to include only a limited set of unique exemplars. To distinguish these exemplars from the original full set of exemplars, I call them *exemplar nodes*.

The reduced exemplar model

To formalize the above ideas, the reduced exemplar model was designed to be identical to the generalized context model as defined in (13) and (14), except that the set of stored exemplars $\mathcal{T} = \{\mathbf{x}_n, k_n; n = 1, \dots, N\}$ is replaced by a reduced set of exemplar nodes $\mathcal{N} = \{\mathbf{x}_j, k_j; j = 1, \dots, J\}$. Just like the individual exemplars of the generalized context model, each exemplar node can be represented by a probability distribution $p(\mathbf{x} | j)$ centered on \mathbf{x}_j and with a fixed diagonal covariance matrix Σ . To explain the reduced number of exemplar nodes, consider a new stimulus \mathbf{x}_n which is extremely similar to a stored exemplar node \mathbf{x}_j . Since the perceptual effect of stimulus \mathbf{x}_n can be described as a probability distribution $p(\mathbf{x} | n)$, and the representation of the stored exemplar node \mathbf{x}_j can be described as a probability distribution $p(\mathbf{x} | j)$, the similarity of \mathbf{x}_n to \mathbf{x}_j can be defined as:

$$s(\mathbf{x}_n, \mathbf{x}_j) = \int_{\mathcal{R}_j} p(\mathbf{x} | n) d\mathbf{x}, \quad (29)$$

where \mathcal{R}_j corresponds to the region in perceptual space covered by the density function $p(\mathbf{x} | j)$ (Ashby & Perrin, 1988). Note that (29) can also be interpreted as a measure of the probability

of *confusing* \mathbf{x}_n for \mathbf{x}_j . Next, it is defined that \mathbf{x}_n is *extremely similar* to \mathbf{x}_j if

$$s(\mathbf{x}_n, \mathbf{x}_j) > \xi, \quad (30)$$

where $\xi > 0$ can be interpreted as a confusion threshold parameter. The important point now is that the reduced exemplar model assumes that if (30) holds, the stimulus \mathbf{x}_n will *not* be stored as a new exemplar node, but instead will contribute to the adaptation of the stored exemplar node \mathbf{x}_j . This adaptation means that \mathbf{x}_j is moved a small step into the direction of \mathbf{x}_n . Of course, if $s(\mathbf{x}_n, \mathbf{x}_j) \leq \xi$, \mathbf{x}_j is simply added to the (reduced) set of exemplars stored in memory.

Definition 3 *The reduced exemplar model is a special case of the generalized context model as defined in (11), (12) and (14). However, only a reduced number of exemplar nodes $\mathcal{N} = \{\mathbf{x}_j, k_j; j = 1, \dots, J\}$ where $J \leq N$ is stored in memory. According to the reduced exemplar model, only new stimuli \mathbf{x}_n that are not extremely similar to a stored exemplar node \mathbf{x}_n as defined in (29) and (30) are stored in memory. The class-conditional density function can be written as*

$$p(\mathbf{x} | C_k) = \sum_{n=1}^{J_k} \frac{1}{J_k} p(\mathbf{x} | j), \quad (31)$$

where $p(\mathbf{x} | j)$ are multivariate distributions with mean vector \mathbf{x}_j , and a common diagonal covariance matrix Σ . If $q = r = 2$, $p(\mathbf{x} | j)$ is multivariate normal and the diagonal elements of Σ are given by

$$\frac{1}{2 c^2 w_i},$$

for $i = 1, \dots, d$. If $q = r = 1$, $p(\mathbf{x} | j)$ is multivariate Laplacian and the diagonal elements of Σ are given by

$$\frac{1}{2 c w_i},$$

for $i = 1, \dots, d$.

The value of the confusion threshold parameter ξ determines the number of exemplar nodes that are stored in memory. Clearly, there is a monotonic relationship between ξ and J . If ξ is a relatively small number, many stimuli will be extremely similar to an existing exemplar node, and J will be a small number. By contrast, if ξ is large, many new exemplar nodes are stored, and J will approximate N .

Corollary 1 *The reduced exemplar model becomes equivalent to the generalized context model if ξ is infinitely large.*

In this case, $J = N$. The strong relationship between the reduced exemplar model and the generalized context model is also reflected in the parameter estimation procedure used for fitting these models to a data set. The procedures are identical, except that a preliminary step is needed for the reduced exemplar model to find the coordinates of the exemplar nodes. Once the exemplar nodes have been found, they replace the full set of exemplars used by the GCM, and the free parameters (the category biases b_k , the attention weights w_i , the discriminability parameter c , and a response determinism parameter γ) can be found by maximizing the likelihood function in (26).

