Numerical Method for LTG Synthesis

Prof. Mohamad Hassoun©

September 1999

Suppose we have a data set of the form $\{(\mathbf{x}^1, d^1), (\mathbf{x}^2, d^2), \ldots, (\mathbf{x}^m, d^m)\}$, where $\mathbf{x}^i \in R^{n+1}$ and $d^i = 1$ if $\mathbf{x}^i$ is in *class I* and $d^i = 0$ if $\mathbf{x}^i$ is in *class II*. Assuming the data set is linearly separable, a single LTG can be synthesized to correctly classify the data points $\mathbf{x}^i$ if an $(n + 1)$-dimensional weight vector $\mathbf{w}$ (here, and in order to unify the representation of the threshold parameter T with the LTG weight parameters, we set $x^i_{n+1} = 1$, for all i, and we represent $-T$ as the parameter w_{n+1} of the weight vector) is computed which satisfies the following set of m inequalities

$$(\mathbf{x}^i)^{\mathrm{T}}\mathbf{w} \begin{cases} > 0 & \text{if } d^i = 1 \\ < 0 & \text{if } d^i = 0 \end{cases} \qquad \text{for } i = 1, 2, \ldots, m$$

Next, if one defines a set of m new vectors $\mathbf{z}^i$ according to

$$\mathbf{z}^i = \begin{cases} +\mathbf{x}^i & \text{if } d^i = 1 \\ -\mathbf{x}^i & \text{if } d^i = 0 \end{cases} \qquad \text{for } i = 1, 2, \ldots, m$$

and we let

$$\mathbf{Z} = [\mathbf{z}^1 \ \ \mathbf{z}^2 \ \ \ldots \ \ \mathbf{z}^m]$$

then the above set of inequalities may be rewritten as the single matrix equation

$$\mathbf{Z}^{\mathrm{T}}\mathbf{w} > \mathbf{0}$$

Now, defining an m-dimensional positive margin vector $\mathbf{b}$ ($\mathbf{b} > \mathbf{0}$) and using it in the above equation, one arrives at the following equation

$$\mathbf{Z}^{\mathrm{T}}\mathbf{w} = \mathbf{b}$$

Thus, if one defines $\mathbf{A} = \mathbf{Z}^{\mathrm{T}}$ the LTG synthesis may now be accomplished by solving $\mathbf{Aw} = \mathbf{b}$ for $\mathbf{w}$, subject to the constraint $\mathbf{b} > \mathbf{0}$. Here, three cases arise: (*i*) $m = n+1$, (*ii*) $m < n+1$, and (*iii*) $m > n+1$. First, for case(*i*), and assuming linearly independent vectors $\mathbf{z}$ (or $\mathbf{x}$), the inverse of $\mathbf{A}$ exists and the solution is given by

$$\mathbf{w} = \mathbf{A}^{-1}\mathbf{b}$$

Second, for case (*ii*), known as the underdetermined case, there are more unknowns (weights) than equations. Here, an infinite number of solutions exist if $\{\mathbf{x}^i\}$ is linearly independent. In particular, the *minimum norm solution* (Rao and Mitra, 1971), that minimizes the norm $\|\mathbf{w}\|$, may be used and is given by

$$\mathbf{w} = \mathbf{A}^{\mathrm{T}}\left(\mathbf{A}\mathbf{A}^{\mathrm{T}}\right)^{-1}\mathbf{b}$$

Third, the overdetermined case, case (*iii*), has more equations than unknowns and therefore no solution for $\mathbf{w}$ exists which satisfies the equation $\mathbf{Z}^{\mathrm{T}}\mathbf{w} = \mathbf{b}$. Here, one might seek the best solution in the sense of minimizing a performance (objective) function. In practice, the solution

$$\mathbf{w} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b} = \left(\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\right)^{-1}\mathbf{Z}\mathbf{b}$$

is used that minimizes the *sum of the squared error* (*SSE*) criterion (objective) function $J(\mathbf{w})$ given by

$$J(\mathbf{w}) = \|\mathbf{A}\mathbf{w} - \mathbf{b}\|^2 = \sum_{i=1}^{m} \left(y(\mathbf{x}^i) - b_i \right)^2$$

It should be noted that the minimum SSE solution does not necessarily lead to a zero *classification error* solution; i.e., the SSE solution may not lead to an LTG solution (hyperplane separating surface) that correctly separates all data points $\mathbf{x}^i$, when the vectors are linearly separable (more on this in Section 3.1.2). In Chapter 3, alternative learning (synthesis) rules are derived which are guaranteed to lead to zero classification error solutions.

