DEPARTMENT OF THE NAVY

OFFICE OF COUNSEL
NAVAL UNDERSEA WARFARE CENTER DIVISION
1176 HOWELL STREET
NEWPORT RI 02841-1708

IN REPLY REFER TO:

Attorney Docket No. 84513
Date: 30 March 2005

The below identified patent application is available for licensing. Requests for information should be addressed to:

PATENT COUNSEL
NAVAL UNDERSEA WARFARE CENTER
1176 HOWELL ST.
CODE 00OC, BLDG. 112T
NEWPORT, RI 02841

Serial Number 10/789,050

Filing Date 27 February 2004

Inventor Paul M. Baggenstoss

If you have any questions please contact James M. Kasischke, Patent Counsel, at 401-832-4736.

Attorney Docket No. 84513
Customer No. 23523

CHAIN RULE PROCESSOR

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN THAT PAUL M. BAGGENSTOSS, citizen of the United States of America, employee of the United States Government and resident Newport, County of Newport, State of Rhode Island, has invented certain new and useful improvements entitles as set forth above of which the following is a specification:

JAMES M. KASISCHKE, ESQ.
Reg. No. 36562
Naval Undersea Warfare Center
Division Newport
Newport, RI 02841-1708
TEL: 401-832-4736
FAX: 401-832-1231

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as U.S. EXPRESS MAIL, Mailing Label No. EV326644774US In envelope addressed to: Commissioner for Patents, Alexandria, VA 22313 on 27 Feb 2004
(DATE OF DEPOSIT)

James M. Kasischke
APPLICANT'S ATTORNEY

27 Feb 2004
DATE OF SIGNATURE

Attorney Docket No. 84513

CHAIN RULE PROCESSOR

STATEMENT OF GOVERNMENT INTEREST

The invention described herein may be manufactured and used by or for the Government of the United States of America for governmental purposes without the payment of any royalties thereon or therefor.

BACKGROUND OF THE INVENTION

(1) Field of the Invention

This invention generally relates to a signal classification system for classifying an incoming data stream. More particularly, the invention relates to a modularized classifier system that can be used for easily assembling different classifiers.

(2) Description of the Prior Art

In order to determine the nature of an incoming signal, the signal type must be determined. A classifier attempts to classify a signal into one of M signal classes based on features in the data. M-ary classifiers utilize neural networks for extracting these features from the data. In a training stage the neural networks incorporated in the classifier are trained with labeled data allowing the neural networks to learn the patterns associated with each of the M classes. In a testing stage, the classifier is tested against unlabeled data based on the learned patterns. The performance of the classifier is defined as the probability that a signal is correctly classified.

The so-called M-ary classification problem is that of assigning a multidimensional sample of data $x \in R^N$ to one of M classes. The statistical hypothesis that class j is true is denoted by H_j, $1 \leq j \leq M$. The statistical characterization of x under each of the M hypotheses is described completely by the probability density functions (PDFs), written $p(x|H_j), 1 \leq j \leq M$. Classical theory as applied to the problem results in the so-called Bayes classifier, which simplifies to the Neyman-Pearson rule for equiprobable prior probabilities:

$$j^* = \arg\max_j p(x|H_j). \tag{1}$$

Because this classifier attains the minimum probability of error of all possible classifiers, it is the basis of most classifier designs. Unfortunately, it does not provide simple solutions to the dimensionality problem that arises when the PDFs are unknown and must be estimated. The most common solution is to reduce the dimension of the data by extraction of a small number of information-bearing features $z = T(x)$, then recasting the classification problem in terms of z:

$$j^* = \arg\max_j p(z|H_j). \tag{2}$$

This leads to a fundamental tradeoff: whether to discard features in an attempt to reduce the dimension to something manageable or to include them and suffer the problems associated with estimating a PDF at high dimension. Unfortunately, there may be no acceptable compromise. Virtually all methods which attempt to find decision boundaries on a high-dimensional space are subject to this tradeoff or "curse" of dimensionality. For this reason, many researchers have explored the possibility of using class-specific features.

The basic idea in using class-specific features is to extract M class-specific feature sets $z_j = T_j(x)$, $1 \le j \le M$ where the dimension of each feature set is small, and then to arrive at a decision rule based only upon functions of the lower dimensional features. Unfortunately, the classifier modeled on the Neyman-Pearson rule

$$j^* = \arg\max_j p(z_j | H_j). \quad (3)$$

is invalid because comparisons of densities on different feature spaces are meaningless. One of the first approaches that comes to mind is to computes for each class a likelihood ratio against a common hypothesis composed of "all other classes." While this seems beneficial on the surface, there is no theoretical dimensionality reduction since for each likelihood ratio to be a sufficient statistic, "all features" must be included when testing each class against a hypothesis that includes "all other classes." A number of other approaches have emerged in recent years to arrive at meaningful decision rules. Each method makes a strong assumption (such as that the classes fall into linear subspaces) that limits the applicability of the method or else uses *ad hoc* method of combining the likelihoods of the various feature sets.

Prior art methods include the following. A method used in speech recognition (Frimpong-Ansah, K. Pearce, D. Holmes, and W. Dixon, "A stochastic/feature based recognizer and its training algorithm," in *Proc. ICASSP*, vol. 1, 1989, pp. 401-404.) uses phoneme-specific features. While, at first, this method appears to use class-specific features, it is actually using the same features extracted from the raw data but applying different models to the time evolution of these features.

A method of image recognition (E. Sali and S. Ullman, "Combining class-specific fragments for object classification," in *Proc. British Machine Vision Conf.*, 1999, pp. 203-213.) uses class-specific features to detect various image "fragments." The method uses a nonprobabilistic means of combining fragments to form an image.

A method has been proposed that tests all pairs of classes (S. Kumar, J. Ghosh, and M. Crawford, "A versatile framework for labeling imagery with large number of classes," in *Proc. Int. Joint Conf. Neural Networks*, Washington, DC, 1999, pp. 2829-2833.). To be exhaustive, this method has a complexity of $0(M^2)$ different tests and may be prohibitive for large M. A hierarchical approach has been proposed based on a binary tree of tests ("A hierarchical multiclassifier system for hyperspectral data analysis," in *Multiple Classifier Systems*, J. Kittler and F. Roli, Eds. New York: Springer, 2000, pp. 270-279). Implementation of the binary tree requires initial classification into meta-classes, which is an approach that is suboptimal because it makes hard decisions based on limited information.