Finding the coordinates of the exemplar nodes

Ideally, the confusion threshold parameter ξ should be estimated from another data set, e.g., derived from an identification experiment using the same stimuli and the same participant (Ashby & Perrin, 1988). In a category learning model, this parameter could be used to determine which exemplars are stored in memory, and which are not, using the criterion in (30). In this section, however, we assume that no information about the order in which the exemplars were presented is available. As a result, the set of exemplar nodes has to be reconstructed by assuming that the learning process governed by the confusion threshold parameter ξ has resulted in the storage of J exemplar nodes⁶. Because there is a monotonic relationship between the ξ and J , we can use J (instead of ξ) to specify a particular model. The problem is to find the coordinates of the J exemplar nodes, given the training set $\mathcal{T} = \{\mathbf{x}_n, k_n; n = 1 \dots, N\}$ in such a way that they reflect the assumptions of the reduced exemplar model. The coordinates of an exemplar node should reflect the central tendency of a set of previously seen exemplars for which (30) holds. By definition, this set contains all exemplars for which

$$s(\mathbf{x}, \mathbf{x}_j) > s(\mathbf{x}, \mathbf{x}_k) \quad (32)$$

for all $j \neq k$. To compute these similarity comparisons, it is necessary to evaluate the integral in (29). However, since at this point, the common covariance matrix Σ of the exemplar distributions is unknown (as it is part of the parameter estimation procedure itself), this is not possible. Hence, the following criterion is used instead:

$$d(\mathbf{x}, \mathbf{x}_j) < d(\mathbf{x}, \mathbf{x}_k) \quad (33)$$

for all $j \neq k$ where $d(\mathbf{x}, \mathbf{x}_j)$ denotes the Euclidean distance from \mathbf{x} to \mathbf{x}_j . The problem of finding the coordinates of the J exemplar nodes is essentially a vector quantization (or clustering) problem (Gersho & Gray, 1992). In the vector quantization literature, a set of N vectors is reduced to a set of J vectors ($J < N$) in one of several ways. The J vectors are typically called *reference vectors* or *codebook vectors* or even *prototype vectors*. The complete set of the J reference vectors is called the *codebook*. There exist many different algorithms for finding a suitable set of reference vectors from a data set, but a simple and widely known method is the LBG (or generalized Lloyd) algorithm (Linde, Buzo, & Gray, 1980; Lloyd, 1982). The LBG algorithm tries to minimize the mean squared distance between a vector \mathbf{x}_n and its nearest reference vector \mathbf{x}_j :

$$\frac{1}{N} \sum_{n=1}^N \left[\min_{j=1}^J d(\mathbf{x}_n, \mathbf{x}_j)^2 \right]. \quad (34)$$

In brief, the LBG algorithm can be described as follows:

1. Initialize the J reference vectors (by setting them to a random position)
2. Divide the N data points in J disjoint subsets \mathcal{X}_j according to the criterion in (33).
3. Move the reference vector \mathbf{x}_j to the centroid (the sample mean) of the corresponding subset \mathcal{X}_j .
4. Repeat steps 2 and 3 until the \mathbf{x}_j change no more.

⁶The method described in this section to find the coordinates of the exemplar nodes is clearly asymptotic: an iterative algorithm is used to estimate the final coordinates of the exemplar nodes. However, it is possible to construct a learning version of the reduced exemplar model where the coordinates of the exemplar nodes are learned in an incremental fashion (needing only one pass through the training exemplars). This algorithm is presented elsewhere (Rosseel, 1998).

Note that a recursive version of this algorithm is equivalent to the k -means clustering algorithm (MacQueen, 1967). An interesting extension of LBG is the LBG-U algorithm, developed by Fritzke (1997). Basically, the LBG-U algorithm consists of repeated runs of the LBG algorithm. Each time LBG has converged, a measure of utility is assigned to each reference vector \mathbf{x}_j . The vector with minimum utility is moved to a new location, and LBG is run again on the resulting modified codebook. This process is repeated until the error in (34) doesn't change any more (see Fritzke, 1997, for details). Since this method provided more consistent results, I have adopted this method to find the coordinates of the exemplar nodes. In practice, the training set $\mathcal{T} = \{\mathbf{x}_n, k_n; n = 1 \dots, N\}$ is split into K subsets: one for each category. On each subset, the LBG-U algorithm is run with a fixed number (J_k) of reference vectors. The set of reduced exemplar nodes can be summarized as $\mathcal{N} = \{\mathbf{x}_j, k_j; j = 1 \dots, J\}$, where k_j denotes the category label associated with exemplar node \mathbf{x}_j . Typically, the same number of exemplar nodes is used for each category.

Since the reduced exemplar model can be regarded as a special version of a mixture classifier, it might have been possible to use the EM algorithm, instead of LBG-U, to find the coordinates of \mathbf{x}_j (i.e., the means of the mixture components). Here, both the mixture proportions and the covariance matrices of the mixture components are fixed, and need not to be estimated. However, in general, the (least squares) estimates of the LBG-U algorithm will differ from the (maximum-likelihood) estimates of the EM algorithm. The EM algorithm can be interpreted as a *soft* clustering method, since each vector is assigned to a mixture component with a certain degree of membership. By contrast, the LBG (and LBG-U) algorithm is a *hard* clustering method, where each vector is assigned to one exemplar node, or in other words, there is only one *winner*. Consequently, hard clustering algorithms are often regarded as *winner-take-all* (WTA) algorithms. Interestingly, it is easy to construct a "winner-take-all" version of the mixture classifiers where $p(\mathbf{x} | C_k)$ is now approximated by the largest term in the summation:

$$p(\mathbf{x} | C_k) = \max_{j=1}^{J_k} p(\mathbf{x} | j). \quad (35)$$

If this WTA approximation of the mixture classifier is used, and if it is further assumed that all mixing proportions are equal ($P(j | C_k) = 1/J_k$) and the common variance of $\Sigma = \sigma^2 \mathbf{I}$ is known, then it can be shown that the maximum likelihood estimates of the mean vectors \mathbf{x}_j are identical to the least squares estimates of these mean vectors (Nowlan, 1991; Duda & Hart, 1973). Therefore, using the EM algorithm with the above constraints for finding the coordinates of the exemplar nodes generally leads to the same results as the LBG-U algorithm.

