Clustering: A neural network approach

Marek Kukačka

MFF UK

October 14, 2010

Content

1 Introduction

2 Clustering methods

3 Cluster validity

Introduction

The source article is ...

- written by K.-L. Du
- published in 2010 in the journal Neural Networks (23, pp. 89-107)
- a survey!

Reference:

Du, K.-L.: Clustering: A neural network approach, *Neural Networks, Vol. 23, No. 1. (29 January 2010), pp. 89-107*

Competitive learning

- implemented by two-layer neural network, with lateral inhibition in the output layer
- competitive learning derived by minimizing MSE, with gradient-descent method
- leads to the *WTA* (winner-takes-all) learning rule:

$$\begin{aligned} c_w(t+1) &= c_w(t) + \eta(t)(x_t - c_w(t)) \\ c_i(t+1) &= c_i(t), i \neq w \end{aligned}$$

- this leads to Vector Quantization (VQ) based on simple competitive learning (SCL)

Competitive learning - variations

- different learning rate for each cluster, depending on the number of samples assigned to the cluster ($\eta_i = 1/N_i$)
- k-WTA networks, based on continuous-time Hopfield network

Kohonen network (SOM)

- self-organizing map
- topology-preserving competitive learning model
- neurons organized into a grid (1D, 2D, or more-dimensional)
- works with more sophisticated lateral feedback than WTA (e.g. Mexican hat function or Gaussian function of distance of neurons along the grid)

Kohonen network - features

- useful for visualization of high-dimensional data
- based on heuristics, not on minimization of any objective function
- gives worse clustering than other models (C-means, neural gas, ART 2A)

Kohonen network - variations

- Adaptive subspace SOM (ASSOM)
  - modular, composed of array of SOMS
- Hyperbolic SOM (HSOM)
  - lattice implemented by a regular triangulation of the input space
  - better for more complex input spaces
- SOM models with recurrence: temporal Kohonen map (TKM), recursive SOM (RSOM), SOM for structured data (SOMSD), merge SOM (MSOM)

Learning vector quantization (LVQ)

- unsupervised LVQ - basically VQ
- supervised LVQ - models LVQ1, LVQ2, LVQ3
  - minimizes MSE with rule resembling the perceptron learning

Learning vector quantization - variations

- Optimized LVQ1
  - individual adaptive learning rate for each neuron
  - faster convergence
- LVQ2, LVQ3
  - adapt two neurons at each step instead of one

C-means clustering

- also known as k-means
- approximates the maximum likelihood of the means of clusters
- based on minimizing MSE
- batch mode
  - samples randomly assigned to clusters, then recalculation of cluster means and sample reassignment alternate until convergence
- incremental mode - by simple competitive learning
- disadvantages:
  - stops at local minima
  - number of clusters has to be pre-specified

C-means clustering - variations

- combinations with genetic algorithms and/or simulated annealing
- including concept of prototype utility (LBG-U)
- combination with SOM to find initial prototypes

Mountain clustering

- grids the input space, each grid point is a potential cluster
- for each grid point, potential is computed based on the density of surrounding samples
- grid with the highest potential is chosen as the next cluster, potentials of other grid points are reduced according to their distance
- repeat until potentials are below a threshold
- exponential complexity caused by high dimension and fineness of the grid

Subtractive clustering

- uses data points instead of grid points (to reduce the exponential size of the problem)
- potential of a point depends on distance to all other points: $P_i = (exp(-dist_{ij}^2 \cdot \alpha))$
- adjustment of potentials after choosing the next cluster center depends on β - normalized neighborhood radius
- needed parameters: α, β, ϵ = fraction of the first potential used as the threshold
- only one pass of the data required, deterministic for given data
- C-means of fuzzy c-means can be used to further process the clustering (to avoid having data points as cluster centers)

Neural Gas

- topology-preserving network
- faster convergence than C-means or SOM, at the cost of higher computational effort
- dynamic neighborhood relations defined during learning
  - by competitive Hebbian learning (creates an induced Delaunay triangulation)
  - two prototypes closest to the presented sample are connected
  - edge aging to remove obsolete edges
  - avoids defects observed in SOM
- the dynamic neighborhood relations influence adaptation

ART networks (1)

- adaptive resonance theory (ART) with biological motivation
- unsupervised networks for clustering and associative memory
- vigilance parameter
  - controls threshold of similarity
  - with similarity below the threshold, a pattern is considered a part of the cluster
  - above the threshold, a new cluster is created

ART networks (2)