Methods based on linear subspaces (H. Watanabe, T. Yamaguchi, and S. Katagiri, "Discriminative metric design for robust pattern recognition," *IEEE Trans. Signal Processing*, vol. 45, pp. 2655-2661, Nov. 1997. P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 711-720, July 1997.) are popular because they use the powerful tool of linear subspace analysis. These methods can perform well in certain applications but are severely limited to problems where when the classes are separable by linear processing.

Support vectors (D. Sebald, "Support vector machines and the multiple hypothesis test problem," *IEEE Trans. Signal Processing*, vol. 49, pp. 2865-2872, Nov. 2001.) are a relatively new approach that is based on finding a linear decision function between every pair of classes.

The inventor has also developed a prior class specific classifier, U.S. Patent No. 6,535,641, showing a class specific classifier for classifying data received from a data source. The classifier has a feature transformation section associated with each class of data which receives the data and provides a feature set for the associated data class. Each feature transformation section is joined to a pattern matching processor which receives the associated data class feature set. The pattern matching processors calculate likelihood functions for the associated data class. One normalization processor is joined in parallel with each pattern matching processor for calculating an inverse likelihood function from the data, the associated class feature set and a common data class set. The common data class set can be either calculated in a common data class calculator or incorporated in the normalization calculation. The inverse likelihood function is then multiplied with the likelihood function for each associated data class. A comparator provides a signal indicating the appropriate class for the input data based upon the highest multiplied result.

As evidenced by the various approaches, there is a strong motivation for using class-specific features. Unfortunately, classical theory as it stands requires operating in a common feature space and fails to provide any guidance for a suitable class-specific architecture.

SUMMARY OF THE INVENTION

Therefore, it is one purpose of this invention to provide a class specific classifier.

Another purpose of this invention is a classifier architecture having reusable modules.

Accordingly, there is provided a modularized classifier which includes a plurality of class specific modules. Each module has a feature calculation section, and a correction section. The modules can be arranged in chains of modules where each chain is associated with a class. The first module in the chain receives raw input data and subsequent modules act on the features provided by the previous module. The correction section acts on the previously computed correction. Each chain is terminated by a probability density function evaluation module. The output of the evaluation module is combined with the correction value of the last module in the chain. This combined output is provided to a compare module that indicates the class of the raw input data. The invention may be implemented either as a device or a method operating on a computer.

BRIEF DESCRIPTION OF THE DRAWINGS

The appended claims particularly point out and distinctly claim the subject matter of this invention. The various objects, advantages and novel features of this invention will be more fully apparent from a reading of the following detailed description in conjunction with the accompanying drawings in which like reference numerals refer to like parts, and in which:

FIG. 1 is a diagram illustrating the chain rule used in this invention;

FIG. 2 is a block diagram of a first example of a classifier implemented utilizing the preferred architecture of the current invention;

FIG. 3 is a block diagram of a second example of a classifier implemented utilizing an alternative architecture of the current invention; and

FIG. 4 is a block diagram of an embodiment of a classifier implemented utilizing another alternative architecture of the current invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

It is well known how to write the PDF of x from the PDF of z when the transformation is 1:1. This is the change of variables theorem from basic probability. Let $z=T(x)$, where $T(x)$ is an invertible and differentiable multidimensional transformation. Then,

$$p_x(x) = |J(x)| p_z(T(x)), \tag{4}$$

where $|J(x)|$ is the determinant of the Jacobian matrix of the transformation

$$J_{ij} = \frac{\partial_{zi}}{\partial_{xj}}. \tag{5}$$

What we seek is a generalization of (4) which is valid for many-to-1 transformations. Define

$$P(T, p_z) = \{p_x(x) : z = T(x) \; and \; z \sim p_z(z)\}, \tag{6}$$

that is, $P(T,p_z)$ is the set of PDFs $p_x(x)$ which through $T(x)$ generates PDF $p_z(z)$ on z. If T( ) is many-to-one, $P(T,p_z)$ will contain more than one member. Therefore, it is impossible to uniquely determine $p_x(x)$ from T() and $p_z(z)$. We can, however, find a particular solution if we constrain $p_x(x)$ such that for every transform pair (x,z), we have:

$$\frac{p_x(x)}{p_x(x|H_0)}=\frac{p_z(z)}{p_z(z|H_0)}, \tag{7}$$

or that the likelihood ratio (with respect to H_0) is the same in both the raw data and feature domains for some pre-determined reference hypothesis H_0. We will soon show that this constraint produces desirable properties. The particular form of $p_x(x)$ is uniquely defined by the constraint itself, namely

$$p_x(x)=\frac{p_z(x|H_0)}{p_z(z|H_0)}p_z(z);\ at\ z=T(x). \tag{8}$$

The PDF projection theorem proves that (8) is, indeed, a PDF and a member of $P(T,p_z)$. Under this theorem let H_0 be some fixed reference hypothesis with known PDF $p_x(x|H_0)$. Let χ be the region of support of $p_x(x|H_0)$. In other words χ is the set of all points x where $p_x(x|H_0) > 0$. Let $z = T(x)$ be a continuous many-to-one transformation (the continuity requirement may be overly restrictive). Let Z be the image of χ under the transformation T(x). Let $p_z(z|H_0)$ be the PDF of z when x is drawn from $p_x(x|H_0)$. It follows that $p_z(z|H_0) > 0$ for all $z \in Z$. Now, let be a any other PDF with the same region of support Z. Then the function (8) is a PDF on χ, thus

$$\int_{x\in\chi} p_x(x)\, dx = 1. \tag{9}$$

Furthermore, $p_x(x)$ is a member of $P(T, p_z)$.