A practical illustration

Again, consider the data sets of Experiment 1 (Condition 1) of the McKinley and Nosofsky (1995) study. Here, several versions of the reduced exemplar model are fitted to these data sets. First, however, a slightly different version of the generalized context model is fitted to the same data. The version of the GCM discussed in McKinley and Nosofsky (1995) contained six free parameters. The version used here contains only four free parameters: a bias (b) and a weight (w) parameter, the discriminability parameter c , and the response determinism parameter γ . Also, for computing the similarities in (12), not *all* previously seen exemplars were used, but only those that belonged to the training set in $\mathcal{T} = \{\mathbf{x}_n, k_n; n = 1 \dots, N\}$ which contained 3700 stimuli. This makes it easier to compare the GCM with the reduced exemplar models, where a strict distinction between training set and test set is necessary. The AIC scores for this version of the GCM are shown at the bottom of Table 4. Although the fits differ slightly from the ones in Table 1, the general results are comparable. Several versions of the reduced exemplar model

were fitted to the same data. The total number of exemplar nodes varied from 2 to 200. Half of these exemplar nodes belonged to one category, the others to the contrasting category. The four free parameters are identical to the ones used by the GCM. The AIC scores for these models are shown in Table 4. In this table, REX- x denotes a reduced exemplar model with x exemplar nodes. Using this notation, the GCM could have been denoted as REX-3700. The general trend

Table 4: Model fits of several versions of the reduced exemplar model for the last 300 trials of Experiment 1 (Condition 1), reported in McKinley and Nosofsky (1995).

Model	Participants				
	1	2	3	4	5
REX-2	173.2	215.7	275.9	205.5	226.3
REX-4	143.4	208.6	268.3	162.9	226.8
REX-6	141.3	201.3	253.4	133.0	194.5
REX-8	150.7	208.9	255.5	150.6	184.6
REX-10	135.0	196.7	256.1	141.9	190.2
REX-12	131.4	196.0	253.8	111.2	183.5
REX-14	130.5	195.5	250.9	108.0	193.8
REX-16	126.9	195.9	250.1	97.5	185.4
REX-18	122.4	194.9	250.9	96.4	185.6
REX-20	125.8	194.4	249.9	105.7	183.0
REX-40	123.2	194.1	246.7	111.4	183.8
REX-80	121.2	193.2	248.7	109.6	179.3
REX-200	123.8	193.1	249.2	111.3	182.5
GCM-D(4)	121.2	194.3	249.3	111.5	180.0

shown in Table 4 is that adding more exemplar nodes leads to better fits. However, once a sufficient number of exemplar nodes is used, adding more exemplar nodes does not necessarily lead to a substantial improvement in fit. At least for this particular data set, it seems that only a fairly small number of exemplar nodes (say, 20, that is 10 for each category) is needed, to get a reasonable fit. Compared to the number of exemplars used by the GCM (3700), this represents a dramatic reduction. Nevertheless, the results of the REX-18 and REX-20 models are comparable with those of a full exemplar model. Figure 3 illustrates how the exemplar nodes are distributed in the stimulus space for the REX-4, REX-10, REX-20 and REX-200 model (for participant 4). Each panel shows the coordinates of the exemplar nodes as computed by the LBG-U algorithm. Figure 4 shows the last 300 responses of participant 4 together with the emergent decision bounds (or equivocality contours) for the corresponding REX models.

Discussion

Parametric and nonparametric models of categorization have played a dominant role in the categorization literature for many years. Both approaches have their merits and their limitations. Parametric models, such as prototype models and decision bound models, are very economic both in terms of storage requirement and the number of computations needed to evaluate new stimuli. Unfortunately, parametric models are often not flexible enough to mimic the ability of humans to learn complex category distributions. Indeed, recent studies have suggested that human categorization is nonparametric, rather than parametric (Ashby & Waldron, 1999; McKinley & Nosofsky, 1995). In those studies, nonparametric models provided significantly better accounts

