Review of MR image segmentation techniques using pattern recognition

J. C. Bezdek
Division of Computer Science, University of West Florida, Pensacola, Florida 32514

L. O. Hall
Department of Computer Science and Engineering, University of South Florida, Tampa, Florida 33647

L. P. Clarke
Department of Radiology and H. Lee Moffitt Cancer and Research Center, University of South Florida, Tampa, Florida 33647

(Received 28 September 1992; accepted for publication 14 April 1993)

TABLE OF CONTENTS

I. INTRODUCTION ........................................ 1033
A. Clinical rationale .................................... 1033
B. Scope of the review .................................. 1034

II. PATTERN RECOGNITION: PROBLEMS AND MODELS ........ 1034
A. Fuzzy models: Nonstatistical uncertainty .......... 1034
B. Pattern recognition: Problems and approaches ...... 1035
C. Statistical pattern recognition: The mixture model .... 1037
D. Computational neutral networks ...................... 1038
E. MR image data for classifier design and cluster analysis ... 1038

III. SUPERVISED (CLASSIFIER) METHODS IN MR IMAGE SEGMENTATION .......... 1040

A. Parametric designs: Bayes classifiers with labeled maximum likelihood estimators ...................... 1040
B. Nonparametric designs: The k-nearest neighbor rule ... 1041
C. Feed forward neural network methods ................ 1042

IV. UNSUPERVISED (CLUSTERING) METHODS IN MR IMAGE SEGMENTATION .......... 1044
A. Parametric designs: Bayes classifiers with unlabeled maximum likelihood estimators .................... 1044
B. The hard and fuzzy c-means algorithms; FCM, AFCM, and SFCM .......... 1044
C. Other approaches ...................................... 1046

V. SUMMARY .............................................. 1046

I. INTRODUCTION

A. Clinical rationale

Magnetic resonance imaging (MRI) systems measure spatial distributions of several distinct tissue-related parameters such as relaxation times and proton density. Analogous to LANDSAT imagery, MRI measurements are collections of features (that is, numerical characteristics) from spatial arrays that are aggregated into multidimensional data (from a single anatomical slice). Measured intensities at p different “frequencies” at each spatial (pixel) location can be used as a basis for algorithmically formed tissue clusters in p-dimensional feature space, which is the collection of all possible aggregates of the measured features. As p is increased using different tailored pulse sequences, higher-dimensional feature spaces may yield improved image segmentation when compared to visual interpretation or gray-scale segmentation of single MR images. For example, 2-D or 3-D Fourier-transformed sequences that either provide additional features for segmentation, improve signal to noise ratio, or allow rapid calculation of MR parameters could, in principle, be used. These include trajectory techniques, rapid acquisition with relaxation enhancement, fast magnetization preparation techniques, or proton perfusion or diffusion imaging. Pattern recognition methods, when applied to p-dimensional MRI data, should improve the level of confidence in image segmentation compared to gray-scale approaches using single images. 1-7

Multidimensional segmentation has shown promise in the enhancement of image contrast between normal tissue distributions. This work has been directed toward more objective measures of normal tissue distribution as diagnostic indicators of disease processes. Similarly, multiparameter segmentation methods have been proposed to improve the boundary definition between pathology, edema, and scar tissue that have very similar MR relaxation parameters and that generally present a more difficult segmentation problem than that required for segmentation of normal tissue distributions. 1,6,5,11,12 Their application to 3-D image (voxel) data would potentially allow improved 3-D display of tumor tissue with a more objective criterion for the removal of surrounding normal tissue. 1,6,11,12,17-20 Similarly, supervised and unsupervised (automated) pattern recognition methods have the potential to improve cancer tumor boundary definition as required for 3-D conformal radiation therapy treatment planning or electronic surgery simulation. 1,6,11,12,17-20 MR technology advances in rapid 3-D MR imaging methods have also stimulated interest in intelligent image fusion of multiple 3-D data sets using pattern recognition methods to increase the efficiency of diagnosis, provided appropriate image registration meth-
ods are applied.\textsuperscript{15,20} Finally, the use of segmentation methods as applied to multimodality imaging such as MRI, PET, or SPECT is currently being considered as a means of improved differentiation of tumor and normal tissues.\textsuperscript{20} Multimodality approaches require careful attention to both image registration issues and the requirement for image resolution restoration techniques, particularly for the degraded images obtained by PET or SPECT detectors.\textsuperscript{21–23}

B. Scope of this review

The success of MRI segmentation involves two broadly interrelated fields of research: (a) MRI and radio frequency (RF) coil sensor characteristics, optimization of pulse sequences and related 2-D and 3-D Fourier-transformed imaging protocols, image intensity and artifact corrections, registration corrections, and noise reduction techniques; and (b) the development and optimization of pattern recognition methods for segmentation. This review addresses pattern recognition methods that have the best potential for MRI segmentation, with particular emphasis on comparison between \textit{supervised} and \textit{unsupervised} methods, since these methods have not been extensively reported in the medical imaging literature.\textsuperscript{1,11} An attempt is made to orient readers about both the physical basis of each method and its theoretical justification, since the latter is strongly application-dependent, particularly in the field of medical imaging. Supervised techniques include computational neural networks (CNNs) as they are applied to MRI segmentation; their general application to biomedical research has recently been reviewed in the \textit{Journal of Physics in Medicine and Biology}.\textsuperscript{24} Similarly, gray-scale approaches for segmentation of a single image have been critiqued elsewhere.\textsuperscript{15,20}

MR sensor characteristics have been extensively reviewed elsewhere, so this review will \textit{not} address sensor-related factors that include (a) systematic errors in the calculation of MR-related parameters that may be used as features (that is, numerical characteristics) for image segmentation;\textsuperscript{14} (b) RF coil uniformity corrections necessary if image-intensity-based segmentation methods are used as opposed to MR-parameter-based methods;\textsuperscript{25} (c) influence of eddy current effects, RF excitation profile, and related partial volume effects attributed to slice thickness; and, finally, (d) registration correction requirements.\textsuperscript{26} MRI segmentation methods for 3-D radiation treatment planning\textsuperscript{1,2,27–31} and multimodality and functional imaging\textsuperscript{2,27–31} are likewise surveyed elsewhere. We will not review methods based on gray-scale segmentation or edge and boundary detection. And we do not discuss linear-algebraic approaches such as eigenvector methods that do not take full advantage of the multispectral character of MR imagery, especially if it contains MR relaxation parameters. These methods often fail to segment pathology as opposed to normal tissue distribution.\textsuperscript{1,11,12}

This review focuses on image-intensity-based segmentation that permits the use of multidimensional image data. Corrections for RF coil characteristics were avoided by using a multi-element quadrature resonator head coil and obtaining axial slices within the center of the coil.\textsuperscript{1,11,12} The optimization of MRI methods for segmentation is treated here as a topic for subsequent review, provided they prove as clinically useful as we expect them to be.\textsuperscript{2,18} The primary aim of this work is to address the theoretical basis for and optimization of multiparameter pattern recognition methods as they apply to MRI data. We shall emphasize the need for algorithmic modifications to secure optimal performance, because these methods are strongly dependent on image modalities.\textsuperscript{1} The tailoring of these algorithms for MR segmentation, most of which were developed for defense applications, exemplifies the need for interdisciplinary research in emerging attempts to develop \textit{dual use} technologies for defense and medical applications.\textsuperscript{32}

Section II of the review discusses fuzzy sets and models of uncertainty and gives a general overview of pattern recognition problems. Section III is concerned with supervised methods (classifier design) for MRI segmentation. Section IV describes MRI segmentation approaches based on unsupervised learning or clustering. Section V contains a summary and our estimation of future directions in MRI segmentation.

II. PATTERN RECOGNITION: PROBLEMS AND MODELS

A. Fuzzy models: Nonstatistical uncertainty

Fuzzy sets are a generalization of conventional set theory that were introduced by Zadeh as a way to represent vagueness in everyday life.\textsuperscript{32} Conventional (crisp) sets contain objects that satisfy \textit{precise properties} required for membership. The set of real numbers $H$ from 6 to 8 is crisp; we write $H = \{ r \in \mathbb{R} | 6 < r < 8 \}$. Equivalently, $H$ is described by its \textit{membership function} (MF), $m_H: \mathbb{R} \rightarrow \{ 0, 1 \}$ defined as

$$m_H(r) = \begin{cases} 1, & 6 < r < 8, \\ 0, & \text{otherwise}. \end{cases}$$

The crisp set $H$ and the graph of $m_H$ are shown in the left half of Fig. 1. Every real number ($r$) either is in $H$, or is not. Since $m_H$ maps all real numbers $r \in \mathbb{R}$ onto the two points $\{ 0, 1 \}$, crisp sets correspond to two-valued logic. In logic, values of $m_H$ are called "truth values" with reference to the question "is $r$ in $H$?" The answer is yes, if and only if $m_H(r) = 1$; otherwise, no. Consider next the set $F$ of real numbers that are \textit{close} to 7. Since the property "close to 7"
is fuzzy, there is not a unique membership function for $F$. Rather, the modeler must decide, based on the potential application and properties desired for $F$, what $m_F$ should be. Properties that seem plausible for $m_F$ might include: 

1. Normality: if $r = 0$, then $m_F(r)$ is 1; 
2. Monotonicity: if $r_1 > r_2$, then $m_F(r_1) > m_F(r_2)$; 
3. Symmetry: $m_F(r)$ is symmetric about $r = 0$.