The theorem shows that, provided we know the PDF under some reference hypothesis H_0 at both the input and output of transformation $T(x)$, if we are given an arbitrary PDF $p_z(z)$ defined on z, we can immediately find a PDF $p_x(x)$ defined on x that generates $p_z(z)$. Although it is interesting that $p_x(x)$ generates $p_z(z)$, there are an infinite number of them, and it is not yet clear that $p_x(x)$ is the best choice. However, suppose we would like to use $p_x(x)$ as an approximation to the PDF $p_x(x|H_1)$. Let this approximation be

$$\hat{p}_x(x|H_1) \equiv \frac{p_x(x|H_0)}{p_z(z|H_0)}\, \hat{p}_z(z|H_1) \text{ at } z = T(x)\,. \tag{10}$$

From the PDF projection theorem, we see that (10) is a PDF. Furthermore, if $T(x)$ is a sufficient statistic for H_1 vs H_0, then as $\hat{p}_z(z|H_1) \rightarrow p_z(z|H_1)$, we have

$$\hat{p}_x(x|H_1) \rightarrow p_x(x|H_1). \tag{11}$$

This is immediately seen from the well-known property of the likelihood ratio, which states that if *T(x)* is sufficient for H_1 versus H_0:

$$\frac{p_x(x|H_1)}{p_x(x|H_0)}=\frac{p_z(z|H_1)}{p_z(z|H_0)} \qquad (12)$$

Note that for a given H_1, the choice of *T(x)* and H_0 are coupled so that they must be chosen *jointly*. In addition, note that the sufficiency condition is required for optimality, but is not necessary for (10) to be a valid PDF. Here, we can see the importance of the theorem. The theorem, in effect, provides a means of creating PDF approximations on the high-dimensional input data space without dimensionality penalty using low-dimensional feature PDFs and provides a way to optimize the approximation by controlling both the reference hypothesis H_0 as well as the features themselves. This is the remarkable property of the theorem: that the resulting function remains a PDF whether or not the features are sufficient statistics. Since sufficiency means optimality of the classifier, approximate sufficiency means PDF approximation and approximate optimality.

The PDF projection theorem allows maximum likelihood (ML) methods to be used in the raw data space to optimize the accuracy of the approximation over T and H_0 as well as θ. Let $\hat{p}_z(z|H_1)$ be parameterized by the parameter θ. Then, the maximization

$$\max_{\theta,T,H_0}\left\{\frac{p_x(x|H_0)}{p_z(z|H_0)}\hat{p}_z(z|H_1;\theta), z=T(x)\right\} \tag{13}$$

is a valid ML approach and can be used for model selection (with appropriate data cross-validation).

We now mention a useful property of (7). Let $\boldsymbol{H}_z$ be a *region of sufficiency* (ROS) of z, which is defined as a set of all hypotheses such that for every pair of hypotheses $H_{0a}, H_{0b} \in \boldsymbol{H}_z$, we have

$$\frac{p_x(x|H_{0a})}{p_x(x|H_{0b})}=\frac{p_z(z|H_{0a})}{p_z(z|H_{0b})} \tag{14}$$

An ROS may be thought of as a family of PDFs traced out by the parameters of a PDF, where z is a sufficient statistic for the parameters. The ROS may or may not be unique. For example, the ROS for a sample mean statistic could be a family of Gaussian PDFs with variance 1 traced out by the mean parameter. Another ROS would be produced by a different variance. The " j-function"

$$J(x,T,H_0)\equiv\frac{p_x(x|H_0)}{p_z(T(x)|H_0)}=\frac{p(x|H_0)}{p(z|H_0)} \tag{15}$$

is independent of H_0 as long as H_0 remains within ROS $\boldsymbol{H}_z$. Defining the ROS should in no way be interpreted as a sufficiency requirement for z. All statistics z have an ROS that may or may not include H_1 (it does only in the ideal case). Defining $\boldsymbol{H}_z$ is used only in determining the allowable range of reference hypotheses when using a data-dependent reference hypothesis. For example, let z be the sample variance of x. Let $H_0(\sigma^2)$ be the hypothesis that x is a set of N independent identically distributed zero-mean Gaussian samples with variance σ^2. Clearly, an ROS for z is the set of all PDFs traced out by σ^2. We have

$$p\left(x \middle| H_0\left(\sigma^2\right)\right) = \left(2\pi\sigma^2\right)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N} x_i^2\right\} \qquad (16)$$

and, since z is a $\chi^2(N)$ random variable (scaled by 1/N)

$$p\left(z \middle| H_0\left(\sigma^2\right)\right) = \frac{N}{\sigma^2\Gamma\left(\frac{N}{2}\right)} 2^{-N/2}\left(\frac{N_z}{\sigma^2}\right)^{N/2-1} exp\left(-\frac{zN}{2\sigma^2}\right). \qquad (17)$$

It is easily verified that the contribution of σ^2 is canceled in the J-function ratio.

Because $J\left(x, T, H_0\left(\sigma^2\right)\right)$ is independent of σ^2, it is possible to make σ^2 a function of the data itself, changing it with each input sample. In the example above, since z is the sample variance, we could let the assumed variance under H_0 depend on z according to $\sigma^2 = z$.

However, if $J\left(x, T, H_0\left(\sigma^2\right)\right)$ is independent of σ^2, one may question what purpose does it serve to vary σ^2. The reason is purely numerical. Note that in general, we do not have an analytic form for the J-function but instead have separate numerator and denominator terms. Often, computing $J\left(x, T, H_0\left(\sigma^2\right)\right)$ can pose some tricky numerical problems, particularly if x and z are in the tails of the respective PDFs. Therefore, our approach is to position H_0 to maximize the numerator PDF (which simultaneously maximizes the denominator). Another reason to do this is to allow PDF approximations to be used in the denominator that are not valid in the tails, such as the central limit theorem (CLT).