Example 1.1.1 Find appropriate weights for an LTG to realize the threshold function defined by Equation (1.1.2).

According to the above formulation and with the help of the K-Map in Figure 1.1.6, we may write the system of equations $\mathbf{A}\mathbf{w} = \mathbf{b}$ for this LTG as

$$\begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & -1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & 0 & -1 \\ -1 & 0 & -1 & -1 \\ 1 & 1 & 0 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

where we have set $b_i = 1$ for $i = 1, 2, \ldots, m$, and the threshold is given by $T = -w_4$. The minimum SSE solution is given by $\mathbf{w} = [\ -1/2\ 3/2\ -1/2\ -1/2]^T$ which represents the separating hyperplane

$$g(x_1, x_2, x_3) = -\frac{x_1}{2} + \frac{3x_2}{2} - \frac{x_3}{2} - \frac{1}{2} = 0 \quad \text{or} \quad -x_1 + 3x_2 - x_3 = 1$$

This solution also leads to the correct classification of all points $\mathbf{x}$ by noting that the vector

$$\mathbf{A}\mathbf{w} = [.5\ \ 1\ \ 1\ \ .5\ \ 1\ \ 1.5\ \ .5\ \ 0]^T$$

has all positive components. It is interesting to note that the most sensitive pattern is $\mathbf{x}^8 = [1, 1, 1]^T$ since it lies on the separating surface. One can show that by properly choosing the *margin* component b_8, a preferable solution for the separating hyperplane may be generated (e.g., try $\mathbf{b} = [1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 2]^T$). Section 3.1.4 presents an automated and adaptive method for computing margins that produce weight vectors which give zero classification error.

It should be noted here that the above LTG synthesis method can be readily applied to the more general case $\mathbf{x} \in R^n$. One may also use this method to compute the parameters of the more general class of threshold gates which is considered next.

Extensions to PTG Synthesis

Since the ϕ-mapping (preprocessing layer) has no weights to adjust, the synthesis of a PTG(r) is reduced to the synthesis of an LTG. Here, a d-dimensional weight vector $\mathbf{w} = [w_1\ w_2\ \ldots\ w_{d-1}\ -T]^T$ is desired which leads to the correct classification of a set of m labeled vectors $\{\mathbf{x}^i, d^i\}$, $d^i \in \{0,1\}$. Upon the presentation of the input $\mathbf{x}^i$, the LTG receives the input vector $\boldsymbol{\phi}(\mathbf{x}^i) = [\phi_1(\mathbf{x}^i)\ \phi_2(\mathbf{x}^i)\ \ \ldots\ \phi_{d-1}(\mathbf{x}^i)\ 1]^T$ and generates the output $y = \text{step}[\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}^i)]$, where the *step* function returns a 1 if its argument is positive, and returns a 0 otherwise. Now, the synthesis method described at the end of Section 1.1.1 may be used to compute the weight values, according to the following equations:

$$\mathbf{w} = \begin{cases} \mathbf{A}^{\mathrm{T}}\left(\mathbf{A}\mathbf{A}^{\mathrm{T}}\right)^{-1}\mathbf{b} & \text{for} \quad m < d \\ \mathbf{A}^{-1}\mathbf{b} & \text{for} \quad m = d \\ \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b} & \text{for} \quad m > d \end{cases} \tag{1.3.13}$$

where **b** is an m-dimensional positive vector, and $\mathbf{A} = \mathbf{\Phi}^{\mathrm{T}} = [(\alpha^1\phi(\mathbf{x}^1) \quad \alpha^2\phi(\mathbf{x}^2) \quad \ldots \quad \alpha^m\phi(\mathbf{x}^m))]^{\mathrm{T}}$, with $\alpha^i = 1$ if $d^i = 1$ and $\alpha^i = -1$ if $d^i = 0$.