Pattern Recognition: Statistical, Structural and Neural Approaches

Robert J. Schalfoff
Clemson University

John Wiley + Sons

a significant problem. For example, when $d = 10$, there are $2^{10} = 1024$ probabilities to estimate. A simplification is to assume any two components x_i and xj in $\underline{x}$ are independent. Then

$$P(\underline{x}|w_i) = \prod_{i=1}^{d} P(x_i|w_i) = \prod_{i=1}^{d} p_i^{x_i}(1-p_i)^{1-x_i} \qquad (4-43)$$

from before, where $p_i = P(x_i = 1|w_i)$ and $1 - p_i = P(x_i = 0|w_i)$. We therefore need, for each class, only to estimate p_i, $i = 1, 2, \ldots d$. With this assumption, there exists a trade-off. In the general formulation there are 2^d probabilities with statistically dependent x_i permitted, whereas the simplified formulation requires estimation of d probabilities with the constraint that x_i must be statistically independent.

TECHNIQUES TO DIRECTLY OBTAIN LINEAR CLASSIFIERS

The Concept of Linear Separability

Viewing the samples of a $c = 2$ class training set as points in R^d, we note that some configurations of feature vectors are separable by a (possibly non-unique) hyperplane. Although this is not true for an arbitrary configuration of samples (this is considered in the exercises), the computational and conceptual advantages of a linear decision boundary often motivate us to consider its application, even at the expense of increased classification error rates vis-à-vis using the exact (non-planar) decision surfaces.

DEFINITION: **Linear Separability**

If a hyperplanar decision boundary exists that correctly classifies all the training samples for a $c = 2$ class problem, the samples are said to be linearly separable.

Recall (Appendix 6) that this hyperplane, denoted H_{ij}, is defined by parameters $\underline{w}$ and w_0 in a linear constraint of the form:

$$g(\underline{x}) = \underline{w}^T x - w_0 = 0 \qquad (4-44)$$

$g(\underline{x})$ separates R^d into positive and negative regions R_p and R_n, where

$$g(\underline{x}) = \underline{w}^T \underline{x} - w_0 = \begin{cases} > 0 & \text{if } \underline{x} \in R_p \\ 0 & \text{if } \underline{x} \in H_{ij} \\ < 0 & \text{if } \underline{x} \in R_n \end{cases} \qquad (4-45)$$

Design of Linear Classifiers

Assume a $c = 2$ class training set $H = \{\underline{x}_i\} \quad i = 1, 2, \ldots n$, which may be partitioned into H_1 and H_2, where H_i consists only of samples labeled w_i. The goal is to determine plane H_{12} where, for each $\underline{x}_i$ in H,

$$\underline{w}_{12}^T \underline{x}_i - w_0 = \begin{cases} > 0 & \text{if } \underline{x}_i \in H_1 \\ < 0 & \text{if } \underline{x}_i \in H_2 \end{cases} \tag{4-46}$$

This plane is characterized by $d + 1$ parameters, namely the d elements of $\underline{w}_{12}$ (the normal) and w_0. Defining

$$\underline{w} = \begin{pmatrix} \underline{w}_{12} \\ w_0 \end{pmatrix} \tag{4-47a}$$

and converting each $\underline{x}_i$ in H to a $(d+1) \times 1$ vector by adding '1' as the $(d+1)$st element yields the standard homogéneous coordinate representation (see, for example, [Riesenfeld 1981], [Schalkoff 1989]) as follows:

$$\hat{\underline{x}}_i = \begin{pmatrix} \underline{x}_i \\ 1 \end{pmatrix} \tag{4-47b}$$

Noting that $\underline{w}^T \hat{\underline{x}}_i$ is a scalar quantity allows (4-46) to be rewritten as

$$\hat{\underline{x}}_i^T \underline{w} = \begin{cases} > 0 & \text{if } \hat{\underline{x}}_i \in H_1 \\ < 0 & \text{if } \hat{\underline{x}}_i \in H_2 \end{cases} \tag{4-48}$$

A desirable modification to (4-48) is to replace each homogeneous vector $\hat{\underline{x}}_i$ in H_2 by its negative. This conversion therefore yields the single constraint:

$$\hat{\underline{x}}_i^T \underline{w} > 0 \qquad i = 1, 2, \ldots n \tag{4-49}$$