In our example, the maximum of the numerator clearly happens at $\sigma^2 = z$ because z is the maximum likelihood estimator of σ^2. We will explore the relationship of this method to asymptotic ML theory in a later section. To reflect the possible dependence of H_0 on z, we adopt the notation $H_0(z)$. Thus

$$\hat{p}_x(x|H_1) \equiv \frac{p_x(x|H_0(z))}{p_z(z|H_0(z))}\,\hat{p}_z(z|H_1), \text{ where } z = T(x) . \qquad (18)$$

The existence of z on the right side of the conditioning operator $|$ is admittedly a very bad use of notation but is done for simplicity. The meaning of z can be understood using the following imaginary situation. Imagine that we are handed a data sample x, and we evaluate (10) for a particular hypothesis $H_0 \in \boldsymbol{H}_z$. Out of curiosity, we try it again for a different hypothesis of $H_0' \in \boldsymbol{H}_z$. We find that no matter which $H_0 \in \boldsymbol{H}_z$ we use, the result is the same. We notice, however, that for an H_0 that produces larger values of $p_x(x|H_0(z))$ and $p_z(z|H_0(z))$, the requirement for numerical accuracy is less stringent. It may require fewer terms in a polynomial expansion or else fewer bits of numerical accuracy. Now, we are handed a new sample of x, but this time, having learned our lesson, we immediately choose the $H_0 \in \boldsymbol{H}_z$ that maximizes $p_x(x|H_0(z))$. If we do this every time, we realize that H_0 is now a function of z. The dependence, however, carries no statistical meaning and only has a numerical interpretation. This is addressed below in the text differentiating a fixed reference hypothesis from a variable reference hypothesis.

In many problems $\boldsymbol{H}_z$ is not easily found, and we must be satisfied with *approximate* sufficiency. In this case, there is a weak dependence of $J(x, T, H_0)$ upon H_0. This dependence is generally unpredictable unless, as we have suggested, $H_0(z)$ is always chosen to maximize the numerator PDF. Then, the behavior of $J(x, T, H_0)$ is somewhat predictable. Because the numerator is always maximized, the result is a positive bias. This positive bias is most notable when there is a good match to the data, which is a desirable feature.

We have stated that when we use a data-dependent or variable reference hypothesis, we prefer to choose the reference hypothesis such that the numerator of the J-function is a maximum. Since we often have parametric forms for the PDFs, this amounts to finding the ML estimates of the parameters. If there are a small number of features, all of the features are ML estimators for parameters of the PDF, and there is sufficient data to guarantee that the ML estimators fall in the asymptotic (large data) region, then the variable hypothesis approach is equivalent to an existing approach based on classical asymptotic ML theory. We will derive the well-known asymptotic result using (18).

Two well-known results from asymptotic theory are the following. First, subject to certain regularity conditions (large amount of data, a PDF that depends on a finite number of parameters and is differentiable, etc.), the PDF $p_x(x; \theta^*)$ may be approximated by

$$p_x(x; \theta^*) \cong p_x\left(x; \hat{\theta}\right) \exp\left\{-\frac{1}{2}\left(\theta^* - \hat{\theta}\right)' I\left(\hat{\theta}\right)\left(\theta^* - \hat{\theta}\right)\right\} \quad (19)$$

Where θ^* is an arbitrary value of the parameter $\hat{\theta}$ is the maximum likelihood estimate (MLE) of θ, and $I(\theta)$ is the *Fisher's information matrix* (FIM). The components of the FIM for PDF parameters θ_k, θ_t are given by

$$I_{\theta_k,\theta_t}(\theta) = -E\left(\frac{\partial^2 Inp_x(x; \theta}{\partial\theta_k \partial\theta_t}\right). \quad (20)$$

The approximation is valid only for θ^* in the vicinity of the MLE (and the true value). Second, the MLE $\hat{\theta}$ is approximately Gaussian with mean equal to the true value θ and covariance equal to $I^{-1}(\theta)$ or

$$p_\theta\left(\hat{\theta};\theta\right) \cong (2\pi)^{-P/2}\left|I\left(\hat{\theta}\right)\right|^{1/2} \exp\left\{-\frac{1}{2}\left(\theta-\hat{\theta}\right)' I\left(\hat{\theta}\right)\left(\theta-\hat{\theta}\right)\right\} \tag{21}$$

where P is the dimension of θ. Note that we use $\hat{\theta}$ in evaluating the FIM in place of θ, which is unknown. This is allowed because $I^{-1}(\theta)$ has a weak dependence on θ. The approximation is valid only for θ in the vicinity of the MLE.

To apply (18), $\hat{\theta}$ takes the place of z, and $H_0(z)$ is the hypothesis that $\hat{\theta}$ is the true value of θ. We substitute (19) for $p_x(x|H_0(z))$ and (21) $p_z(z|H_0(z))$. Under the stated conditions, the exponential terms in approximations (19), and (21) become 1. Using these approximations, we arrived at

$$\hat{p}_x(x|H_1) = \frac{p_x\left(x;\hat{\theta}\right)}{(2\pi)^{-P/2}\left|I\left(\hat{\theta}\right)\right|^{1/2}}\,\hat{p}_\theta\left(\hat{\theta}|H_1\right) \tag{22}$$

which agrees with the PDF approximation from asymptotic theory.

To compare (18) and (22), we note that for both, there is an implied sufficiency requirement for z and $\hat{\theta}$, respectively. Specifically, $H_0(z)$ must remain in the ROS of z, whereas $\hat{\theta}$ must be asymptotically sufficient for θ. However, (18) is more general since (22) is valid only when *all* of the features are ML estimators and only holds asymptotically for large data records with the implication that $\hat{\theta}$ tends to Gaussian, whereas (18) has no such implication. This is particularly important in upstream processing, where there has not been significant data reduction, and asymptotic results do not apply. Using (18), we can make simple adjustments to the reference hypothesis to match the data better and avoid the PDF tails (such as controlling variance), where we are certain that we remain in the ROS of z. As an aside, we note that (10) with a fixed reference hypothesis is even more general since there is no implied sufficiency requirement for z.