Given these intuitive constraints, either of the functions shown in the right half of Fig. 1 might be a useful representative of $F$. Function $m_{F_1}$ is discrete (the staircase graph), while $m_{F_2}$ is continuous but not everywhere smooth (the triangle graph). One can easily construct a MF for $F$ so that every number has some positive membership in $F$, but we would not expect numbers “far from 7” (e.g., 20,000/987) to have much.

Fig. 2. Fuzzy sets are membership functions.

Readers often wonder what the fuzzy “set” is, physically. In conventional set theory, any set of real objects is completely equivalent to, and isomorphically described by, a crisp membership function such as $m_F$. However, there is no set-theoretic equivalent of “real objects” corresponding to $m_F$, the function-theoretic representation of $F$. Fuzzy sets are always (and only) functions, from some “universe of objects,” say $X$, into $[0,1]$. This is depicted in Fig. 2, which illustrates that the fuzzy set is the function $m$ that carries $X$ into $[0,1]$. So, the membership function is the basic idea in fuzzy set theory; its values measure degrees to which objects satisfy imprecisely defined properties. Readers will have some questions about fuzzy sets at this point.

Q1. Membership values—What are they? The statement $m_{F_2}(6.8) = 0.8$ (Fig. 1) tells us that 6.8 is “pretty close” to 7. In short, 0.8 is a measure of the similarity of this number to others that possess the imprecise property that $m_{F_2}$ represents. We do not interpret 0.8 as “the probability that 6.8 is pretty close to 7 is 0.8.”

Q2. Membership functions—Where do they come from? Well, where do probability density functions (PDFs) come from? We get them from data and people. Data, for example, are used in parametric estimation of the means and covariances of the component densities in a mixture of normal distributions. And we use data to find fuzzy partition memberships with the fuzzy c-means clustering algorithm. But what of the normal density itself? Is it somehow different than, say, just “adopting” function $m_{F_2}$ because it seems to provide a reasonable and useful model of the process being described? Well, no. The normal distribution was introduced by DeMoivre in 1733 in connection with gambling data because it provided a reasonable explanation for and model of a real process. Today we still use it for the same reason—because it often fits the physical world. So, we do get PDFs and MFs in the same ways, and for the same purposes.

Q3. Is not fuzziness just a clever disguise for probability? No. To see why not, let $L = \{\text{all liquids}\}$ and let fuzzy subset $\mathcal{L} = \{\text{all (potable) liquids}\}$. Suppose you had been in the desert for a week without drink and you came upon two bottles $A$ and $B$, marked with the following information: prob($A \in \mathcal{L}$) = 0.91, memb($B \in \mathcal{L}$) = 0.91. Confronted with these bottles, and given that you must drink from the one that you chose—which would you choose to drink from first? Most readers see that while $B$ could contain, say, swamp water, it would not (discounting the possibility of a Machiavellian fuzzy modeler!) contain liquids such as hydrochloric acid. That is, membership of 0.91 means that the contents of $B$ are “fairly similar”—with respect to the model defined—to perfectly potable liquids (pure water, perhaps). On the other hand, the probability that $A$ is potable = 0.91 simply means that over a long run of experiments, the contents of $A$ are expected to be potable in about 91% of the trials; and in the other 9%? In these cases the contents will be deadly—about 1 chance in 10. Thus, subjects will opt for the swamp water. There is another facet to this example that concerns the idea of observation. Suppose we examine the contents of $A$ and $B$, and discover that $A$ is acid, $B$ is beer. After observation, the membership value for $B$ remains unchanged, while the probability value drops from 0.91 to 0.0. Thus, these two models possess different kinds of information; fuzzy memberships, which represent similarities of objects to imprecisely defined properties; and probabilities, which convey information about relative frequencies. Moreover, there are interpretations of probability that are not based on relative frequencies at all. There are many amusing articles about the relationship between fuzzy sets and probability in the literature. A number of books that deal with elementary notions of fuzzy sets and their generalizations are available.

B. Pattern recognition: Problems and approaches

Figure 3 shows the relationship between the four components of a PR system (PRS). The four nodes in Fig. 3 are not independent. In the ideal world, perfect features make classifier design trivial; and, conversely, a universal classifier would give error-free performance with any set of features. In practice, however, the successful PRS is developed by iteratively revisiting each of the four nodes in Fig. 3 until the system satisfies (or is at least optimized for) a given set of performance requirements. Many texts describe various approaches to the problems depicted in Fig. 3.

Two data structures are used in numerical PRs: object data (feature vectors); and (pairwise) relational data (similarities, proximities). Object data are represented as $X = \{x_1, x_2, \ldots, x_n\}$, $n$ feature vectors in feature space $\mathbb{R}^d$. The $j$th object (some physical entity such as a person, airplane, seismic record, image pixel, etc.) has $x_j$ as its numerical representation; $x_{jk}$ is the $k$th characteristic (or fea-
Fig. 3. Elements of a typical numerical pattern recognition system.

ture) associated with object j. It may happen that we have access to a set of \((n^2)\) numerical relationships, say \(\{r_{jk}\}\), between pairs of objects. That is, \(r_{jk}\) represents the extent to which objects \(j\) and \(k\) are related in the sense of some binary relation. However, this type of data is rare in image processing and will not enter our discussion.

Feature analysis includes scaling, normalization, smoothing, and other "clean-up" techniques necessary to get the best possible results in MRI segmentation. Techniques that fall into this category include the use of variable conductance diffusion to narrow feature vector distributions and multiscale descriptions of image structures for segmentation of biomedical images.\(^{45,46}\) Fuzzy and neural models have been used for problems such as blurring, noise, low contrast, and obscuration.\(^{45,46}\) Segmentation appears in both the classifier and clustering nodes in Fig. 3; Secs. III and IV cover techniques belonging to each of these classes, respectively.

Perhaps the most basic idea in pattern recognition is the class label. There are three types of labels extant in the literature—crisp (or "hard" or nonfuzzy); fuzzy, and probabilistic. Let \(c\) denote the number of classes \(1 < c < n\), and define three sets of label vectors in \(\mathbb{R}^c\) as follows:

\[
N_{fcu} = \{y \in \mathbb{R}^c | y_i \in [0, 1] \ \forall i\}
\]

(1a)

\[
N_{fc} = \left\{ y \in N_{fcu} \mid \sum_{i=1}^{c} y_i = 1 \right\} = \text{(constrained) fuzzy/prob.}
\]

(1b)

\[
N_c = \{y \in N_{fc} | y_i \in \{0, 1\} \ \forall i\} = \text{crisp (or hard)}
\]

(1c)

where \(N_c\) is the canonical (unit vector) basis of Euclidean \(c\)-space; \(N_{fcu}\), a piece of a hyperplane, is its convex hull; and \(N_{fcu}\) is the unit hypercube in \(\mathbb{R}^c\). Figure 4 depicts these sets for \(c=3\). For example, the vector \(y = (0.1, 0.6, 0.3)^T\) is a typical constrained label vector; its entries lie between 0 and 1, and sum to 1. The interpretation of \(y\) depends on its origin. If \(y\) is generated by, say, the fuzzy \(c\)-means clustering method, we call \(y\) a fuzzy label; on the other hand, if \(y\) came from a method such as maximum likelihood estimation in mixture decomposition, \(y\) would be a probabilistic label. The cube \(N_{fcu} = [0, 1]^3\) is called \(\text{unconstrained}\) label vector space; vectors such as \(z = (0.7, 0.2, 0.7)^T\) have each entry between 0 and 1, but are otherwise unrestricted.