Considering all the 'converted' elements of H yields the matrix formulation of (4-49) as

$$A\underline{w} > \underline{0} \tag{4-50}$$

where the $n \times (d+1)$ matrix A consists of the converted vectors from the training set as:

$$A = \begin{pmatrix} \hat{\underline{x}}_1^T \\ \cdot \\ \hat{\underline{x}}_2^T \\ \vdots \\ \hat{\underline{x}}_n^T \end{pmatrix} \tag{4-51}$$

A 'Batch' (Pseudoinverse) Solution. Equation 4-50 is a set of n *linear inequalities*. Many solution procedures exist, including linear programming. A solution is developed that is based on converting (4-50) into a linear constraint, by defining a vector of user-chosen 'offsets', $\underline{b}$, as

$$\underline{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \qquad b_i > 0 \tag{4-52}$$

Thus, (4-50) becomes

$$A\underline{w} = \underline{b} \tag{4-53}$$

and a solution for the parameters of the separating plane, $\underline{w}$, is obtained via forming the pseudoinverse of A

$$\underline{\hat{w}} = A^{\dagger}\underline{b} \tag{4-54}$$

The Solution Region in R^{d+1}. Another approach for solving the system of linear inequalities given by (4-48) or, equivalently, (4-49) is to view these equations as n constraints in $(d+1)$-dimensional space. Each equation of the form of (4-49) together with a user-chosen offset or 'margin' may be written as

$$\underline{\hat{x}}_i^T\underline{w} - b_i > 0 \qquad i = 1, 2, \ldots n \tag{4-55}$$

From Appendix 6, each of the n linear inequality constraints in (4-55) may be visualized, *by viewing $\underline{\hat{x}}_i^T$ as the normal vector to a $(d+1)$-dimensional hyperplane that partitions* R^{d+1}. A requirement for a solution is that the solution vector $\underline{w}$ must lie in the positive half R_p of R^{d+1}, at a distance of $|b_i|/\parallel \underline{\hat{x}}_i^T \parallel$ from the boundary. Moreover, the intersection of the n half spaces of R^{d+1} defined by (4-55) is the overall *solution region* for $\underline{w}$. In problems that are not linearly separable, this region does not exist. Conversely, in linearly separable solutions with non-unique separating planes, this region contains an infinite number of solution points. In addition, by setting the margins $b_i = 0, \quad i = 1, 2, \ldots n$, we find the largest solution region from solving

$$\underline{\hat{x}}_i^T\underline{w} > 0 \quad i = 1, 2, \ldots n \tag{4-56}$$

Iterative (Descent) Procedures. By using a gradient approach (Appendices 2 and 4), an iterative procedure to determine $\underline{w}$ may be found. The form is

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} - \alpha_n \frac{\partial J(\underline{w})}{\partial \underline{w}}\Big|_{\underline{w}=\underline{w}^{(n)}} \tag{4-57}$$

where α_n controls the adjustment at each iteration. The iteration procedure requires a stopping criterion. Examples are

$$\parallel \underline{w}^{(n+1)} - \underline{w}^{(n)} \parallel < \epsilon \tag{4-58a}$$

where ϵ is a user-chosen tolerance, or

$$n = n_{max} \tag{4-58b}$$

where n_{max} is the (predetermined) maximum number of iterations, or

$$J(\underline{w}^{(n)}) \leq J_T \qquad (4-58c)$$

where J_T is an error threshold and $J(\underline{w})$ is a measure of classification error. Often we design $J(\underline{w})$ with the minimum value $J(\underline{w}) = 0$ for perfect classification.

Error Forms. Many forms for the error $J(\underline{w})$ are possible. For example, a vector $\underline{w}$ where

$$\hat{\underline{x}}_i^T \underline{w} < 0 \qquad (4-59)$$

misclassifies sample $\hat{\underline{x}}_i^T$. Therefore, one error measure, the *Perceptron Criterion Function*, is

$$J_p(\underline{w}) = -\sum_{\hat{\underline{x}} \in X_{ERR}(\underline{w})} (\hat{\underline{x}}_i^T \underline{w}) \qquad (4-60)$$

where $X_{ERR}(\underline{w})$ is the set of *samples misclassified by* $\underline{w}$. Note that this set will vary from iteration to iteration in the solution procedure. If $X_{ERR}(\underline{w}) = \emptyset$, then $J_p(\underline{w}) = 0$, and the minimum of the error function is obtained.