In many cases, it is difficult to derive the J-function for an entire processing chain. On the other hand, it may be quite easy to do it for one stage of processing at a time. In this case, the chain rule can be used to good advantage. The chain rule is just the recursive application of the PDF projection theorem. For example, consider a processing chain

$$x \xrightarrow{T_1(x)} y \xrightarrow{T_2(y)} w \xrightarrow{T_3(w)} z \qquad (23)$$

The recursive use of (10) gives

$$p_x(x|H_1) = \frac{p_x(x|H_0(y))}{p_y(y|H_0(y))} \frac{p_y(y|H_0'(w))}{p_w(w|H_0'(w))} \cdot \frac{p_w(w|H_0''(z))}{p_z(z|H_0''(z))} p_z(z|H_1) \qquad (24)$$

where $y = T_1(x)$, $w = T_2(y)$, $z = T_3(w)$, and $H_0(y)$, $H_0'(w)$, $H_0''(z)$ are reference hypotheses (possibly data-dependent) suited to each stage in the processing chain. By defining the J-function of each stage, we may write the above as

$$\begin{aligned} p_x(x|H_1) = J(x,T_1,H_0(y))J(y,T_2,H_0'(w)) \\ J(w,T_3,H_0''(z))p_z(z|H_1). \end{aligned} \tag{25}$$

There is a special embedded relationship between the hypotheses. Let $\boldsymbol{H}_y$, $\boldsymbol{H}_w$, and $\boldsymbol{H}_z$ be the ROSs of y, w, and z, respectively. Then, we have $\boldsymbol{H}_z \subset \boldsymbol{H}_w \subset \boldsymbol{H}_y$. If we use variable reference hypotheses, we also must have $H_0''(z) \in \boldsymbol{H}_z, H_0'(w) \in \boldsymbol{H}_w, \text{ and } H_0(y) \in \boldsymbol{H}_y$. This embedding of the hypotheses is illustrated in FIG. 1. The condition $H_1 \in \boldsymbol{H}_z$ is the ideal situation and is not necessary to produce a valid PDF. The factorization (24), together with the embedding of the hypotheses, we call the chain-rule processor (CRP).

We now summarize the various methods we have discussed for computing the J-function. For modules using a fixed reference hypothesis, care must be taken in calculation of the J-function because the data is more often than not in the tails of the PDF. For fixed reference hypotheses, the J-function is

$$J(x, T, H_0) = \frac{p_x(x|H_0)}{p_z(z|H_0)}. \tag{26}$$

The numerator density is usually of a simple form, so it is known exactly. The denominator density $p_z(z|H_0)$ must be known exactly or approximated carefully so that it is accurate even in the far tails of the PDF. The saddlepoint approximation (SPA) provides a solution for cases when the exact PDF cannot be derived but the exact moment-generating function is known. The SPA is known to be accurate in the far tails of the PDF.

For a variable reference hypotheses, the J-function is

$$J(x, T, H_0(z)) = \frac{p_x(x|H_0(z))}{p_z(z|H_0(z))}. \quad (27)$$

Modules using a variable reference are usually designed to position the references hypothesis at the peak of the denominator PDF, which is approximated by the CLT.

A special case of the variable reference hypothesis approach is the ML method, when z is an MLE. Whenever the feature is also a ML estimate and the asymptotic results apply (the number of estimated parameters is small and the amount of data is large), the two methods are identical. The variable reference hypothesis method is more general because it does not need to rely on the CLT.

One-to-one transformations do not change the information content of the data, but they are important for feature conditioning prior to PDF estimation. Recall from that the PDF projection theorem is a generalization of the change-of-variables theorem for 1:1 transformations. Thus, for 1:1 transformations, the J-function reduces to the absolute value of the determinant of the Jacobian matrix (4)

$$J(x,T) = |\mathbf{J}_T(x)| \quad (28)$$

Application of the PDF projection theorem to classification is performed by substituting (18) into (1). In other words, we implement the classical Neyman-Pearson classifier but with the class PDFs factored using the PDF projection theorem

$$j^* = \arg\max_j \frac{p_x\left(x \middle| H_{0,j}(z_j)\right)}{p_z\left(z \middle| H_{0,j}(z_j)\right)} \hat{p}_z\left(z_j \middle| H_j\right) \text{ at } z_j = T_j(x) \qquad (29)$$

where we have allowed for class-dependent, variable, reference hypotheses.

FIG. 2 shows an example of a classifier 10 constructed with the architecture of the current invention. Raw data X having a plurality of time samples and falling into a plurality of classes is provided to the classifier 10. Raw data X is provided to chains 11 of class-specific modules 12. Each class is associated with a chain 11 of class-specific modules 12.

Each module 12 receives a feature calculation input which it provides to a feature calculation section 14. The feature calculations section performs calculations on the feature calculation input. The feature calculation input can be data or previously computed features from previous feature calculation outputs. Upon completing these calculations the module 12 provides a feature calculation output. Each module 12 also includes a Log J-Function section 16. The Log J-Function section 16 computes a correction factor that can be summed at summer 18 with the correction factors provided by the correction output of Log J-Function sections 16 in previous modules 12 to allow chaining of modules 12.

Modules 12 are joined in chains so that the first module in the chain receives raw data X at its feature calculation input and zero or a null value at its correction input. Each succeeding module 12 then receives its inputs from the preceding module 12 in the chain 11. Chain 11 can have any number of modules 12. The last module 12 in the chain 11 is joined to a probability density function evaluation section 20. The probability density function evaluation section 20 receives the feature calculation output from the last module in the chain and converts it into a form for summing at summer 22 with the correction output of the last module 12 in the chain 11. The output of summer 22 applies the probability density function for the class associated with the chain 11 to the raw data and produces a value indicating the likelihood that the raw data is a member of the class. A compare module 24 is joined to the output of each summer 22. The compare module 24 provides an output that indicates that the raw data X is of the class having features indicated by high values at the outputs of summers 22.

Class specific modules 12 have been built for feature transformations including various invertible transformations, spectrograms, arbitrary linear functions of exponential random values, the autocorrelation function (contiguous and non-contiguous), autoregressive parameters, cepstrum, order statistics of independent random values, and sets of quadratic forms. These represent some of the many feature transformations that can be incorporated as modules in a classifier built using the chain rule.

FIG. 3 shows an example of a classifier 10' constructed using an alternative embodiment of the architecture of the current invention. This architecture utilizes a J-Function 26, instead of a Logarithmic J-Function 18, in each module 12. This J-Function can be multiplied with the previous correction outputs at multiplier 30. The probability function evaluation section 34 can then provide an output which can be multiplied at 32 with the output of the last module. The multiplied output can then be used as the probability density function for the feature.