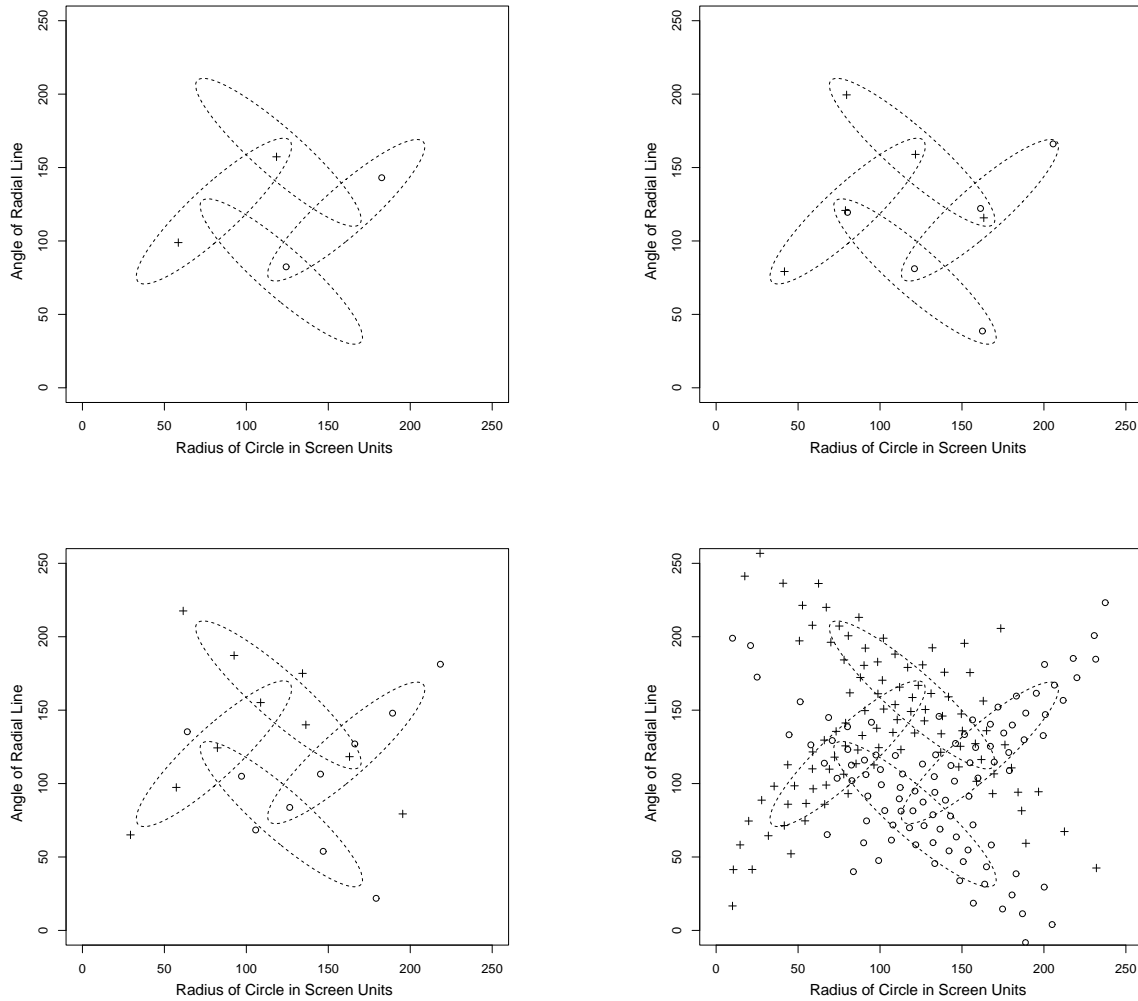


Figure 3. The coordinates of the exemplar nodes as computed by the LBG-U algorithm. The number of exemplar nodes is 4 (top-left), 10 (top-right), 20 (bottom-left) and 200 (bottom-right). The circles refer to category C_A exemplar nodes; the crosses refer to category C_B exemplar nodes.

of the data than the parametric alternatives. An important class of nonparametric classifiers are the exemplar models. A typical characteristic of these exemplar models (and all kernel-based models), is that both the storage requirements and the number of (similarity) computations needed to evaluate new stimuli, grow linearly with the number of previously seen exemplars, and not with the complexity of the category distributions. However, a more problematic aspect of exemplar models is that they essentially predict that with enough training, observers are capable of learning any category distribution, no matter how complex.

In order to combine the advantages of both parametric and nonparametric models, it seems reasonable to investigate semi-parametric models which are more flexible than strictly parametric models, but where the storage and computational requirements of the model only grow with the complexity of the category distributions. In the last decade, several models have been proposed that could be considered as semi-parametric. This includes Anderson's (1991) rational model, the covering version of Kruschke's (1992) ALCOVE model, and the striatal