Since

$$\nabla_{\underline{w}} J_p(\underline{w}) = -\sum_{\hat{\underline{x}}_i \in X_{ERR}(\underline{w})} \hat{\underline{x}}_i \qquad (4-61)$$

the iterative procedure of (4-57) becomes

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} + \alpha_n \sum_{\hat{\underline{x}}_i \in X_{ERR}(\underline{w}^{(n)})} \hat{\underline{x}}_i \qquad (4-62)$$

Notice that when $X_{ERR}(\underline{w}^{(n)}) = \emptyset$, the adjustments to $\underline{w}^{(n)}$ cease.

Training by Sample and Training by Epoch. Equation 4-62 suggests that at each iteration the entire set of samples misclassified by $\underline{w}^{(n)}$ be used to form the correction at the next iteration. This represents a consideration of the entire training set for each adjustment of $\underline{w}$ and, thus, *training by epoch*. Another alternative is to *adjust* $\underline{w}$ *as soon as a single classification error is made*. This represents *training by sample*, and may be viewed as a 'correct as soon as possible' strategy. It is often unclear whether training by epoch or training by sample is preferable, and this concern carries over into our training of certain similar neural network structures in Chapter 12. In the case of training by sample, (4-62) becomes

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} + \alpha_n \hat{\underline{x}}_i \qquad (4-63)$$

where $\hat{\underline{x}}_i$ is the first sample misclassified by $\underline{w}^{(n)}$.

Procedures to Find Both $\underline{w}$ ***and*** $\underline{b}$. In the previous procedures, it was necessary to choose the 'margin' vector $\underline{b}$. We consider a procedure [Ho/Kashyap 1965] based on iterative refinement of both $\underline{w}$ and $\underline{b}$ that is derived from the approach of (4-53). Choosing an error measure as

$$J_H(\underline{w}, \underline{b}) = \| A\underline{w} - \underline{b} \|^2 = (A\underline{w} - \underline{b})^T (A\underline{w} - \underline{b}) \qquad (4-64)$$

Vector-Matrix Differentiation Formulae

Examples of properties using the above definitions may be easily derived and are summarized in the discussion that follows. For a matrix A and vectors $\underline{x}$ and $\underline{y}$

$$\frac{d}{d\underline{x}}(A\underline{x}) = A$$

$$\frac{d}{d\underline{x}}(\underline{y}^T A\underline{x}) = A^T\underline{y}$$

$$\frac{d}{d\underline{x}}(\underline{x}^T A\underline{x}) = (A + A^T)\underline{x} \qquad (A.1-3)$$

LEAST SQUARES TECHNIQUES (DETERMINISTIC)

The Formulation of a Pseudoinverse of a Matrix

The problem of forming an inverse of a rectangular matrix A with specified properties has been studied for some time [Rao 1971]. The pseudoinverse of an $m \times n$ real matrix A is an $n \times m$ matrix denoted by $A^\dagger$. Examples of desirable properties are

$$AA^\dagger A = A$$

$$A^\dagger AA^\dagger = A^\dagger$$

$$(AA^\dagger)^T = AA^\dagger \qquad (A.1-6)$$

If A has full column rank, one inverse of considerable interest is the so-called least squares inverse, denoted by

$$A^\dagger = (A^T A)^{-1}A^T \qquad (A.1-7)$$

The properties of this solution are considered extensively in references on least squares estimation, interpolation, and the like. One important note is that the formulation of this pseudoinverse requires the inversion of an $M \times M$ nonsingular matrix (or conversely, the solution of the so-called normal equations, which are also of order M). Numerous successful algorithms for the solution of this problem are available.

Basic Formulation

Suppose we are given an overdetermined linear equation of the form:

$$\underline{b} = A\underline{x} \qquad (A.1-8)$$

where $\underline{b}$ is $m \times 1$, $\underline{x}$ is $n \times 1$, $m > n$ and A is $m \times n$ with rank n. There is no way to exactly satisfy this equation for arbitrary $\underline{b}$. We define an $m \times 1$ error function vector corresponding to some approximate solution, $\hat{\underline{x}}$:

$$\underline{e} = \underline{b} - A\hat{\underline{x}} \qquad (A.1-9)$$

and then determine a procedure to minimize some function of this error. Often, in unweighted least squares, this function, denoted J, is chosen to be

$$J = \underline{e}^T\underline{e} \qquad (A.1-10)$$

To find the minimum of this function, we set

$$\frac{dJ}{d\underline{x}} = \underline{0} \qquad (A.1-11)$$

and use (A.1-3) to develop the so-called normal equations, that is,

$$A^T A\hat{\underline{x}} = A^T\underline{b} \qquad (A.1-12)$$

from which $\underline{b}$ may be determined. Note that in theory $A^T A$ may be inverted to yield $\hat{\underline{x}}$.