Paper one

Multi-View Intact Space Learning

Motivation

It is known from practice that the information obtained from a single view is insufficient thus the task of multi-view learning is vital. However, existing multi-view learning algorithms have their shortcomings so the authors want to present a more powerful method, which is capable to address the problem of single-view insufficiency and avoid losing information during training. Moreover, the paper tries to give us a theoretical support of multi-view learning.

Related definitions and mathematical theories

In multi-view learning, $z^v = f^v(x) + \varepsilon^v$, where z^v is multi-view feature. For a linear function $f^v(x) = W_v x$, what we want is to learn $\{W_v\}_{v=1}^m$ by minimizing $\{z^v - f^v(x)\}_{v=1}^m$.

Cauchy estimator:

Since using L1 and L2 loss to minimize $\{z^v - f^v(x)\}_{v=1}^m$ has been proved to be not robust to outliers, the authors decide to use Cauchy estimator: $\rho(x) = \log(1 + (x/c)^2)$, whose upper bounded infuence function is more robust. The reasons can be easily understood because least-quare estimators consider all the points equally some of which are even outliers. Moreover, L2 loss's derived function increases linearly with the size of its error. L1 loss's

derived function has no cut-off. Using cauthy estimator function can avoid above problems.

Objective:

Therefore, the objective function using caucy estimators with regularization terms are as followed:

$$\min_{x,W} \frac{1}{mn}\sum_{v=1}^{m}\sum_{i=1}^{n}\log\left(1+\frac{\|z_i^v - W_v x_i\|^2}{c^2}\right) + C_1\sum_{v=1}^{m}\|W_v\|_F^2 + C_2\sum_{i=1}^{n}\|x_i\|_2^2,$$

- For fixed view generation functions $\{W_v\}_{v=1}^m$

The objective function is: $\min_x \mathcal{J} = \frac{1}{m}\sum_{v=1}^{m}\log\left(1+\frac{\|z^v - W_v x\|^2}{c^2}\right) + C_2\|x\|_2^2.$. We set the gradient of objective function to 0, such that we have $\sum_{v=1}^{m} -\frac{2W_v^T(z^v - W_v x)}{c^2 + \|z^v - W_v x\|_2^2} + 2mC_2 x = 0,$.

Rewrite it as $x = \left(\sum_{v=1}^{m} W_v^T Q_v W_v + mC_2\right)^{-1}\sum_{v=1}^{m} W_v^T Q_v z^v.$, where $Q = \left[\frac{1}{c^2+\|r^1\|^2},\ldots,\frac{1}{c^2+\|r^m\|^2}\right]$.

- For fixed data points $\{x_i\}_{i=1}^n$

The objective function is: $\min_W \mathcal{J} = \frac{1}{n}\sum_{i=1}^{n}\log\left(1+\frac{\|z^i - Wx_i\|^2}{c^2}\right) + C_1\|W\|^2.$. Similarly, we set the gradient of J to 0 and rewrite it as $W = \sum_{i=1}^{n} z_i Q_i x_i^T\left(\sum_{i=1}^{n} x_i Q_i x_i^T + nC_1\right)^{-1}$, where $Q = \left[\frac{1}{c^2+\|r_1\|^2},\ldots,\frac{1}{c^2+\|r_n\|^2}\right]$. But we should notice that r^m in the former Q and r_n in this Q are different. r^m represents that $r^v = z^v - W_v x$ but r_n represents that $r_i = z^i - Wx_i$.

Algorithms

==========

- For fixed view generation functions $\{W_v\}_{v=1}^m$

Input: $\{z^1,...,z^m\}, \{W_v\}_{v=1}^m, x_0$

Output: $\{x_i\}_{i=1}^n$

1. Initialization: initial x_0 and choose the parameter C_2, compute $\{r^v\}_{v=1}^m$ by calculating $r^v = z^v - W_v x_0$
2. For k =1 ,… do

   Compute Q according to $Q = \left[\frac{1}{c^2 + \left\| r^1 \right\|^2}, ..., \frac{1}{c^2 + \left\| r^m \right\|^2} \right]$

   Compute x_k based on $x = \left(\sum_{v=1}^m W_v^T Q_v W_v + mC_2 \right)^{-1} \sum_{v=1}^m W_v^T Q_v z^v$

   Update $\{r^v\}_{v=1}^m$ by calculating $r^v = z^v - W_v x_k$

   If $x_k = x_{k-1}$,which means x has reach convergence, then

   break;
3. Then we have already obtained $\{x_i\}_{i=1}^k$

==========

- For fixed data points $\{x_i\}_{i=1}^n$

Input: $\{z^1,...,z^m\}, \{x_i\}_{i=1}^n, W_0$

Output: $\{W_v\}_{v=1}^m$

1. Initialization: initial W_0 and choose the parameter C_1, compute $\{r_i\}_{i=1}^n$ by calculating $r_i = z^i - W_0 x_i$
2. For k =1 ,… do