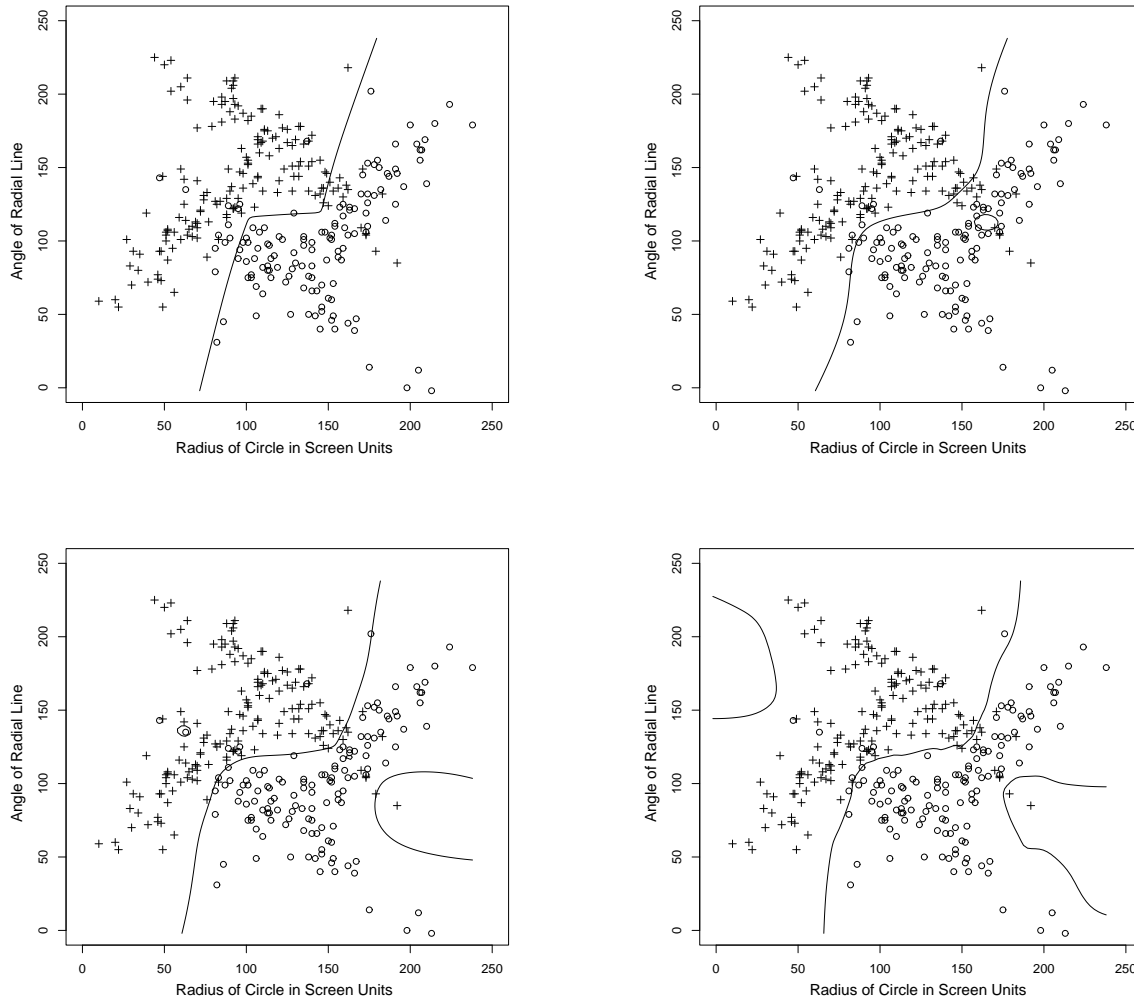


Figure 4. The last 300 responses of participant 4, together with the emergent decision bounds for several versions of the reduced exemplar model: REX-4 (top-left), REX-10 (top-right), REX-20 (bottom-left) and REX-200 (bottom-right).

pattern classifier (SPC) (Ashby & Waldron, 1999). Although these models differ in many ways, they all share the property that a category is represented by a (typically small) number of subgroups (or clusters, or hidden nodes, or striatal units). This gives them much more flexibility than strictly parametric classifiers. However, the number of subgroups is typically much smaller than the number of exemplars. Therefore, these semi-parametric models may be better suited (than kernel-based classifiers) to describe the suboptimal response patterns often observed in data collected from human categorization experiments.

In this article, I proposed a family of semi-parametric classifiers where categories are represented by a finite mixture distribution. The advantage of these mixture classifiers is that they can describe a large number of (new and existing) categorization models within a common framework. For instance, I showed that both decision bound models and the generalized context model can be interpreted as two special cases of the generic mixture framework. And although no formal proof has been given in this paper, it is possible to show that other parametric and

nonparametric models, including the prototype model (Reed, 1972), the array model (Estes, 1986) and the context model (Medin & Schaffer, 1978), can be interpreted as special cases of the mixture classifier framework. Furthermore, the category representations of several existing semi-parametric models (including the rational model, the covering version of ALCOVE, and the SPC) seem to be well captured within the generic framework of mixture classifiers.

To illustrate how new models can be derived from the generic mixture framework, I considered two approaches. In the first approach, categories were represented by a Gaussian mixture model with a small number of mixture components, but with full covariance matrices. These models can be regarded as a straightforward extension of parametric models where only one single distribution is used to represent a category. Several versions of the Gaussian mixture classifier were fitted to a data set reported by McKinley and Nosofsky (1995). The results showed that by allowing for more than one mixture component per category, the models fitted the data significantly better than did the parametric models. In fact, the results of the best fitting models were comparable to the fits of the (nonparametric) exemplar model. As a second approach, I proposed the reduced exemplar model, which is almost equivalent to the generalized context model, except that only a reduced set of exemplar nodes is stored in memory (instead of all exemplars). Nevertheless, when fitted to the McKinley and Nosofsky (1995) data, the fits of the reduced exemplar models suggested that (at least for large categories with highly similar stimuli) only a relatively small number of exemplar nodes was needed to get a reasonable fit of the data.