Clustering in unlabeled data set \(X\) is the assignment of (hard or fuzzy or probabilistic) label vectors to the objects generating \(X\). If the labels are hard, we hope they identify \(c\) "natural subgroups" in \(X\). Clustering is also called \(\text{unsupervised learning}\), the word \(\text{learning}\) referring here to learning the correct labels (and possibly vector prototypes or quantifiers) for "good" subgroups in the data. \(c\)-partitions of \(X\) are characterized as sets of \((cn)\) values \(\{u_{ik}\}\) satisfying some or all of the following conditions:

\[
0 < u_{ik} < 1, \ \forall i, k,
\]

(2a)

\[
0 < \sum_{k} u_{ik} < n, \ \forall i,
\]

(2b)

\[
\sum_{i} u_{ik} = 1, \ \forall k.
\]

(2c)

Using Eqs. (2) with the values \(\{u_{ik}\}\) arrayed as a \((c \times n)\) matrix \(U = [u_{ik}]\), we define

\[
M_{fcu} = \{U \in \mathbb{R}^{cn} | u_{ik} \text{ satisfies (2a) and (2b) } \forall i, k\},
\]

(3a)

\[
M_{fc} = \{U \in M_{fcu} | u_{ik} \text{ satisfies (2c) } \forall i, k\},
\]

(3b)

\[
M_{cn} = \{U \in M_{fcu} | u_{ik} = 0 \text{ or } 1 \ \forall i, k\}.
\]

(3c)

Equations (3a)–(3c) define, respectively, the sets of unconstrained fuzzy (or probabilistic), constrained fuzzy (or probabilistic), and crisp \(c\)-partitions of \(X\). We represent
clustering algorithms as mappings of $\mathcal{X} \rightarrow \mathcal{M}_{\text{fuzzy}}$. Each column of $U$ in $M_{\text{fuzzy}} (M_{\text{fuzzy}})$ is a label vector from $N_{\text{fuzzy}} (N_{\text{fc}}, N_{c})$. The reason these matrices are called partitions follows from the interpretation of $u_{ik}$ as the membership of $x_i$ in the $i$th partitioning subset (cluster) of $X$. $M_{\text{fuzzy}}$ and $M_{\text{fuzzy}}$ can be more realistic physical models than $M_{c}$, for it is common experience that the boundaries between many classes of real objects (e.g., tissue types in magnetic resonance images) are really fuzzy. We give an example. Let $X = \{x_1, x_2, x_3\} = \{\text{peach, plum, nectarine}\}$, and let $c = 2$. Typical 2-partitions of these three objects are shown in Table I. The nectarine $x_3$ is shown as the last column of each partition and, in the hard case, it must be (erroneously) given full membership in one of the two crisp subsets partitioning this data; in $U_1$, $x_3$ is labeled “plum.” Fuzzy partitions enable algorithms to (sometimes!) avoid such mistakes. The final column of the first fuzzy partition in Table I allocates most (0.6) of the membership of $x_3$ to the plum class; but also assigns a lesser membership of 0.4 to $x_3$ as a peach. The last partition in Table I illustrates an unconstrained set of membership assignments for the objects in each class. Columns like the one for the nectarine in the two fuzzy partitions serve a useful purpose—weak memberships in several classes are a signal to “take a second look.” Hard partitions of data cannot suggest this. In the present case, the nectarine is a hybrid of peaches and plums, and the memberships shown for it in the last column of either fuzzy partition seem more plausible physically than crisp assignment of $x_3$ to an incorrect class. Statistical clustering algorithms—e.g., unsupervised learning with maximum likelihood—also produce solutions in $M_{\text{fuzzy}}$. Fuzzy clustering began with Ruspini; 

<table>
<thead>
<tr>
<th>Object</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peaches</td>
<td>[1 0 0]</td>
<td>[0.9 0.2 0.4]</td>
<td>[0.9 0.4 0.5]</td>
<td>[0.9 0.5 0.5]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plums</td>
<td>[0 1 1]</td>
<td>[0.1 0.8 0.6]</td>
<td>[0.6 0.8 0.7]</td>
<td>[0.6 0.8 0.7]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The convex combination

$$f(x) = \sum_{i=1}^{c} \pi_{i} g(x | i)$$

is itself a PDF whose distribution is called a mixture of the components $\{\pi_{i} g(x | i)\}$. Let the posterior probability that, given $x$, $x$ came from class $i$, be denoted by $\pi_{i}(x | i)$. Bayes’ rule relates the elements of (4) to the probabilities $\{\pi_{i}(x | i)\}$ as follows:

$$\pi_{i}(x | i) = \frac{\pi_{i} g(x | i)}{f(x)}.$$

For a finite sample $X = \{x_1, x_2, ..., x_n\}$ drawn from mixture $f$, the $c \times n$ matrix $\Pi = [\pi_{i}(x_1 | x)] = [\pi_{ik}]$ of posterior probabilities satisfies constraints (2); thus, $\Pi \in M_{\text{fuzzy}}$, the space of constrained fuzzy c-partitions of $X$. In the present context $\pi_{ik}$ is indeed a probability, and plays much the same role in statistical PR that the fuzzy membership value $u_{ik}$ plays in fuzzy PR. If the elements of the right-hand side (RHS) of (5) are known, we may compute, at any $x \in \mathbb{R}^p$, the posterior probability vector $\pi(z | x) = (\pi(1 | z), \pi(2 | z), ..., \pi(c | z))$. This vector lies in the set $N_{fc}$ shown in Fig. 4; it is a probabilistic label vector. Using the LHS of (5), we define the statistical Bayes’ classifier $D_{\text{pb}}: \mathbb{R}^p \rightarrow N_{fc}$ as follows:

$$D_{\text{pb}}(z) = \pi(\ast | z).$$

$D_{\text{pb}}$ is a probabilistic classifier. For example, the vector $y = D_{\text{pb}}(z) = (0.1, 0.6, 0.3)^T$ in Fig. 4 might be such a vector derived from (6) for some $z \in \mathbb{R}^p$. Fuzzy labels are usually “defuzzified” (reduced to a crisp label, one of the vertices $e_i$ displayed in Fig. 4) in order to render a crisp or hard decision about $z$. For the same reason (and, in fact, often in the same way), probabilistic labels need to be converted to hard labels. The Bayes’ decision rule (BDR) is used to convert the probabilistic classifier at (6) into a crisp one using the simple and intuitively plausible rationale that, given $z$, we should assign $z$ to the class of maximum posterior probability. In the form of a crisp classifier function, we write $D_{\text{B}}: \mathbb{R}^p \rightarrow N_c$ as follows:

$$D_{\text{B}}(z) = e_i \iff \pi(i | z) = \max_j (\pi(j | z)).$$

$D_{\text{B}}$ is a crisp classifier. Put more directly, we can state the BDR as follows:

BDR: Decide $z \in \text{class } i \iff \pi(i | z) > \pi(j | z)$ for $j \neq i.$

For example, if the vector $y = D_{\text{B}}(z) = (0.1, 0.6, 0.3)^T$ is a posterior probability vector, we find, using (7) or (8), that...
the crisp label assigned to z is class 2, since the maximum (0.6) occurs in the second component of y. That is,
\[ \mathbf{D}_{\text{Bayes}}(z) = (0,1,0.6) \] or \[ \mathbf{z} = (0,1,0) \]. This rule minimizes the expected total misclassification error (or error rate) of any classifier \( \mathbf{D}: \mathbb{R}^p \rightarrow N_{\text{class}} \) under the assumptions implicit in (3)–(5) about mixtures, provided the cost of misclassification is equal for all wrong decisions, and there is no cost for correct decisions (the 0–1 loss matrix assumption). In this case \( \mathbf{D}_{\text{Bayes}} \) is often called the “optimal Bayes’ classifier.” Figure 5 depicts the assumptions and geometry of the mixture model. Shown there is a mixture of \( \mathcal{C} = 2 \) univariate \( \{p_i\} \) normal distributions with prior probabilities \( \{p_i\} \) and class conditional densities \( g(x | i) = \frac{1}{(2\pi)^{p_i/2} \sigma_i^2} \exp \left[ -\frac{(x - \mu_i)^2}{2\sigma_i^2} \right] \), where \( \sigma_i \) is the standard deviation and \( \mu_i \) is the mean, \( i = 1, 2 \). The optimal Bayes’ classifier is defined by the point \( x^* \); BDR is decide \( z \in \text{class} \) 1 iff \( z < x^* \), and otherwise, decide \( z \in \text{class} \) 2. The regions \( D_{b1} \) and \( D_{b2} \) are called the Bayes’ decision regions of this classifier, and the (minimum) Bayes’ error rate \( E_{\text{Bayes}} \) is the sum of the two areas shown in Fig. 5; \( E_{\text{Bayes}} = A_1 + A_2 \). Moving the decision boundary from \( x^* \) to, say, \( x \), as in Fig. 5, results in a positive increase in the probability of error by the amount \( \delta A \).

