CSE 475: Statistical Methods in AI **Monsoon 2019**

SMAI-M-2019 13: More on Linear Methods

Lecturer: C. V. Jawahar *Date: DATE*

13.81 Multiclass Classification

We had seen binary classification schemes. You will see more of them. We know what we need is a multi class classification for many problems. How do we extend these binary classifiers to a multiclass setting in general? We can *fuse* the results of many binary classifiers to obtain a multi class classifier.

Let us consider we have K classes.

1. We can build K "one Vs rest" classifiers.
2. We can build ${}_KC_2$ pairwise ($ivsj$) classifiers.
3. We can build many (find the combinatorics!) classifier that seperates samples from a set or classes from that of another set of classes.

How do we arrive at a final classification decision in all these cases?

13.81.1 Hierarchical

We achive multiclass classification by hierarchically arranging a number of classifiers of type 3. The advantage is that we can achieve the final class in log() evaluations. However, the challenge is in designing. How many classes need to be trained? How they should be arranged? Popularly this type of classifiers are called Binary Hierarchical Classifier (BHC).

- Q: Suggest a useful heuristics (a greedy algorithm) to design a heirarchical classifier?

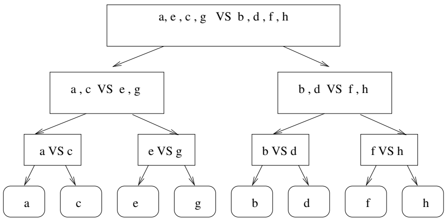

Figure 13.9: Example BHC. How many such BHCs are possible for an 8 class problem?

13.81.2 One vs Rest

If the K one vs rest classifiers can provide probabilities, life is simple as we had seen in the case of softmax. Pick the class with highest probability.

If probabilities are not available (and only class label is predicted), then we may have two cases to worry. All classifiers says "rest". or multiple classifiers assign a class-label. In either case, the final decision is difficult without some additional information.

13.81.3 Pairwise Classifiers and Fusion

When there are ${}_KC_2$ classes, one can use simple majority voting to find the best classifier. One can also use DAG (directed acyclic graph) like structure.

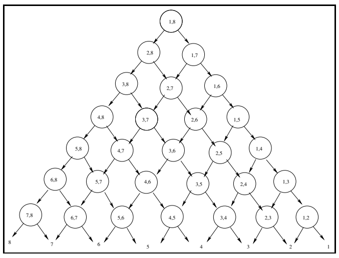

Figure 13.10: A DAG based solution for Eigtht classes

13.82 More on Linear Dimensionality Reduction

We had seen the notion of linear dimensionality reduction

$$\mathbf{z} = \mathbf{W}\mathbf{x}$$

Dimensionality reduction is a vast, important area in the classical machine learning with many implications. Broadly methods are split into:

1. Linear or Non-Linear Methods
2. Local or Global Methods
3. Supervised or Unsupervised Methods

We will not see examples of all these in this course. PCA is an example of (i) Unsupervised (ii) Linear (iii) Global Method.

We will see later, LDA, which is a Supervised dimensionality reduction.

13.82.1 PCA: Summary

- Objective of PCA is to project the data to direction which best *preserves its covariance structure*.
- Let us assume that the data is centered $\sum_i \mathbf{x_i} = 0$. Also $\mathbf{x_i} \in R^N$, $i = 1, \ldots, M$. Centering is done by subtracting the mean from all the samples.
- Covariance matrix $C = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x_i}\mathbf{x_i^T}$ is first calculated. C is a $N \times N$ matrix. We avoid the use of Σ for the covariance matrix here. Let us not get it confused with the $\sum$ we use for Summation later extensively here.
- Assume we have a set of M centered observations: $\mathbf{x_k}, k = 1 \ldots M, \mathbf{x}_k \in \mathbf{R}^N, \sum_{k=1}^{M} \mathbf{x}_k = 0$

  For linear PCA, we solve the equation

$$C\mathbf{v} = \lambda\mathbf{v}$$

- This leads to eigen vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots$. There are a total of $\min(M, N)$ nonzero eigen values.
- And data is projected to P eigen vectors corresponding to top $P < N$ eigen values of the covariance matrix C.

We see two obvious scopes for improvement:

- PCA does not focus on discriminative nature. It aims at compression/compaction. There is no explicit objective that helps separation.
- PCA is linear. A nonlinear dimensionality reduction could have been more useful.

13.83 LDA

Let us now consider a "discriminative dimensionality reduction". Popularly this is called Linear Discriminant Analysis or Fisher Discriminant Analysis. This is supervised in nature (i.e., we use the class labels y_i also.).

Let us assume that we have two classes A and B. We are interested in finding a new feature $z = \mathbf{u}^T\mathbf{x}$ by projecting on to a new vector $\mathbf{u}$. What are our requirements so that this is good for the classification?