Indeed, there seems to be some converging evidence that the category representations learned by observers during a categorization task are well captured by a semi-parametric model. It is worth commenting, however, that the different models still sharply disagree about the underlying learning processes that are used by observers to construct these (semi-parametric) representations. The rational model, for instance, proposed a Bayesian type of learning, while ALCOVE has used error-correction learning. The SPC currently lacks a (formal) learning scheme, but according to Ashby and Waldron (1999), procedural learning should be used. Finally, Rosseel (1998, 1997) has proposed a category learning model called *the growing mixture model*. In this model, a combination of supervised and unsupervised learning is used to construct a mixture representation of the categories. No doubt, a lot of future research will involve the comparison of different category learning models, all sharing a common semi-parametric representation of the categories.

References

- Akaike, H. (1974). A new look at statistical model identification. *IEEE Transactions on Automatic Control*, *19*, 716–723.
- Anderson, J. R. (1991). The adaptive nature of human categorization. *Psychological Review*, *98*, 409–429.
- Ashby, F. G. (1992). Multidimensional models of categorization. In F. G. Ashby (Ed.), *Multidimensional models of perception and cognition* (pp. 449–483). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Ashby, F. G., & Alfonso-Reese, L. A. (1995). Categorization as probability density estimation. *Journal of Mathematical Psychology*, *39*, 216–233.
- Ashby, F. G., & Lee, W. W. (1991). Predicting similarity and categorization from identification. *Journal of Experimental Psychology: General*, *120*, 150–172.
- Ashby, F. G., & Maddox, W. T. (1992). Complex decision rules in categorization: Contrasting novice and experienced performance. *Journal of Experimental Psychology: Human Perception and Performance*, *16*, 598–612.

- Ashby, F. G., & Maddox, W. T. (1993). Relations between prototype, exemplar, and decision bound models of categorization. *Journal of Mathematical Psychology*, *37*, 372–400.
- Ashby, F. G., & Perrin, N. A. (1988). Toward a unified theory of similarity and recognition. *Psychological Review*, *95*, 124–150.
- Ashby, F. G., & Townsend, J. T. (1986). Varieties of perceptual independence. *Psychological Review*, *93*, 154–179.
- Ashby, F. G., & Waldron, E. M. (1999). On the nature of implicit categorization. *Psychonomic Bulletin & Review*, *6*, 363–378.
- Ashby, F. G., Waldron, E. M., Lee, W. W., & Berkman, A. (in press). Suboptimality in human categorization and identification. *Journal of Experimental Psychology: General*.
- Banfield, J. D., & Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, *49*, 803–821.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society series B*, *39*, 1–38.
- De Soete, G. (1993). Using latent class analysis in categorization research. In I. Van Mechelen, J. Hampton, R. S. Michalski, & P. Theuns (Eds.), *Categories and concepts* (Vol. 29, pp. 309–330). New York: Academic Press.
- Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: Wiley.
- Estes, W. K. (1986). Array models for category learning. *Cognitive Psychology*, *18*, 500–549.
- Fritzke, B. (1997). The LBG-U method for vector quantization – an improvement over LBG inspired from neural networks. *Neural Processing Letters*, *5*(1), 35–45.
- Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (Second ed.). Boston: Academic Press. ([First edition, 1972])
- Furman, W., & Lindsay, B. (1994). Testing for the number of components in a mixture of normal distributions using moment estimators. *Computational Statistics and Data Analysis*, *17*, 473–492.
- Gersho, A., & Gray, R. M. (1992). *Vector quantization and signal compression*. Boston: Kluwer Academic Publishers.
- Ghahramani, Z., & Jordan, M. I. (1994). Supervised learning from incomplete data via an em approach. In J. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems 6* (pp. 255–262). San Mateo, CA: Morgan Kaufmann Publishers.
- Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, *6*, 181–214.
- Kruschke, J. K. (1992). Alcové: An exemplar-based connectionist model of category learning. *Psychological Review*, *99*, 22–44.
- Linde, Y., Buzo, A., & Gray, R. M. (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communication*, *COM-28*, 84–95.
- Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*, 128–137.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam & J. Neyman (Eds.), *Proceedings of the fifth berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297). Berkeley, CA: University of California Press.
- McKinley, S. C., & Nosofsky, R. M. (1995). Investigations of exemplar and decision bound models in large, ill-defined category structures. *Journal of Experimental Psychology: Human Perception and Performance*, *21*, 128–148.