Several widely known and often used segmentation methods discussed below are based on the mixture model (or are related to it asymptotically). There are many good comprehensive introductions to this area of classifier design. 40,42,54

D. Computational neural networks (CNNs)

CNNs attempt to replicate human performance by imitating what we presume to be true about sensory data processing. 55–58 CNNs have shown great promise at various image processing tasks. 59,60 CNNs can be broadly categorized as classifier nets (supervised learning) and clustering nets (unsupervised learning). Our review will illustrate and discuss both types.

A CNN consists of many processing elements or “nodes,” each of which may have a local memory and can carry out localized processing. Figure 6 reflects current thinking about the processing that should be performed at local nodes. 58 Two mathematical functions are active at each node. An integrator function \( f \) aggregates the (synaptic) weights \( \{w_i\} \) with the inputs \( \{x_i\} \) to the node. Usually \( f \) is the Euclidean dot product plus a constant,

\[
y = f(x) = \sum_{i=1}^{p} w_i x_i + \alpha.
\]

This \( f \) defines the hyperplane \( H \) which is orthogonal to \( w = (w_1, w_2, \ldots, w_p) \) and displaced from the origin by \( \alpha \). \( H \) divides \( \mathbb{R}^p \) into plus and minus half-spaces, i.e., regions where \( f \) is positive or negative; \( f \) is zero on \( H \). The term \( 1*\alpha = \alpha \) is called the offset or bias at this node. The terms \( (\alpha \text{ and } 1) \) are shown with a dotted (shaded) line to indicate that \( \alpha \) does not mimic a “synaptic weight.” Nonetheless, it is an unknown parameter for each node that must be set or learned during training. When using

\[
y = f(x) = \sum_{i=1}^{p} w_i x_i + \alpha.
\]

we can regard the augmented vectors \( x = (x_1, x_2, \ldots, x_p, 1) \) as input and weight vectors in \( \mathbb{R}^{p+1} \). In subsequent discussion we assume the structure of Fig. 6 for each node, and regard it as having \( p + 1 \) unknowns.

The integrated value \( y = f(x) \) is fed to a transfer function \( F \). The most common choice for \( F \) is the logistic or sigmoidal function, \( z = F(y) = 1/(1 + e^{-y}) \). The output \( z \) of the transfer function \( F \) is then broadcast to all nodes connected to the node that computes the value. This review concentrates on the use of two classifier networks, the feed forward/back propagation and feed forward/cascade correlation designs, which are described in Sec. III. Clustering networks can also be used for image segmentation, 60–63 but have not been extensively applied to MR images at this writing.

E. MR image data for classifier design and cluster analysis

In MR image segmentation classifiers are “trained” to “learn” the boundaries of different tissue types in the image. Input data will typically be some subset of \( T1, T2, \) and proton density pixel intensities at each spatial location in the image; or more generally, a vector of features con-
structured from them. Specifically, let $T_{1,ij}$, $T_{2,ij}$, and $\rho_{ij}$ denote the spin lattice relaxation, transverse relaxation, and proton density of pixel $(i,j)$ in an MR slice of overall dimensions $(m \times n)$. We aggregate these into the pixel vector $x_{ij} = (T_{1,ij}, T_{2,ij}, \rho_{ij})$ in $\mathbb{R}^3$, and data set $X = \{x_{ij}, \ldots, x_{ij}, \ldots, x_{mn}\}$. For a $256 \times 256$ image this results in a set of 65,536 pixel vectors in $\mathbb{R}^3$. Figure 7 illustrates this procedure for $p$-dimensional data.

Segmentation of an MR slice (or any $p$-dimensional multispectral image) begins with aggregation. One may choose to aggregate only selected bands—e.g., we might choose to concentrate on the pair $(T1, T2)$, so $X \subset \mathbb{R}^2$. The first two rows of Fig. 8 show typical MRI data (256 $\times$ 256) in parameters $T1$, $T2$, and $\rho$ for pre- and postcontrast enhanced images. The segmentations of these images shown in rows 3 and 4 will be discussed later.

Supervised methods need crisp labels for some of the pixel vectors in $X$, so $X$ is partitioned into a training set $(X_T)$ and a test set $(X_F)$, $X_T = X - X_F$, $|X_T| = (mn - n_L)$. Each training subset $X_L$ has $n_L$ points for class $(i)$, and $\Sigma n_L = n_L$. There are more sophisticated methods for subdividing $X$ into training and test sets. $X_L$ has a hard label matrix $U_L$ whose $k$th column exhibits the membership of $x_k$ in one of the $c$ tissue classes, for $k = 1, \ldots, n_L$. Most of the methods we review use crisp labels (every column of $U_L$ is a vertex $e_i$ as in Fig. 4). However, many of these methods can be adapted to use fuzzy or probabilistic labels for training, and methods based on these techniques are being investigated by many researchers at this time.

Where do the labels of training set $(X_L)$ for MR image pixel vectors come from? An operator familiar with human brain anatomy must select small spatial regions in visual displays of MR imagery, such as cerebrospinal fluid (CSF), white matter, gray matter, etc., and assign every pixel vector in each region a (crisp) physical label (color). Since each patient has different anatomical structure, it is necessary at present to obtain labeled data for algorithmic training for each subject whose MR scans are to be analyzed with any of the methods reported in this section. This is a time-consuming procedure. A hope of the medical computing community is that eventually enough experience across different subjects will lead to a "library" of prelabeled prototypical images that can be used to initialize unsupervised algorithms and thereby eliminate this costly step.

In the unsupervised approach $X$ is processed with all the pixels in the image unlabeled. In this case clustering algorithms try to group like (tissue) pixels together. Hopefully, mathematically similar pixels will belong to the same physical tissue types. This approach also requires human intervention, but in this case it occurs at the end of the computational cycle instead of at the beginning. Specifically, clustering unlabeled data will result in image subregions that possess mathematical (numerical) labels, so a human must inspect the synthetically colored image to assign physical labels to each region. Both approaches are unsatisfactory in that they require human intervention to produce MRI segmentations.
The point of intervention has very different effects on final segmentations of the MR data. In supervised learning, the operator must be skilled at visually identifying physical tissue classes in real MR data, as these will be used to automatically assign physical labels to the test pixels in the image after training. In the unsupervised approach, the operator views segmentations and must be able to assign realistic tissue class labels to algorithmically produced subregions of the image. Efforts to reduce the dependency of MR segmentation on humans will probably never produce a fully automated process.

Figure 9 summarizes the difference between the supervised and unsupervised approaches. The right side of Fig. 9 emphasizes that the operator intervenes at the beginning of the procedure in supervised approaches. Here, the choice of "good" training data is crucial. Different training sets lead to great disparities in training time, as well as potential differences in the final classifier function $D$ and, subsequently, the segmentation itself. We emphasize that in this approach the operator decides, based on the MR data being inspected, how many tissue classes exist in the image (that is, the operator explicitly chooses $c$, the "correct" number of tissue classes that the classifier will be trained to recognize). Moreover, the operator will specify a color for each tissue class that is automatically preserved by $D$.

In unsupervised learning (the left side of Fig. 9), $c$ must be specified by the operator to the clustering algorithm, perhaps incorrectly, since this is (in principle) done without benefit of inspection of the MR data. After clustering, each class possesses an algorithmic label (color), and the operator must complete the final step in the procedure. Since the operator is asked to assign a physical label to each region (cluster), it is crucial that the operator be able to imagine tissue structure as it must be in the data in order to label the segmented regions correctly in the unsupervised case.

### III. SUPERVISED (CLASSIFIER) METHODS IN MR IMAGE PROCESSING

#### A. Parametric designs: Bayes’ classifiers with maximum likelihood estimators

One of the simplest schemes for segmentation is the parametric Bayesian design. This begins by assuming functional forms for the density functions $g(x|i)$ in the mixture model, and leads to the problem of parametric estimation of the Bayes' classifier. We indicate this by attaching an unknown parameter vector $\theta_i$ to each PDF in the mixture, writing

$$f(x;\theta) = \sum_{i=1}^{c} \pi_i g(x|i;\theta_i)$$

(9)

to indicate that the mixture density now depends on a parameter vector $\theta = (\theta_1, \theta_2, ..., \theta_c)$. Many schemes are available for estimation of the $\{\theta_i\}$ using either labeled or unlabeled data. One of the most popular ways to do this is maximum likelihood estimation (MLE). Given $X$, we form the likelihood function of the samples and try to maximize it. If the form of the $g(x|i)$ is "nice," this can be done by constrained optimization methods. In particular, if each of the component densities is $p$-variate normal, $g(x|i) = n(\mu_i, \Sigma_i)$, where $\mu_i$ is the mean vector and $\Sigma_i$ is the covariance matrix for class $i$, the parameter vector for each class is $\theta_i = (\pi_i, \mu_i, \Sigma_i)$. In this case, necessary conditions for the MLE of the parameter vectors in (9) can be derived, and lead to two subcases, depending on whether the data are labeled or not. The unlabeled case is discussed in the first subsection of Sec. IV.