Compute Q according to $Q = \left[\frac{1}{c^2 + \|r_1\|^2}, ..., \frac{1}{c^2 + \|r_n\|^2} \right]$

Compute W_k based on $W_k = \sum_{i=1}^{n} z^i Q_i x_i^T \left(\sum_{i=1}^{n} x_i Q_i x_i^T + nC_1 \right)^{-1}$

Update $\{r_i\}_{i=1}^{n}$ by calculating $r_i = z^i - W_k x_i$

If $W_k = W_{k-1}$,which means W has reach convergence, then

break;

3. Then we have already obtained $\{W_v\}_{v=1}^{m}$

==========

Summary

This paper presents a novel robust method to learn the latent intact representations of multi-view with Iteratively Reweight Residuals technique. By employing Cauchy loss as the error measurement, the algorithm becomes more robust. Moreover, this paper gives us many mathematical theories to support its view point and experiments show that this method works well for practical applications.

My Idea

In order to better understand the mathematical theory of this method, I learn some materials about M-estimators and Kernel extension. To sum up, M-estimators is a robust method commonly used in statics and Kernel function is often used to increase dimensionality for better representing or better dividing or something else.

Paper two

AnyDBC: An Efficient Anytime Density-based Clustering Algorithm for Very Large Complex Datasets

Original DBSCAN

Input: Dataset D=$\{x_1, x_2, ..., x_m\}$, parameters(ε,MinPts), distance measure

Output: Clusters c=$\{C_1, ...\}$

1. Range query

   For i = 1,2,…,m, do

      For j=1,2,…,m, do

         If $\|x_i - x_j\| \le \varepsilon$, then x_j joins x_i's neighbor set $N_\varepsilon\{x_i\}$

   For i = 1,2,..,m, do

      If $|N_\varepsilon\{x_i\}| \ge MinPts$, then x_i joins core set Ω

2. Label propagation

   Initial k = 1

   While $\Omega \ne \Phi$ do

      Choose a core o from Ω, then $\Omega_{cur} = \{o\}$, $C_k = \{o\}$,k=k+1, $\Gamma = \Gamma - \{o\}$

      While $\Omega_{cur} \ne \Phi$ do

         Choose a core p from Ω_{cur}, then $C_k = C_k \cup (N_\varepsilon\{p\} \cap \Gamma)$

         Update $\Gamma = \Gamma - (N_\varepsilon\{p\} \cap \Gamma)$, $\Omega_{cur} = \Omega_{cur} \cup (N_\varepsilon\{p\} \cap \Omega)$

Thus, the time complexity of range query is $O(\theta n^2)$, where θ is the complexity of the distance measure. Label propagation has $O(n^2)$ time complexity for assigning labels to objects.

AnyDBC

1. Building an initial cluster structure

   Continuously take a set of untouched objects S and perform range query on each object o in S:

   - If o is a core then mark it and its neighbors;
   - Else if o is a noise then put o and its neighbors into the noise list L.

2. Creating a cluster graph G=(V,E)

   Put all primitive clusters into V as nodes and find states of all edges in E :

   Circularly check whether E only has edges of yes or no state.

   - If true, then the graph has been set up; [stopping condition]
   - Else if false, the following steps should be taken:

   Find and merge connected components(determined by yes-edge) of G into a single node. Calculate the state of each edge of the new graph G.

   Calculate the score of each unprocessed objects by calculating node statistic and node degree, where node statistic, node degree and score of object are important concepts introduced by this paper.

   Then find a set of objects with highest scores to perform range queries to each object p:

   - If p is not a core, it should be marked as processed-border
   - Else if p is a core, the states of its and its neighbors' should be updated and they all should be merged into all nodes that contain p.

   Update the states of all edges in E

3. Check outliers

   Check each object q in noise List L whether it is a border or a noise object

Contribution

This paper introduces AnyDBC algorithm which addresss the problem that traditional density-based method has high time and space complexity. Moreover, it is an anytime algorithm, which is able to produce an approximate result in a short time and to refine it until reaching a satisfactory solution. Also, AnyDBC can even work very well in complex and large datasets.

My idea

In order to better understand AnyDBC algorithm, I firstly read the famous but traditional DBSCAN algorithm. Comparing with old method, AnyDBC selects the most meaningful objects to do the range queries work thus saving a lot of time. Though the new method builds a graph and needs to calculate the states of every edge even need to update the states when graph is updated, it will still run faster since the numbers of nodes and edges after fast initial clustering are much more less than the number of queries for each object in DBSCAN. Moreover, it can deal with large dataset since it is an anytime algorithm and it is capable to take only a set of objects to do the clustering work without loading them all in one time.