FIG. 4 is another alternate embodiment 10" of a classifier utilizing the architecture taught by the current invention. In this embodiment, a thresholding module 36 is provided for each class between summer 22 and compare module 24. Thresholding module 36 does not allow summer 22 to send a value to compare module 24 if the value does not exceed a threshold value. This threshold value can be set as one value for all of the chains or set independently for each chain. The threshold value can be calculated based on the level of background noise in the raw input data. Use of thresholding modules 36 allows weak samples to be ignored rather than forcing them into a poorly fitting class. While thresholding is shown applied to the log J-function embodiment, it can also be applied to the J-function embodiment of the invention.

The J-function and the feature PDF provide a factorization of the raw data PDF into trained and untrained components. The ability of the J-function to provide a "peak" at the "correct" feature set gives the classifier a measure of classification performance without needing to train. In fact, it is not uncommon that the J-function dominates, eliminating the need to train at all. This we call the *feature selectivity effect*. For a fixed amount of raw data, as the dimension of the feature set decreases, indicating a larger rate of data compression, the effect of the J-function compared with the effect of the feature PDF increases. An example where the J-function dominates is a bank of matched filter for known signals in noise. If we regard the matched filters as feature extractors and the matched filter outputs as scalar features, it may be shown that this method is identical to comparing only the J-functions. Let $z_j = |\mathbf{w'}_j \mathbf{x}|^2$, where $\mathbf{w}_j$ is a normalized signal template such that $\mathbf{w}'_j \mathbf{w}_j = 1$. Then, under the white (independent) Gaussian noise (WGN) assumption, z_j is distributed $\chi^2(1)$. It is straightforward to show that the J-function is a monotonically increasing function of z_j. Signal waveforms can be reliably classified using only the J-function and ignoring the PDF of under each hypothesis. The curse of dimensionality can be avoided if the dimension of z_j is small for each j. This possibility exists, even in complex problems, because z_j is required only to have information sufficient to separate class H_j from a specially chosen reference hypothesis $H_{0,j}$.

This invention has been disclosed in terms of certain embodiments. It will be apparent that many modifications can be made to the disclosed apparatus without departing from the invention. Therefore, it is the intent of the appended claims to cover all such variations and modifications as come within the true spirit and scope of this invention.

Attorney Docket No. 84513

CLASS SPECIFIC CLASSIFIER

ABSTRACT OF THE DISCLOSURE

A modularized classifier is provided which includes a plurality of class specific modules. Each module has a feature calculation section, and a correction section. The modules can be arranged in chains of modules where each chain is associated with a class. The first module in the chain receives raw input data and subsequent modules act on the features provided by the previous module. The correction section acts on the previously computed correction. Each chain is terminated by a probability density function evaluation module. The output of the evaluation module is combined with the correction value of the last module in the chain. This combined output is provided to a compare module that indicates the class of the raw input data.

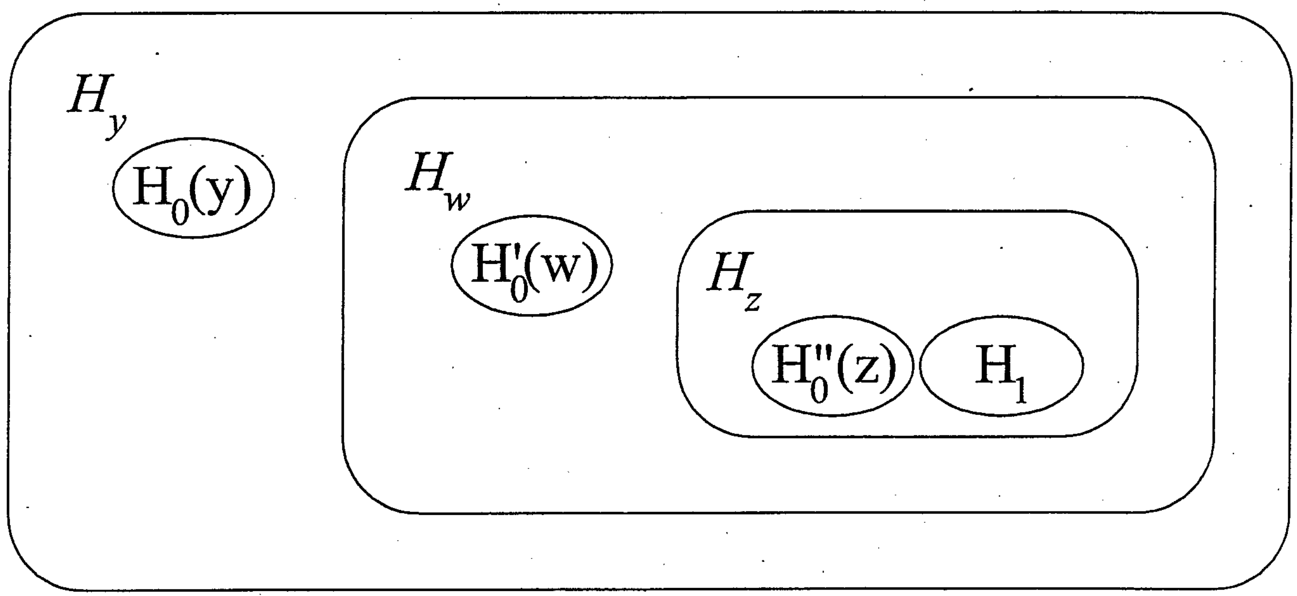
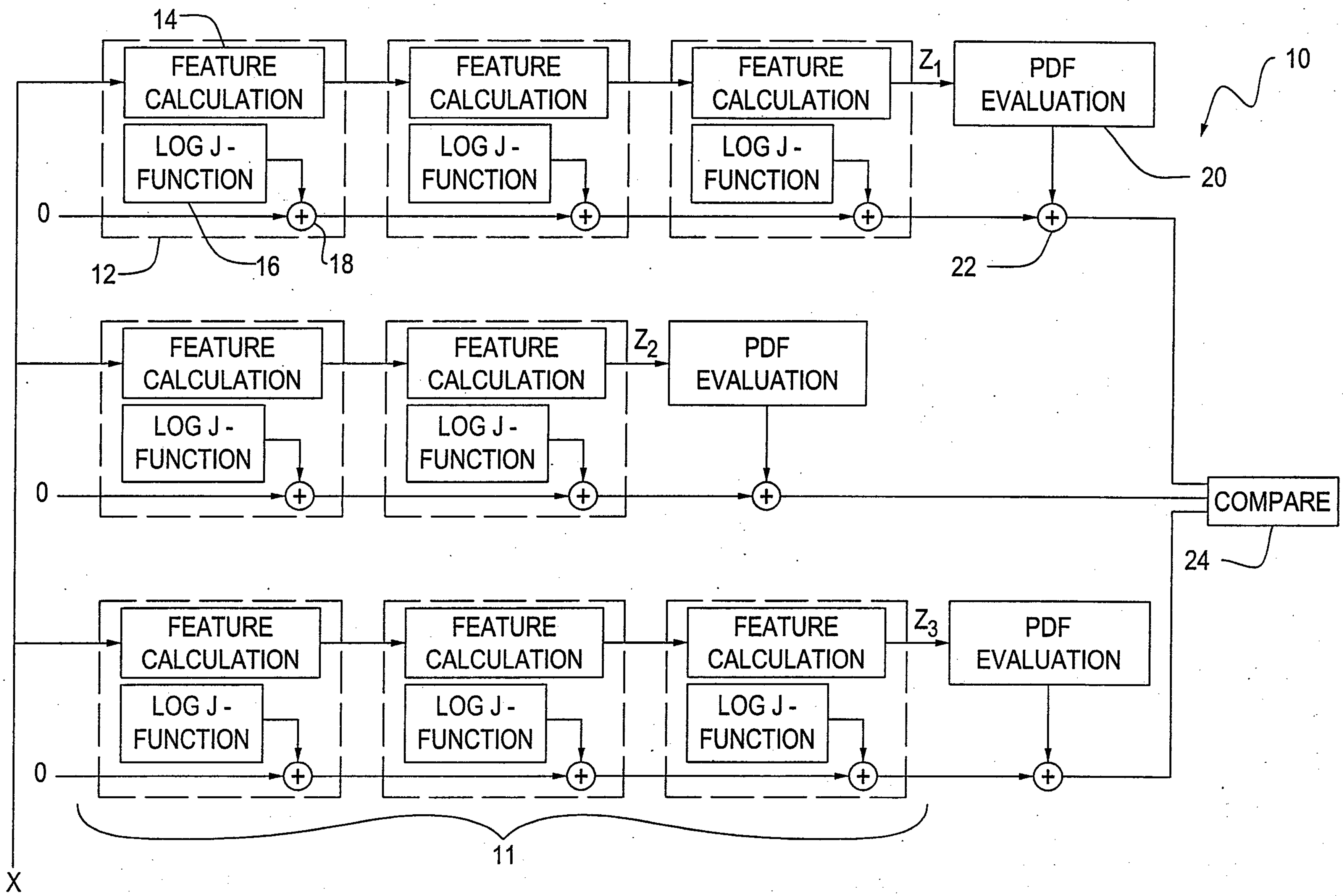