If the data are labeled, the sample posterior probability matrix $P = [p_{ik}] = [p(i|x_k)] = U_L$, so the equations for MLE of each $\theta_i = (\pi_i, \mu_i, \Sigma_i)$ are uncoupled across classes and parameters, and hence are explicit (noniterative). In this case, we simply take the training data $X_{UL}$ for each class (each tissue class in the image), substitute it into the necessary equations, compute MLEs of the parameters $\theta_i = (\pi_i, \mu_i, \Sigma_i)$ of each density, and use the resulting functions as estimators of the RHS of (5). This yields a parametric estimate of the posterior probability vector (in the parlance of neural network terminology, we have at this point "trained" the classifier $D_{UL}$ by finding its parameters), where $bL$ stands here for the "Bayes, labeled" data case. We shall refer to this scheme for segmentation as the labeled ML method (LMLM). Once this step is complete, segmentation of the remaining portion of the MR image proceeds by simply computing $D_{UL}(x)$ for each pixel vec-
tor z from the image test data \(X_T\), and assigning a tissue class label to it using the BDR. Finally, we display the results as an artificially colored image by assigning a color to each tissue class, and coloring each pixel in the spatial array according to its class label.

MR data does not generally support the assumptions that underlie the ML method. That is, the distribution of tissue classes is not necessarily a statistical mixture, and this often results in MLE yielding poor segmentations. The use of supervised methods, including ML, has proved difficult with one training set used across different patients.\(^54\) In fact, in the cited study maximum likelihood had the least stability across patients among six segmentation methods. This means ML requires the choice of very "good" training sets for each patient to which it is applied. Hence, ML methods will be difficult to use without careful supervision on every slice to be segmented.

### B. Nonparametric designs: The k-nearest neighbor rule

The scheme in Sec. III A is called a parametric method, because we assume some knowledge of the components of the mixture, and then use labeled data and a principle of statistical inference such as ML to estimate the parameters of the functions in (9). Another widely used method for MR segmentation is the k-nearest neighbor (k-nn) rule. This is called a nonparametric method, because the k-nn rule does not require any knowledge or assumptions about statistical properties of the data.\(^40\) The k-nn rule essentially relies on having a large number of (presumably) correctly labeled samples from each tissue class. Figure 10 displays the geometry of this scheme, which can be easily described without recourse to formulae. Basically, all that is needed is to choose \(k\), the number of nearest neighbors to find in the neighborhood of any unlabeled pixel vector \(z\) in \(X_T\); and some measure of distance between pairs of vectors in \(\mathbb{R}^p\), usually Euclidean distance

\[
d(z, x_i) = \|z - x_i\| = \sqrt{(z - x_i)^T(z - x_i)}.
\]

One must also choose a voting scheme, which is often to accept a simple majority of the votes for any class represented by points in the k-nn neighborhood. In Fig. 10 with \(k = 6\) nearest neighbors having \(c = 3\) class labels, the point \(z\) will be labeled (and subsequently colored) as a class 2 pixel, because three of its nearest six neighbors have this crisp label.

To understand the relationship between this approach to MR segmentation and the LMLM discussed in Sec. III A, we formalize the k-nn rule as follows.

**Hard or fuzzy k-nn algorithm—Simple majority voting case:**

1. Store training data \(X_L\) with its hard or fuzzy c-partition \(U_L\).
2. Choose \(k\) = number of neighbors to find.
3. Choose \(d: \mathbb{R}^P \rightarrow \mathbb{R}\) any metric (distance measure) on \(\mathbb{R}^P\).
4. For all vectors \(z\) in \(X_T\): using \(X_L = \{x_i\}\), compute and rank order the distances \(d(z, x_i)\) as \(d_1 < d_2 < \cdots < d_k < d_{k+1} < \cdots < d_n\). Find the columns in \(U_L\) corresponding to the \(k\)-nearest neighbor indices \(\{1, 2, \ldots, k\}\). Calculate the vector \(u(*)|z\) = \{(u(1)|z), u(2)|z\),...,u(c)|z\}\)\(^T\) with the nn labels:

\[
u(i)|z\} = \frac{1}{k} \sum_{j=1}^{k} u_{L,i}^{*|z|}, \text{ for } i = 1, 2, \ldots, c. \quad (10)
\]

**Calculate**

\[
D_{nn,k}(z) = e_i \Leftrightarrow u(i)|z} = \max_j\{u(j)|z}\}. \quad (11)
\]

Continue.

This procedure is not iterative; each unlabeled pixel in the image is processed once with this method. The vector \(u(*)|z\) is again a crisp or fuzzy label vector \(f(z)\) in \(N_F\); Eq. (11) defuzzifies the label vector exactly as in Eq. (7), to yield a crisp tissue assignment to each pixel in the image. In order to compare this method to the LMLM (and, subsequently, feed-forward neural network schemes as well), we let \(E_{nn}\) denote the expected error rate of the nn classifier \(D_{nn,k}\). Note that \(u(*)|z\) is similar to the posterior probability vector \(\pi(*)|z\) = \{(\pi(1)|z), \pi(2)|z\},...,\pi(c)|z\}\)\(^T\) in that it lies in \(N_F\), and thus has exactly the same formal properties as \(\pi(*)|z\). A famous and remarkable (because the k-nn rule has no statistical assumptions whatsoever!) theorem due to Cover and Hart\(^65\) shows that, under suitable restrictions on \(k\) and \(n_L\), for crisply labeled training data, \(E_{nn}\) converges to \(E_0\) as \(n_L\) approaches infinity, and \(u(*)|z\) becomes an arbitrarily good estimate of the Bayes' vector \(\pi(*)|z\). That is, the distance \(\|\pi(*)|z\) - u(*)|z\|\) approaches 0 as \(n_L \rightarrow \infty\).

This means that the k-nn rule is asymptotically equivalent to Bayes' rule. Although this result is psychologically reassuring, an operator usually labels only a very few pixels in an MR image, so it is unfair to expect the k-nn rule to produce the Bayes' error rate on small samples. On the other hand, the mixture assumptions on which the Bayes' error rate are based are not usually sustained in MR tissue regions, so the k-nn rule may, in fact, produce much better results than the LMLM. Indeed, our experience indicates that this is certainly the case.

Figures 8(g) and 8(j) are k-nn segmentations of, respectively, the pre- and postcontrast enhanced MR data shown in the top two rows of Fig. 8. This pair of segmentations was produced using Euclidean distance, \(k = 7\) neighbors with simple majority voting, and a small (about 300 pixels/class) set of labeled pixel vectors from each of
Fig. 11. MR segmentations, pre- and postchemotherapy pairs. (a) and (b): k-nn rule. (c) and (d): FF/CC. (e) and (f): FCM-crisp colors. (g) and (h): FCM-fuzzy colors.

c = 7 operator selected tissue classes. The data were treated as described in Sec. II E, i.e., images 8(a)–8(c) and 8(d)–8(f) were regarded as sets \( X_{abc} \) and \( X_{def} \) of 256 \( \times \) 256 = 65536 pixel vectors in \( R^3 \), respectively, composed as shown in Fig. 7. The k-nn rule produces very clear definition of the tumor, especially in view 8(j), which resulted from processing image \( X_{def} \), the MR image collected after gadolinium was injected for contrast enhancement. Away from the tumor, the segmentations produced by other methods to be discussed below, including unsupervised methods, are sometimes preferred.1

There are usually three parameters associated with the k-nn rule: (i) k itself; (ii) the measure of distance; and (iii) the method of counting votes. To these, we may add a fourth—the type of labels. Observe that the label vectors used for voting in our example in Fig. 10 were crisp. However, Eq. (10) works just the same when the label matrix \( U_L \) is a fuzzy partition of the training data. In this case, the k-nn's to each \( z \) in \( X_T \) will have fuzzy label vectors, and the "vote" for each label will be noninteger. This last method is the fuzzy k-nn rule; its implementation is exactly the same as is illustrated in Fig. 10, with the exception of the labels attached to each neighbor. This extension of the hard k-nn rule has not been tried for MR segmentation as yet, but success with this scheme in other domains argues well for its use in this context.65,67

Figures 11(a) and 11(b) offer a second example of segmentation with the k-nn rule. The segmentations in this figure pertain to precontrast enhanced MR data of a patient in pre- and postchemotherapy situations. The computing parameters for the k-nn rule segmentations in 11(a) and 11(b) were the same as above (\( p = 3, c = 7, k = 7, d = \) Euclidean). Observe that the tumor in the lower right-hand corner appears smaller. The tumor has grown over time, as shown in the upper middle section of the slice.