- After the projection, classes should be compact (i.e., samples from A should all come closer and samples from B should all come closer.). All the samples come closer to their own means.
- After the projection classes A and B should be well separated (i.e., they are easily separable.) Means of the classes become farther.

Let us introduce two new notations i.e., within scatter and between scatter of the classes. Both are captured as:

$$S_B = [\mu_1 - \mu_2][\mu_1 - \mu_2]^T$$

$$S_W = S_1 + S_2$$

$$= \sum_{\mathbf{x_i} \in \omega_1} [\mathbf{x_i} - \mu_1][\mathbf{x_i} - \mu_1]^T + \sum_{\mathbf{x_i} \in \omega_2} [\mathbf{x_i} - \mu_2][\mathbf{x_i} - \mu_2]^T$$

Assume we project onto a new (unknown) direction $\mathbf{u}$. After the projection S_B becomes $\mathbf{u}^T\mathbf{S_B}\mathbf{u}$ and S_w becomes $\mathbf{u}^T\mathbf{S_W}\mathbf{u}$.

- Here B indicates "between class"
- and W indicates "within class"

We convert this into a new objective function

$$J(\mathbf{u}) = \frac{\mathbf{u}^T\mathbf{S_B}\mathbf{u}}{\mathbf{u}^T\mathbf{S_W}\mathbf{u}}$$

We would like to maximize this quantity. Since the optima is invariant scaling, let us consider the problem as

$$\max_{\mathbf{u}} \frac{1}{2}\mathbf{u}^T\mathbf{S_B}\mathbf{u}$$

such that $\mathbf{u}^T\mathbf{S_W}\mathbf{u} = 1$. The maximization problem now (cf: Lagrangian)

$$\max_{\mathbf{u}} \frac{1}{2}\mathbf{u}^T\mathbf{S_B}\mathbf{u} - \frac{\lambda}{2}(\mathbf{u}^T\mathbf{S_W}\mathbf{u} - 1)$$

Differentiating with respect to $\mathbf{u}$ and equating to zero.

$$\mathbf{S_B}\mathbf{u} = \lambda\mathbf{S_W}\mathbf{u} \tag{13.25}$$

Such problems are called generalized eigen value problems. $\mathbf{S_B}\mathbf{u}$ is a vector along $[\mu_1 - \mu_2]$. Therefore if $\mathbf{S_W}$ can be inverted,

$$\mathbf{u} = \alpha\mathbf{S_W}^{-1}[\mu_1 - \mu_2]$$

Many questions are left to you to figure out.

- How do we extend this to multi class (beyond two classes)? How does the definition of S_w and S_B change?
- How do we get multiple direction/features/$\mathbf{u}$ than one? What is the second best direction, given that the first one is identified?
- What is the generalized eigen value problem mentioned here? Isn't it LDA also an eigen vector solution?

13.83.1 Tricks

There are a number of tricks that we could do to make our life easy.

1. For example, $\mathbf{S_W}$ could be redefined as:

$$\mathbf{S_W} = \mathbf{S_W} + \rho\mathbf{I} \tag{13.26}$$

   where ρ is a small real quantity. How does this way of regularizing $\mathbf{S_W}$ help?

2. Alternatively, one could define $\mathbf{S_W}$ as (see "Multiple-Exemplar Discriminant Analysis for Face Recognition" for details). Here the scatter is computed not with respect to mean but with respect to each samples and added.

$$\mathbf{S_W} = \sum_{i=1}^{C} \frac{1}{N_i^2} \sum_{j=1}^{N_i}\sum_{k=1}^{N_i} [\mathbf{x_j^i} - \mathbf{x_k}^i][\mathbf{x_j^i} - \mathbf{x_k}^i]^T \tag{13.27}$$

   The new definitions of $\mathbf{S_W}$ is some what different from that of the original one. However, both these help in making the $\mathbf{S_W}$ full rank.

3. Another trick is to first apply PCA, and then LDA. This also helps in making $\mathbf{S_W}$ full rank.

Q: *Explain how all the three above methods help in practice?*

13.84 Multi Dimensional Scaling (MDS)

Another popular methods for dimensionality reduction is Multi Dimensional Scaling (MDS).

Like in the previous methods, we can define the dimensionality reduction as a linear transformation:

$$\mathbf{z}_i = \mathbf{W}\mathbf{x}_i$$

Let $\mathbf{D}$ is a $N \times N$ matrix with each element $D(i,j)$ be the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$.

$$D(i,j) = dist(\mathbf{x}_i, \mathbf{x}_j)$$

Let us also look at $\mathbf{D}'beanother\mathrm{N}\times N$ matrix of distances in the lower dimension.

$$D'(i,j) = dist(\mathbf{z}_i, \mathbf{z}_j)$$

The MDS aims at finding a dimensionality reduction (read $\mathbf{W}$) such that both these distances are as close as possible.

$$\arg\min_{\mathbf{W}} ||\mathbf{D} - \mathbf{D}'||$$