FIG. 1



14
FEATURE CALCULATION
LOG J - FUNCTION
12
16
18
0
Z1
Z2
Z3
PDF EVALUATION
10
20
22
COMPARE
24
11
X

FIG. 2

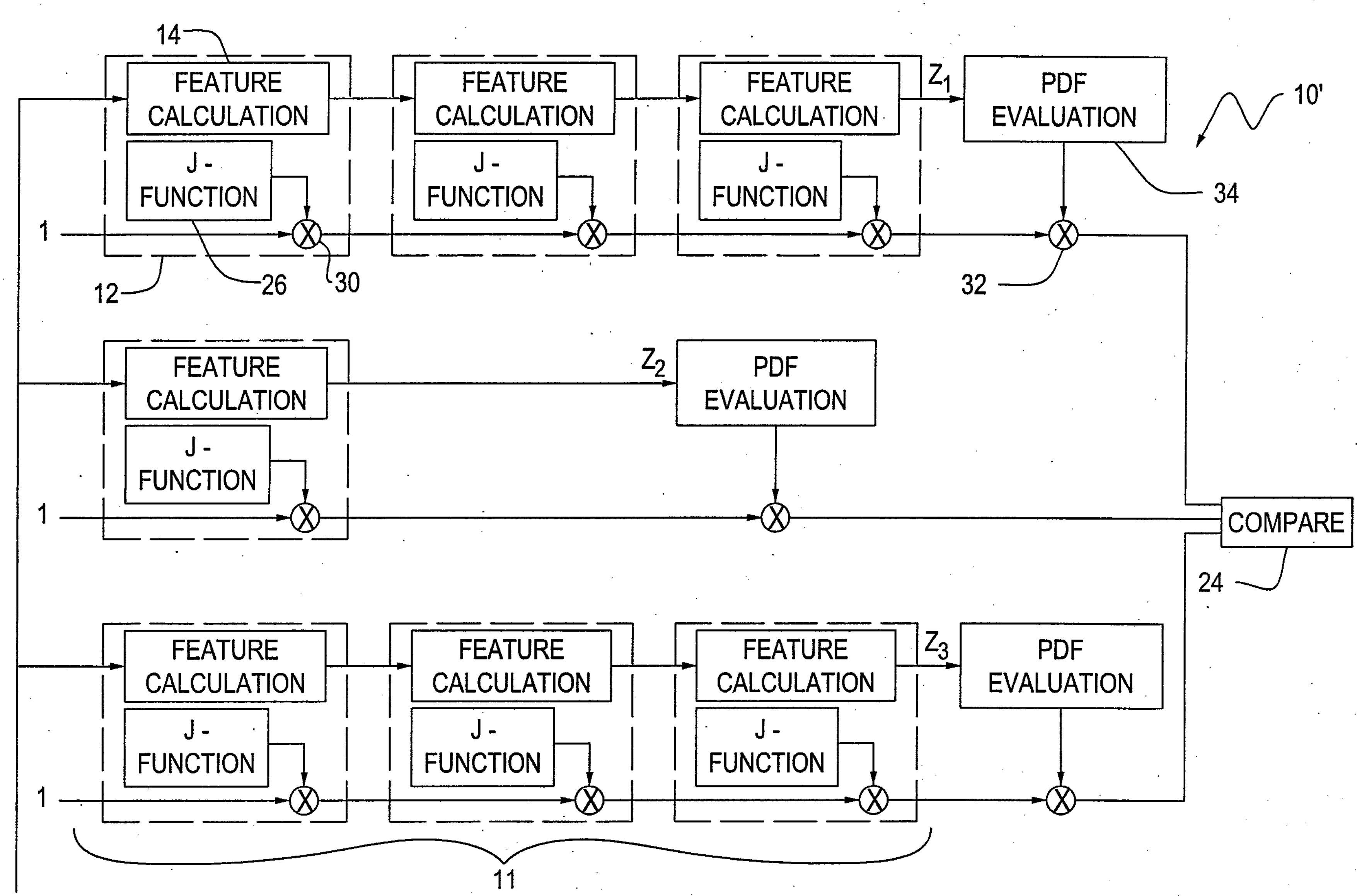

10'
14
FEATURE CALCULATION
J - FUNCTION
FEATURE CALCULATION
J - FUNCTION
FEATURE CALCULATION
J - FUNCTION
Z1
PDF EVALUATION
34
1
12
26
30
32
FEATURE CALCULATION
J - FUNCTION
Z2
PDF EVALUATION
1
COMPARE
24
FEATURE CALCULATION
J - FUNCTION
FEATURE CALCULATION
J - FUNCTION
FEATURE CALCULATION
J - FUNCTION
Z3
PDF EVALUATION
1
11
X

FIG. 3

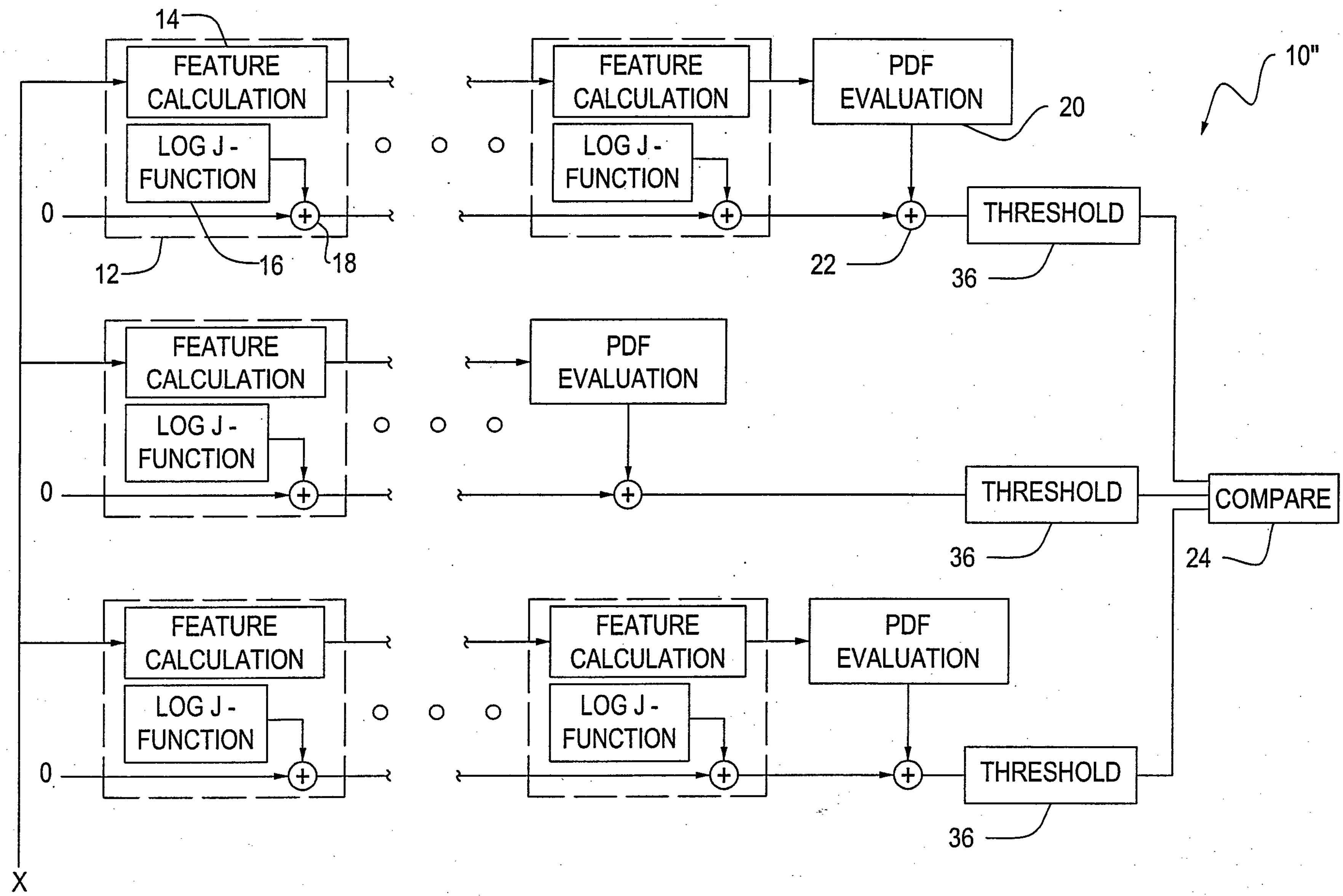

10"
14
FEATURE CALCULATION
LOG J - FUNCTION
12
16
18
FEATURE CALCULATION
LOG J - FUNCTION
PDF EVALUATION
20
22
THRESHOLD
36
FEATURE CALCULATION
LOG J - FUNCTION
PDF EVALUATION
THRESHOLD
36
COMPARE
24
FEATURE CALCULATION
LOG J - FUNCTION
FEATURE CALCULATION
LOG J - FUNCTION
PDF EVALUATION
THRESHOLD
36
0
0
0
X

FIG. 4