- McLachlan, G. J., & Basford, K. E. (1988). *Mixture models: Inference and applications to clustering*. New York: Marcel Dekker.
- Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, *85*, 207–238.
- Myung, I. J. (1994). Maximum entropy interpretation of decision bound models and context models of categorization. *Journal of Mathematical Psychology*, *38*, 335–365.
- Nosofsky, R. M. (1984). Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *10*, 104–114.
- Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General*, *115*, 39–57.
- Nosofsky, R. M. (1988). On exemplar-based exemplar representations: Reply to ennis (1988). *Journal of Experimental Psychology: General*, *117*, 412–414.
- Nosofsky, R. M. (1992). Exemplar-based approach to relating categorization, identification, and recognition. In F. G. Ashby (Ed.), *Multidimensional models of perception and cognition* (pp. 363–393). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Nowlan, S. (1991). *Soft competitive adaptation: Neural network learning algorithms based on fitting statistical mixtures*. Unpublished doctoral dissertation, School of Computer Science, Carnegie Mellon University.
- Ormonet, D., & Tresp, V. (1998). Averaging, maximum penalized likelihood and Bayesian estimation for improving Gaussian mixture probability density estimates. *IEEE Transactions on Neural Networks*, *9*(4), 639–650.
- Parzen, E. (1962). On the estimation of a probability density function and mode. *Annals of Mathematical Statistics*, *33*, 1065–1076.
- Reed, S. K. (1972). Pattern recognition and categorization. *Cognitive Psychology*, *3*, 382–407.
- Rosseeel, Y. (1997). The growing mixture model: a dynamical framework for modeling the nonverbal categorization system. In *Proceedings of the 30th annual meeting of the society for mathematical psychology*. Bloomington.
- Rosseeel, Y. (1998). *Categorization as probability density estimation: statistical and computational models of categorization and category learning*. Unpublished doctoral dissertation, Department of Experimental Psychology, University of Ghent.
- Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. London: Chapman & Hall.
- Titterton, D. M., Smith, A. F. M., & Makov, U. E. (1985). *Statistical analysis of finite mixture distributions*. Chichester: Wiley.
- Xu, L., & Jordan, M. I. (1996). On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, *8*, 129–151.

Appendix A

This appendix describes the standard EM algorithm for estimating the parameters of a Gaussian mixture model. The unknown parameters are summarized in the parameter vector $\boldsymbol{\theta} = \{P(j), \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}$ where $j = 1, \dots, J$. The problem can be described as follows: given an unlabeled data set $\mathcal{X} = \{\mathbf{x}_n; n = 1, \dots, N\}$, estimate the unknown parameters in $\boldsymbol{\theta}$ by maximizing the likelihood function

$$\begin{aligned} \mathcal{L}(\mathcal{X}|\boldsymbol{\theta}) &= \prod_{n=1}^N p(\mathbf{x}_n|\boldsymbol{\theta}) \\ &= \prod_{n=1}^N \left\{ \sum_{j=1}^J P(j) \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \right\}. \end{aligned} \quad (36)$$

A well established maximum likelihood algorithm for estimating these parameters is based on the *Expectation Maximization* (EM) algorithm (Dempster, Laird, & Rubin, 1977). The EM algorithm is an iterative procedure for finding maximum likelihood parameter estimates when the data set can be viewed as *incomplete*. In the context of mixture models, the *missing* values are the component labels j to which the stimulus vectors belong. The EM algorithm tackles this problem by first estimating the posterior probabilities $P(\mathbf{x}_n|j)$ for every n and for every j using the current estimates of the parameters in $\boldsymbol{\theta}$, and then uses these posteriors to re-estimate the parameters by maximizing the (complete) likelihood function. The first step is known as the **E-step**, while the latter is called the **M-step**. The EM algorithm starts with some initial guess $\boldsymbol{\theta}^{(0)}$ of the parameter values and then repeatedly applies the **E-step** and the **M-step** to generate successively better parameter estimates $(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots)$. The complete EM algorithm for a mixture of Gaussians model can be described as follows:

1. Set the iteration counter $t = 0$. Initialize the parameter set $\boldsymbol{\theta}^{(0)} = \{P(j)^{(0)}, \boldsymbol{\mu}_j^{(0)}, \boldsymbol{\Sigma}_j^{(0)}\}$.
2. **E-step**: for every data point in \mathcal{X} and for every mixture component j , compute the posterior probability $P(j|\mathbf{x}_n)$ as follows:

$$P(j|\mathbf{x}_n) = \frac{P(j)^{(t)} \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Sigma}_j^{(t)})}{\sum_{j=1}^J P(j)^{(t)} \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Sigma}_j^{(t)})}, \quad (37)$$

where $\boldsymbol{\theta}^{(t)} = \{P(j)^{(t)}, \boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Sigma}_j^{(t)}\}$ is the set of parameters after the t -th iteration.

3. **M-step**: re-estimate the parameters using the data set weighted by the posterior probabilities $P(j|\mathbf{x}_n)$:

$$P(j)^{(t+1)} = \frac{1}{N} \sum_{n=1}^N P(j|\mathbf{x}_n) \quad (38)$$

$$\boldsymbol{\mu}_j^{(t+1)} = \frac{\sum_{n=1}^N P(j|\mathbf{x}_n) \mathbf{x}_n}{\sum_{n=1}^N P(j|\mathbf{x}_n)} \quad (39)$$

$$\boldsymbol{\Sigma}_j^{(t+1)} = \frac{\sum_{n=1}^N P(j|\mathbf{x}_n) (\mathbf{x}_n - \boldsymbol{\mu}_j^{(t+1)})(\mathbf{x}_n - \boldsymbol{\mu}_j^{(t+1)})^T}{\sum_{n=1}^N P(j|\mathbf{x}_n)}. \quad (40)$$

4. Until the algorithm has converged, go to step 2.

The above algorithm will always converge to at least a local maximum of the likelihood function (36) after a finite, typically small number of iterations (Xu & Jordan, 1996).

