Notes on Statistical and Machine Learning

Self-Organizing Maps

Chapter: *27* **Prepared by:** *Chenxi Zhou*

This note is prepared based on

- *Chapter 14, Unsupervised Learning* in Hastie, Tibshirani, and Friedman (2009), and
- *Chapter 12, Clustering Analysis* in Izenman (2009).

I. Introduction

1. **Overview:** *Self-organizing map* (SOM) can be viewed as a constrained version of K-means clustering in which the prototypes are encouraged to lie in a one- or two-dimensional manifold in the feature.

2. **Primary Use:** The primary use of an SOM is

   (a) in reducing high-dimensional data to a lower-dimensional nonlinear manifold, and

   (b) in displaying graphically the results of such data reduction.

3. **Goal:** The goal of SOM is to map the projected data to discrete interconnected nodes, where each node represents a grouping or cluster of relatively homogeneous points.

4. **End Product:** The end product of the SOM algorithm is a image called an *SOM plot*. The SOM plot is displayed in output space and consists of a grid (or network) of a large number of interconnected nodes.

   Remark. In two dimensions, the nodes are typically arranged as a square, rectangular, or hexagonal grid.

5. **Example of an SOM Plot in Two-dimensional Case:** Consider to create an SOM plot in two-dimensional setting with rectangular grids. Let the set of rows be $\mathcal{K}_1 := \{1, 2, \cdots, K_1\}$ and the set of columns be $\mathcal{K}_2 := \{1, 2, \cdots, K_2\}$, where K_1 (the height) and K_2 (the width) are chosen by the user.

   Then, a node within the SOM plot is defined by its coordinates, $(k_1, k_2) \in \mathcal{K}_1 \times \mathcal{K}_2$. It will be convenient to map the collection of nodes into an ordered sequence, so that the node (k_1, k_2) is relabeled as $k := (k_1 - 1)K_2 + k_2 \in \mathcal{K}$, where $\mathcal{K} := \{1, 2, \cdots, K_1K_2\}$.

   Remark. The total number of nodes, $K := K_1K_2$, is usually chosen by trial and error, initially much larger than the suspected number of clusters in the data.

   After an initial SOM analysis, one can reconfigure the SOM by reducing the number of row and column nodes so as to reduce the value of K.

II. On-line SOM Algorithm

1. **General Idea of SOM Algorithm:** We associate with the k-th node in an SOM plot a representative in input space, $\mathbf{m}_k \in \mathbb{R}^p$, where $k \in \mathcal{K}$ and p is the dimensionality of the feature variables.

2. **Initialization of $\{\mathbf{m}_k\}_{k=1}^K$:** It is usual to initialize the components of $\mathbf{m}_k$ to be random numbers, for all $k \in \mathcal{K}$.

3. **c-Grid Neighbor and c-Neighborhood Set:** Let $c > 0$ be fixed. A node $k' \in \mathcal{K}$ is defined to be a *c-grid neighbor* of the node $k \in \mathcal{K}$ if the distance between $\mathbf{m}_k$ and $\mathbf{m}_{k'}$ is smaller than a given threshold c. The set of nodes that are c-grid neighbors of the node k, denoted by $\mathcal{N}_c(k)$, is called the *c-neighborhood set* for that node.

4. **On-line Version of SOM Algorithm:** The on-line version of SOM algorithm processes each observation $\mathbf{x}_i \in \mathbb{R}^p$ individually and sequentially. The complete algorithm is shown in Algorithm 1.

Algorithm 1 On-line Version of SOM Algorithm

Require: map size, that is, K_1 and K_2;
Require: initializing points of all representatives $\{\mathbf{m}_k\}_{k=1}^K$;
Require: data $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$.

1: **for** a sequence of data points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_n$, **do**
2: *Standardization of Data Point:* Standardize $\mathbf{x}_i \in \mathbb{R}^p$ so that each component of $\mathbf{x}_i$ has zero mean and variance one. We still use $\mathbf{x}_i$ to denote the data point after standardization.
Remark. By standardization, no component variable has undue influence on the results just because it has a large variance or absolute value.
3: *Find the Best-matching Unit:* Compute the distance between $\mathbf{x}_i$ and each representative $\mathbf{m}_k$, and find the node whose representative yields the smallest distance to $\mathbf{x}_i$. In other words, we solve the following optimization problem

$$\underset{k \in \mathcal{K}}{\text{minimize}} \, \|\mathbf{x}_i - \mathbf{m}_k\|_2^2,$$

and let the minimizer be k_i^*. The representative $\mathbf{m}_{k_i^*}$ is declared as the "winner", and k_i^* is referred to as the *best-matching unit* (BMU) or *winning node* for $\mathbf{x}_i$.
4: *Update the Representatives of Neighbors:* We update the representatives corresponding to k_i^* and each of its c-grid neighbors so that each $\mathbf{m}_k$, $k \in \mathcal{N}_c(k_i^*)$, is closer to $\mathbf{x}_i$ through

$$\mathbf{m}_k \quad \longleftarrow \quad \mathbf{m}_k + \alpha(\mathbf{x}_i - \mathbf{m}_k), \qquad \text{for all } k \in \mathcal{N}_c(k_i^*), \tag{1}$$

where $\alpha \in (0, 1)$ is the learning rate.
5: Repeat the preceding process for a large number of times.
6: **end for**

Remark. The effect of the update (1) is to move the prototypes closer to the data, but also to maintain a smooth two-dimensional spatial relationship between the prototypes.

5. **Updating Representatives Using Distance-weighted Function:** A "distance-weighted" version of (1) is

$$\mathbf{m}_k \quad \longleftarrow \quad \mathbf{m}_k + \alpha h_k(\|\mathbf{m}_k - \mathbf{m}_{k_i^*}\|_2)(\mathbf{x}_i - \mathbf{m}_k), \qquad \text{for all } k \in \mathcal{N}_c(k_i^*),$$

where the neighborhood function h_k depends upon how close the neighboring representatives are to $\mathbf{m}_{k_i^*}$. The closer $\mathbf{m}_k$ is to $\mathbf{m}_{k_i^*}$, the more weights are given to $\mathbf{m}_k$.

Example. A popular choice of h_k is the Gaussian kernel function given by

$$h_k(\|\mathbf{m}_k - \mathbf{m}_{k_i^*}\|) = \exp\left(\frac{\|\mathbf{m}_k - \mathbf{m}_{k_i^*}\|_2^2}{2\sigma^2}\right)$$

where $\sigma > 0$ is the neighborhood radius.

6. **Effects of Threshold c:** If we take the threshold value c to be so small that each neighborhood contains only a single point, we lose the dependencies between representatives. The SOM algorithm reduces to an on-line version of K-means clustering, where K is the total number of nodes.

7. **Choices of c, α and σ:** During the course of running the SOM algorithm, all three parameters, c, α and σ, should be monotonically decreasing.

References

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. Vol. 1. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Izenman, Alan J (Mar. 2009). *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. en. Springer Science & Business Media.