C. Feed forward neural network methods

The most popular classifier net is the feed-forward (i.e., no data fed backward) neural network (FF/NN).38 Figure 12 illustrates the major components of this architecture. Each feature vector is fed into an input layer, which fans out the values to nodes in one or more hidden layers. Finally, there is an output layer of nodes that produces c numbers, shown as \( u \in R^c \) in Fig. 12. Thus, the FF/NN is a mapping of \( R^p \) to \( R^c \), i.e., \( NN: R^p \rightarrow R^c \). Any FF/NN is a classifier net if \( NN: R^p \rightarrow N_{out} \). \( NN \) is an implicit function—we never have a closed-form expression for the mapping. Each node in the hidden and output layers may be configured much as in Fig. 6. If, additionally, the learning scheme used during training is some form of back propagation (BP) of corrections for squared errors between the observed and target outputs, we call the design a FF/BP classifier net. A FF/NN with a single input and output layer is called a perceptron. Perceptrons cannot differentiate between classes that are not linearly separable.38

Multilayer FF/NNs are the type most commonly used in MR image segmentation. These networks have one or more hidden layers (layers of nodes that are neither input nor output nodes), as shown in Fig. 12 (some of the connections have been omitted for visual clarity).

Supervised training methods exist to find the weights on the connections between nodes.55,58 Basically, the weights are updated to minimize the error at each node using the method of steepest descent. The squared error between the observed and target labels is written as a function of the node weights in the deepest hidden layer (the one closest to the output layer); steepest descent is applied to the objective function, and error corrections are backpropagated to them. This procedure continues backward until all node weights have been adjusted. Then the next data input is presented, and the training cycle continues in an iterative fashion until the error is reduced to an acceptable level. It is possible to become stuck at a local minimum and not be able to reduce the error below an acceptable level; however, this does not usually happen in practice. The weight update equations are omitted here, but are available in any book on neural networks.55

In developing a feed-forward neural network, several choices must be made. They include the number of hidden layers, how many nodes to use on each hidden layer, the learning rule (and any parameters it requires, such as the learning rate), and whether weight training should be done after each example is seen (incremental training) or after
the error for the entire training set has been calculated (cumulative or batch training). MRI segmentation problems are nonlinearly separable and require multilayer networks.

It has been found that FF/BP neural networks are very sensitive to the training set in MR image segmentation of the brain.\textsuperscript{64} FF/BP nets provide adequate brain segmentations provided that the training data are quite good. They can learn effectively on as few as 250 pixels per class in a \(256 \times 256\) MR image using a multilayer FF/BP network with between three and ten hidden units, which means that training and testing are relatively fast. Efforts to find a "universal" training set that would be useful on many different MR images have been made.\textsuperscript{1} However, instability across different subjects is a continuing problem. That is, results obtained by classifying tissue regions for an entire image derived from one subject by a FF/BP net that was trained using labeled subregions from a different subject are not very good for at least some other subjects. This indicates that FF/BP methods must, at least for now, continue to rely on operator intervention to select good training data for each subject and each slice of data.

While the most common learning rule for multilayer networks is backpropagation, improved learning rules such as "quickprop" are being developed.\textsuperscript{65} Most improvements are in terms of speed, as opposed to performance (lower error rates). Supervised learning methods that self-configure the network architecture have also been developed, such as the feed-forward cascade correlation (FF/CC) network.\textsuperscript{68,69} Self-configuring methods can save time by eliminating the determination of a network architecture by experimental methods (trial and error). The FF/CC whose architecture is illustrated in the Fig. 13 net has been tested on MR brain images.\textsuperscript{1}

All inputs are connected to all outputs. Instead of pre-specifying a number of hidden layers and nodes/layer as in the FF/BP net of Fig. 12, in this scheme one hidden node is introduced between the input and output layers, with augmented weight vector \(w_1 = (w_{h,1,1}, w_{h,2,1}, \ldots, w_{h,p,1}, \alpha_t)^T \in \mathbb{R}^{p+1}\) as shown in Fig. 13. This node is also connected to all inputs and outputs, and any such hidden node will also have all previously introduced hidden nodes connected to it, so the dimension of the weight vector increases by unity for each additional hidden node as illustrated in the figure. This provides the "cascading" architecture. Weights learned after each presentation of the training set are fixed on their links. Hence, the introduction of a new hidden unit, say the \(q\)th hidden node, requires learning only the values for the \((p + q)\) parameters in the augmented weight vector \(w_q = (w_{1,1}, w_{2,1}, \ldots, w_{p,q}, \alpha_t)^T \in \mathbb{R}^{p+1}\). Initially, the outputs of the new node are not yet connected to the active network. A number of passes are made over the training patterns, adjusting the weights of the input connections to the new nodes after each pass. This adjustment is done in the direction that maximizes \(S\), the sum over all output nodes of the magnitude of the correlation between the candidate node's output value \(z\) and the residual output error \(E_o\), observed at each output node \(o\). Strictly speaking, \(S\) is a covariance; if \(c\) is the number of output nodes and \(n_L\) the number of training patterns, \(S\) is defined as

\[
S(w) = \sum_{o=1}^c \left( \sum_{k=1}^{n_L} \left( E_k - \bar{E} \right) (E_{k,o} - \bar{E}_o) \right),
\]

(12)

where \(E_{k,o} = (z_{k,o} - u_{k,o})\), \(z_{k,o}\) is the output of the candidate node on training vector \(x_k\), \(u_{k,o}\) is the desired output value at output \(o\), and \(\bar{E}\) and \(\bar{E}_o\) are the values of \(z\) and \(E_o\) averaged over the \(n_L\) patterns. Then \(\frac{\partial S(w)}{\partial w_1}\), the partial derivative of \(S\) with respect to each of the candidate node's incoming weights, \(w_1\), can be computed as

\[
\frac{\partial}{\partial w_l} (S(w)) = \sum_{o=1}^c \left( \sum_{k=1}^{n_L} \left( w_o (E_{k,o} - \bar{E}_o)f_k I_{i,k} \right) \right),
\]

(13)

where \(s_o\) is the sign of the correlation between the candidate's value and output \(o\), \(f_k\) is the derivative for pattern \(k\) of the candidate node's transfer function with respect to the sum of its inputs, and \(I_{i,k}\) is the input the candidate node receives from node \(i\) for pattern \(k\). Finally, quickprop is used for a fast-converging ascent to maximize \(S\). When the magnitude of the correlation stops improving, the new candidate node is installed in the active network and its output is connected to the output nodes. Finally, the input weights for this node are frozen, and its output weights are trained to minimize the network's output error. This is done in the same way the direct connections were trained, using quickprop. If the network's performance is not satisfactory and no significant error reduction is achieved, a new hidden node is introduced and the cycle is restarted. This cycle is repeated until the error \(E_{cc}(w)\) becomes acceptably small (or until we give up). Sometimes, FF/CC is implemented by considering at each hidden node addition not one but a pool of candidate nodes. These nodes are initialized with different random weights, and then the candidate with the best correlation is installed. There are several implementation improvements that further optimize our simplified description of FF/CC.\textsuperscript{69}

FF/CC has provided better results than FF/BP in segmenting brain images.\textsuperscript{1} Figures 11(c) and 11(d) are segmentations of a pre- and postchemotherapy patient obtained using FF/CC. These outputs correspond to the \(k\)-nn rule results immediately adjacent [Figs. 11(a) and 11(b)]
in that identical training data subsets for \( c = 7 \) tissue classes were used for both algorithms. In the \( k\)-nn approach training simply means that the operator assigns crisp labels to the training data, whereas in FF/CC the network structure (weight vectors and hidden nodes) of Fig. 13 evolves in an effort to replicate the operator-selected label vectors. FF/CC is not as sensitive to the choice of training regions as FF/BP. However, FF/CC segmentations sometimes exhibit noisy outputs in the sense that major tissues are not always consistently put in the same classes. This problem is again a function of the training set and only occurs in some cases. The ability of FF/CC to find its own architecture fitted to the training data has enabled convergence for some abnormal cases for which backpropagation could only find a local minimum with quite a bit of residual error. In Figs. 8 and 11, the \( k\)-nn rule provides a smoother, more realistic, and less noisy segmentation than FF/CC. The results of \( k\)-nn are slightly better for the prechemotherapy case and essentially equivalent in the postchemotherapy data around both the tumor and edema. We remark that the training set used was optimized for \( k\)-nn in subtle ways by the operator, who originally used the \( k\)-nn rule to validate the training set.