Appendix B

The algorithm described in Appendix A assumes that the datapoints are unlabeled. That is, they all belong to the same class (or no class information is available). If, on the other hand, the data belong to different categories, and the category labels are available for all datapoints, we can use this information to build a mixture classifier. In this appendix, the standard EM algorithm is adapted to estimate the parameters of a Gaussian mixture classifier (see Jordan & Jacobs, 1994). The unknown parameters are now $\theta = \{P(j | C_k), \mu_j, \Sigma_j\}$ for $j = 1, \dots, J$ and $k = 1, \dots, K$. The likelihood function can be written as

$$\begin{aligned} \mathcal{L}(\mathcal{X}_l | \theta, \Sigma_p) &= p(\mathcal{X}_l | \theta, \Sigma_p) \\ &= \prod_{k=1}^K \left\{ \prod_{n=1}^{N_k} p(\mathbf{x}_n | \theta, \Sigma_p) \right\}. \end{aligned} \quad (41)$$

where the data is now the labeled data set $\mathcal{X}_l = \{\mathbf{x}_n, k_n; n = 1 \dots, N\}$ where k_n denotes the category label of stimulus \mathbf{x}_n . As can be seen in (41), the estimates also depend on a given covariance matrix Σ_p . This matrix can be considered as a *regularization* term which is sometimes used in the EM algorithm for smoothing the mixture distribution (Ormonoit & Tresp, 1998; Ghahramani & Jordan, 1994). The regularization scheme consists of adding a constant regularization matrix (Σ_p) to each covariance matrix Σ_j in each iteration of the EM algorithm. A complete overview of this regularized version of the EM algorithm for estimating the parameters of Gaussian mixture classifiers can be described as follows:

1. Set the iteration counter $t = 0$. Initialize the parameter set $\theta^{(0)} = \{P(j | C_k)^{(0)}, \mu_j^{(0)}, \Sigma_j^{(0)}\}$.
2. **E-step:** for every stimulus vector in \mathcal{X}_l and for every mixture component j , compute the posterior probabilities $P(j | \mathbf{x}_n)$ as follows:

$$P(j | \mathbf{x}_n) = \frac{P(j | C_{k_n})^{(t)} \mathcal{N}(\mathbf{x}_n; \mu_j^{(t)}, \Sigma_j^{(t)})}{\sum_{j=1}^J P(j | C_{k_n})^{(t)} \mathcal{N}(\mathbf{x}_n; \mu_j^{(t)}, \Sigma_j^{(t)})} \quad (42)$$

where $\theta^{(t)} = \{P(j | C_k)^{(t)}, \mu_j^{(t)}, \Sigma_j^{(t)}\}$ is the set of parameters after the t -th iteration. C_{k_n} is the class label of \mathbf{x}_n .

3. **M-step:** re-estimate the parameters using the data set weighted by the posterior probabilities $P(j | \mathbf{x}_n)$:

$$P(j | C_{k_n})^{(t+1)} = \frac{1}{N_k} \sum_{n=1}^{N_k} P(j | \mathbf{x}_n) \quad (43)$$

$$\mu_j^{(t+1)} = \frac{\sum_{n=1}^N P(j | \mathbf{x}_n) \mathbf{x}_n}{\sum_{n=1}^N P(j | \mathbf{x}_n)} \quad (44)$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{n=1}^N P(j | \mathbf{x}_n) (\mathbf{x}_n - \mu_j^{(t+1)}) (\mathbf{x}_n - \mu_j^{(t+1)})^T}{\sum_{n=1}^N P(j | \mathbf{x}_n)}. \quad (45)$$

4. Regularize the covariance matrix $\Sigma_j^{(t+1)}$ by adding the regularizer Σ_p :

$$\Sigma_j^{(t+1)} = \Sigma_j^{(t+1)} + \Sigma_p. \quad (46)$$

5. Until the algorithm has converged, go to step 2.

Notice carefully that in (44) and (45), the sum is taken over all data points ($n = 1, \dots, N$), while in (43) only the category k_n data points are used ($n = 1, \dots, N_k$). Similarly, only stimuli that belong to the same class are used to compute the posteriors $P(j | \mathbf{x}_n)$ in (42). For all fits reported in this article, the parameters of the Gaussian mixture classifiers were initialized as follows: (a) the LBG algorithm was used to initialize the mean vectors $\boldsymbol{\mu}_j$, (b) the covariance matrices were initialized as $\boldsymbol{\Sigma}_j = \sigma^2 \mathbf{I}$ where σ was set to the distance to the nearest other component center, and finally (c) the class-conditional mixture proportions $P(j | C_k)$ were all set to $1/J$.