IV. UNSUPERVISED (CLUSTERING) METHODS IN MR IMAGE PROCESSING

When image data are unlabeled, as described in Sec. II E and illustrated in Fig. 9, we can segment the image using clustering algorithms. In this case, no human finds and labels subsets of training data. Instead, the entire set of \( mn \) pixel vectors is submitted to a cluster analysis scheme, and the clusters are found algorithmically. It is necessary in the methods used to date to specify \( c \), the number of tissue classes to search for. Moreover, once found, algorithmic clusters (regions that have the same crisp labels) must somehow be assigned physical (tissue) labels. At present, this is a concern that does not have a satisfactory computational solution. Indeed, humans must interfere at the end of the procedure (instead of at the beginning, as is the case in supervised methods) to assess segmentations that clustering algorithms produce. Current efforts to automate the selection of \( (c) \) and evaluation of cluster outputs lie in an area of research called dynamic cluster validity; this is an important step on the way to fully automated segmentation of MR imagery.\(^{70-72}\)

A. Parametric designs: Bayes classifiers with unlabeled maximum likelihood estimators

The main difference between this method and the scheme described in Sec. III A is that here data set \( X \) is not labeled. We refer to this method as the unsupervised MLM (UMLM). If the data are unlabeled, the sample posterior probability matrix \( P = [p_{ik}] = [p(i|x_i)] \) is unknown, and this matrix is very large—\( (c \times mn) \) with \( mn = 65536 \) and \( c \) in the range of 2 to 15. The equations for MLE of each \( \theta_i = (\pi_i, \mu_i, \Sigma_i) \) are now coupled across classes and parameters. In this case, applying Lagrange multipliers to find necessary conditions for maxima of the likelihood function of the samples from (9) results in, even for the case of \( p \)-variate normal distributions, a very large coupled set of nonlinear equations in the unknowns \((P, \theta)\). In this case we resort to a numerical scheme such as the expectation-maximization (EM) algorithm to find approximate solutions to the necessary conditions using iterative optimization.\(^{54}\) This method suffers from all the usual problems of iterative schemes (local traps, initialization, numerical convergence, etc.), but it does enable us to compute MLEs of the parameters \( \theta_i = (\pi_i, \mu_i, \Sigma_i) \) of each density, and again use the resulting functions as estimators of the RHS of (5). In clustering, recall, each point in the data acquires an algorithmically assigned label vector. Parametric estimates from UMLM will be different than the ones obtained with LMLM (Sec. III A), but also yield a (direct) estimate of the posterior probability vector \( \pi(\ast|x) \) at every pixel location in the image, these probabilistic label vectors being the columns of the terminal estimate of \( P \). This completes segmentation of the MR image.

Next we must decide (i) how to color each pixel, and (ii) what each class is, physically. After assigning a crisp color to each tissue class [using the maximum rule at (7), for example], we color each pixel in the spatial array according to its class label—but this still does not assign tissue labels to the segmented regions. Another possibility is to use the values in each column of \( U \) or \( P \) to assign "fuzzy" colors to each pixel to create fuzzy displays\(^{5}\). This method begins by assigning each pixel to its maximum membership (or probability, if a probability-based method is used) class. Each class is assigned a specific color and pixels in that class are assigned shades of the chosen color based on their strength of membership (or probability) in the class. Lighter shades are assigned to the pixels with strong membership values and darker shades are used for lower membership values. This has the effect of outlining borders where classes are intermixed and has been preferred by physicians.\(^{1}\) The choice of which color to use for which region, and how to shade regions in images, is a seemingly trivial part of segmentation; however, visual display issues greatly affect the utility of computed outputs, so this issue deserves careful attention.\(^{73}\)

B. The hard and fuzzy \( c \)-means algorithms: FCM, AFCM, and SFCM

The most well known objective function for clustering in \( X \) is the classical within-groups sum-of-squared-errors function, defined as\(^{48}\)

\[
J_1(U,\nu;X) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}\|x_k - \nu\|^2,
\]

where \( \nu = (\nu_1, \nu_2, \ldots, \nu_r) \) is a vector of (unknown) cluster centers (weights or prototypes), \( \nu_i \in \mathbb{R}^r \) for \( 1 < i < c \), and \( U \) is a hard or conventional \( c \)-partition of \( X \). Optimal partitions \( U^* \) of \( X \) are taken from pairs \((U^*, \nu^*)\) that are "local minimizers" of \( J_1 \). Generalization of (14) to the infinite family called the fuzzy \( c \)-means functionals is done as follows: Put
\[ J_m(U, P, X) = \sum_{k=1}^{n} \sum_{i=1}^{m} (u_{ik})^m D_{ikA}, \]  

(15)

where

\[ m \in [1, \infty) \] is a weighting exponent on each fuzzy membership,

\[ U \in M_{fcm} \] is a constrained fuzzy c-partition of \( X \),

\[ \mathbf{v}=(v_1, v_2, \ldots, v_c) \] are \( c \) vector prototypes in \( \mathbb{R}^p \),

\[ A= \text{any positive definite} \ (p \times p) \text{ matrix}, \]

\[ D_{ikA} = \| x_k - v_i \|_A = \sqrt{\sum_j (x_{kj} - v_{ij})^2 A_{jj}}. \]  

(16a, 16b, 16c, 16d, 16e)

The same optimization scheme used to find necessary conditions that lead to the ULML method leads to conditions that are necessary to minimize \( J_1 \) and \( J_m \).

**Hard c-Means (HCM) Theorem:** Assume \( D_{ikA} > 0 \), \( \forall i, k \). Then \((U, v)\) may minimize \( J_m \) subject to the constraints shown in (16) for \( m > 1 \) only if

\[ u_{ik} = \begin{cases} 1, & \| x_k - v_i \|_A < \| x_k - v_j \|_A \quad j=1,\ldots,c, j \neq i, \\ 0, & \text{otherwise}, \end{cases} \]

for all \( i, k \),

\[ v_i = \frac{\sum_{k=1}^{n} u_{ik} x_k}{\sum_{k=1}^{n} u_{ik}}, \quad \text{for all } i. \]  

(17a, 17b)

HCM produces a partition \( U \) that contains hard clusters, so each pixel will receive a unique class assignment from this method. The well-known generalization is the following theorem.

**Fuzzy c-Means (FCM) Theorem:** Assume \( D_{ikA} > 0 \), \( \forall i, k \). Then \((U, v)\) may minimize \( J_m \) subject to the constraints shown in (16) for \( m > 1 \) only if

\[ u_{ik} = \left[ \sum_{j=1}^{c} \left( \frac{\| x_k - v_j \|_A}{\| x_k - v_i \|_A} \right)^{2/(m-1)} \right]^{-1}, \quad \text{for all } i, k, \]

\[ v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}, \quad \text{for all } i. \]  

(18a, 18b)

The FCM and HCM procedures approximately minimize \( J_m \) or \( J_1 \) by Picard iteration through (18) or (17), respectively. A brief specification of these procedures follows.

**Hard/Fuzzy c-Means (FCM/HCM) Algorithms:**

\[ \text{cM1. Given unlabeled data set } X = \{x_1, x_2, \ldots, x_n\}. \text{ Fix } c, \]

\[ T, \| \cdot \|_A, \text{ and } e > 0. \]

\[ \text{cM2. Initialize } U_0 \in M_{fcm}. \text{ Choose } m=1 \text{ (HCM)} \text{ or } m>1. \text{ Compute all } (c) \text{ weight vectors } \{v_{i0}\}. \]

\[ \text{cM3. For } t=1,2,\ldots,T, \text{ (a) compute all } (cn) \text{ memberships } \{u_{ik,t}\}; \text{ (b) update all } (c) \text{ weight vectors } \{v_{it+1}\}. \]

\[ \text{cM4. Compute } E_t = \sum_{i=1}^{c} \sum_{t=1}^{T} \| v_{it+1} - v_{it} \|. \]

\[ \text{cM5. If } E_t < e, \text{ stop; else, next } t. \]

FCM has been used by a number of researchers for MR segmentation.\(^{1,17}\) Figures 8(h) and 8(k) are segmentations of the data sets \( X_{ab} \) and \( X_{def} \) derived as above using FCM with \( c=8, \ m=2, \) and the Euclidean distance in (15). The colors shown in Figs. 8(h) and 8(k) were assigned to each region (cluster) on the basis of crisp numerical labels that were derived for each pixel vector \( x_{ij} \) by applying the maximum membership rule to it. That is, the fuzzy partition \( U_{FCM} \) obtained by submitting \( X_{ab} \) or \( X_{def} \) to FCM was subsequently defuzzified using the maximum membership conversion.

**Maximum membership (MM) conversion of \( U \) in \( M_{fcm} \) to \( U_{MM} \) in \( M_{fc} \):**

\[ u_{MM,ij,k} = \begin{cases} 1, & u_{ij,k} > u_{ij,j}, \ 1 < i < c, \ s \neq k, \\ 0, & \text{otherwise,} \end{cases} \]

\[ 1 < k < c, \ 1 < i < m, \ 1 < j < n. \]  

(19)

Tissue class assignments for these segmentations are subsequently done by a human operator, who inspects the segmentations as colored and simply states "orange is csf," etc. The images in Figs. 8(i) and 8(l) arise by applying the color shading scheme for coloring pixels described above,\(^1\) instead of giving each pixel one of \( c \) crisp colors as in (19). In other words, the only difference between the pairs \([8(h), 8(i)]\) and \([8(k), 8(l)]\) is the method used to color each pixel. Figures \([11(e), 11(g)]\) and \([11(f), 11(h)]\) are image pairs of the same kind; that is, FCM, \( c=8, m=2, d=\text{Euclidean on the pre- and post-chemotherapy patient, the only difference in each pair again being the method used to color each pixel. The fuzzy display provides better visual evidence of the tumor in Fig. 8(i), where the tumor is outlined with but one small break in its boundary. In Figs. 11(e) and 11(g) the contrast between crisp and fuzzy display is slight. Figure 11(h), however, provides a slightly darker region in the upper middle tumor area, which may be of interest. Differing color shades on the edges of the tumor area in the middle and the edema surrounding it also provide information about the existence of different tissue types that is not available in the crisp display.**

Recent studies indicate that the approximate FCM (AFCM) and semisupervised FCM (SFMC) algorithms\(^{74,75}\) may be more useful for segmentation than FCM. When \( c, p, \) and/or \( n \) become large, FCM is time consuming. These parameters are in the ranges \( 2 < c < 9, \ 1 < p < 3, \) and \( n=256 \times 256 = 65 \ 536 \) for a typical three-parameter slice of human brain. Moreover, typical MR studies consist of 20–128 slices, each having the dimensions given. Each iteration of FCM on one slice might take, e.g., 30 s of CPU time on a SUN 4. AFCM accelerates FCM by replacing the exponentiations in Eqs. (18a) and (18b) by lookup table approximations.\(^74\) AFCM is defined only for data whose features are discrete: Image intensities are usually such that \( 0 \leq x_{ij} \leq G_L \), where \( G_L \) is the number of gray levels (in our case, \( G_L=256 \) possible values for each feature). AFCM provides (roughly) one order of magnitude speed up for FCM. On the other hand, the optimization problem that AFCM solves is not the one specified by
(15), because roundoff in the AFMC approach may cause the effective value of \( (m) \) in AFMC to vary slightly at each iteration. Consequently, FCM and AFMC may terminate at different pairs \((U^*, v^*)\).\(^74\)

C. Other approaches (Kohonen networks, ART, and hybrids)

Lin et al.\(^76\) have recently discussed the use of CNNs to segment x-ray CT, MR, and positron emission tomography images. The image segmentation problem is formulated as a constraint satisfaction problem, and a neural architecture is used to perform the segmentation (Kohonen's self-organizing feature map is used for the assignment of initial conditions).

Kohonen's learning vector quantization (LVQ) and adaptive resonance theory networks may prove useful for segmentation. Kohonen's network has only input and output layers. There will be an output for each of the expected tissue classes. Training involves modifying the weight vector for each output node that is marked for a given input. One modification of LVQ used for image segmentation updates all weight vectors in the metrical neighborhood of the winning output node.\(^77\)

Adaptive resonance theory (ART) is another unsupervised scheme that was developed by Carpenter and Grossberg.\(^78\) It is intended to make learning stable in the face of irrelevant inputs, adapt to significant inputs, and remember what it has learned, as it learns new information. FFBP models must be constantly reminded of what they have learned as new information is presented. This requires old examples to be saved for re-presentation and is one of the problems the ART family of algorithms is designed to solve. ART2 networks, which allow for continuous or gray-scale inputs, would be most suited to MR image segmentation.\(^79\)

In Ref. 80, a precursor to FCM (a variant of HCM) is used with an information-theoretic measure to provide segmentation of atherosclerosis shown by MRI angiography. Supervised CNNs that self-organize their hidden layers are another promising development.\(^81\) The use of morphological operators with histograms (or counts of pixels with "similar" characteristics) is combined for segmentation in Ref. 82. Shape information and physical parameters of voxels for segmentation have also been tested.\(^83\) Hierarchical clustering based on histograms of pixels has also been tried for segmentation.\(^84\) A mixture of Markov random field theory and ideas from \( k \)-nn and LML have been used for segmentation in Ref. 85. A new edge detection algorithm has also been applied to the segmentation of MR images of the brain.\(^86-87\) The edges found are interpreted as boundaries between tissue types and closed areas are interpreted as (unknown) types of tissue. LML techniques with modifications are explored in several works.\(^88-90\)

V. SUMMARY

This paper has reviewed, with somewhat variable coverage, the nine MR image segmentation techniques itemized in Table II. A wide array of approaches have been discussed; each has its merits and drawbacks. We have also given pointers to other approaches not discussed in depth in this review. The methods reviewed fall roughly into four model groups: \( c \)-means, maximum likelihood, neural networks, and \( k \)-nearest neighbor rules. Both supervised and unsupervised schemes require human intervention to obtain clinically useful results in MR segmentation. Unsupervised techniques require somewhat less interaction on a per patient/image basis.

Maximum likelihood techniques have had some success, but are very susceptible to the choice of training region, which may need to be chosen slice by slice for even one patient. Generally, techniques that must assume an underlying statistical distribution of the data (such as LML and UML) do not appear promising, since tissue regions of interest do not usually obey the distributional tendencies of probability density functions. The most promising supervised techniques reviewed seem to be FF/NN methods that allow hidden layers to be configured as examples are presented to the system. An example of a self-configuring network, FF/CC, was also discussed. The relatively simple \( k \)-nearest neighbor rule algorithms (hard and fuzzy) have also shown promise in the supervised category.

Unsupervised techniques based upon fuzzy \( c \)-means clustering algorithms have also shown great promise in MR image segmentation. Several unsupervised connectionist techniques have recently been experimented with on MR images of the brain and have provided promising initial results. A pixel-intensity-based edge detection algorithm has recently been used to provide promising segmentations of the brain. This is also an unsupervised technique, older versions of which have been susceptible to oversegmenting the image because of the lack of clear boundaries between tissue types or finding uninteresting boundaries between slightly different types of the same tissue.

To conclude, we offer some remarks about improving MR segmentation techniques. The better unsupervised techniques are too slow. Improving speed via parallelization and optimization will improve their competitiveness with, e.g., the \( k \)-nn rule, which is the fastest technique covered in this review. Another area for development is dynamic cluster validity. Unsupervised methods need better ways to specify and adjust \( c \), the number of tissue classes found by the algorithm. Initialization is a third im-
portant area of research. Many of the schemes listed in Table II are sensitive to good initialization, both in terms of the parameters of the design, as well as operator selection of training data. Experience and careful documentation of many MR case studies are needed to make progress in this area. Knowledge-based labeling of regions in unsupervised segmentations can make unsupervised algorithms more powerful by providing consistent coloring of regions. The instability of supervised techniques with one training set across patients and MR slices might be improved by the generation of more globally representative training sets. Finally, the general accuracy of techniques that perform well needs to be further investigated across MRI machines and different types of patient problems.

ACKNOWLEDGMENTS

J.C.B. was supported by NSF Grant No. 1R1-9003252 and NCI Grant No. 1RO1CA61359-01. L.O.H. was supported by NCI Grant No. 1RO1CA61359-01 and by the Whitaker Foundation. L.P.C. was supported by NASA, ACS, and NCI Grant No. 1RO1CA61359-01.


1048 Bezdek, Hall, and Clarke: Review Article: Image segmentation using pattern recognition


58] DARPA Neural Network Study (AFCEA, Fairfax, VA, 1988).