- two-layered architecture
- short-term memory (STM) and long-term memory (LTM) differential equations characterize the model
- three types of ART implementations:
  - full mode - both STM and LTM equations are realized
  - STM steady-state mode - only LTM is done by diff. equations, STM done by nonlinear algebraic equations
  - fast learning mode - both STM and LTM implemented by nonlinear algebraic equations (inexpensive, most popular)

ART1 model

- works with binary patterns
- order of pattern presentation influences the resulting clustering
- no restrictions on memory capacity
- variations: improved ART1 (IART1), AHN, fuzzy ART, fuzzy AHN, projective ART (PART)

ART2 model

- works with analog or binary input sequences
- more complex, including normalization and noise suppression
- computationally expensive, behavior similar to C-means
- ART 2A variation - faster, better parallelization
- ART-C 2A - able to control the number of created clusters, by adaptive vigilance parameter

ARTMAP model

- supervised
- self-organizing, self-stabilizing, match learning
- faster than BP
- sensitive to the order of training patterns
- for binary inputs
- fuzzy ARTMAP - for binary or analog inputs
- many variations

Other ART models

- ART 3
- distributed dART
- efficient EART
- simplified SART
- symmetric fuzzy ART (S-Fuzzy ART)
- Gaussian ART
- fully self-organizing SART (FOSART)

Fuzzy clustering

- finds natural boundaries in data
- Fuzzy C-means (FCM):
  - clusters are treated as fuzzy sets
  - vectors are assigned to multiple clusters with some degree of certainty
  - fuzzifier parameter - values close to 1 results in crisp partitioning, large values cause more fuzzy partition (typically 1.5 - 2.0)
  - in turns adjusts the membership matrix and the cluster centers, until the change is sufficiently small

Fuzzy C-means - variations

- penalized FCM
- compensated FCM
- weighted FCM
- FCM with partial supervision
- variations using Mahalanobis distance:
  - Gustafson-Kessel algorithm
  - adaptive fuzzy clustering (AFC)
- algorithms based on volume criteria:
  - minimum scatter volume (MSV)
  - minimum cluster volume (MCV)

Fuzzy clustering - variations

- *fuzzy SOM* - learning rate is replaced by cluster membership
- *fuzzy LVQ* - combines ideas of fuzzy membership values for learning rates and structure and self-organizing rules of SOM network
- combinations of fuzzy logic and ART networks

Supervised clustering (1)

- creates clustering based on the input as well as the input patterns
- class membership is the output in case of classification problems
- examples: LVQ algorithms, ARTMAP family, supervised C-means

Supervised clustering (2)

Techniques:

- combining the input and output pattern into one input vector, then unsupervised clustering
  - parameter for balancing the output and input needed
- augmenting the learning rule
  - the case of supervised LVQ

The under-utilization problem (1)

- also called *dead unit problem*
- trivial solution, not always effective: initialize the prototypes with random samples

The under-utilization problem (2)

Competitive learning with *conscience*:

- leaky learning strategy: all prototypes are updated, non-winners with much slower learning rate
- conscience strategy: increase *conscience* of frequent winners, which causes penalty to be added to its distance from the input
- result: each unit wins with approximately equal probability
- frequency sensitive competitive learning (FSCL) - ensures prototypes are updated with similar probability, via keeping count of the number of wins

The under-utilization problem (3)

Rival-penalized competitive learning:

- winning unit is adapted toward the input
- the second-place winner (the *rival*) is adapted by smaller step *AWAY* from the input
- automatically allocates an appropriate number of prototypes - the redundant units are pushed to infinity
- may encounter over-penalization or under-penalization problem
- lotto type competitive learning - all losers are penalized equally

The under-utilization problem (4)

Soft competitive learning:

- *winner-takes-most* - more than one winner, to a degree
- examples: SOM (in early stages), NG, GNG, maximum-entropy clustering, FCM, FCL (fuzzy competitive learning)
- lowers the probability of dead units and entrapment in local minima

Non-Euclidian distance measures

- Euclidian distance clustering:
  - favors hyperspherical cluster of the same size
- Mahalanobis distance:
  - searches for hyperellipsoid shaped clusters, but cluster may be too large or too small
- Hyperellipsoidal clustering (HEC):
  - combines Mahalanobis and Eclidian distance into one measure
- many extensions of C-means and FCM for looking for hyperspherical shells
- different distance measures can be used to detect clusters shaped as lines, planes, circles, spherical shells, ellipses, curves, rectangles, etc.
- even usable for relational data - algorithm: non-Euclidian relational FCM

Hierarchical clustering (1)

Clustering methods described so far are *partitional*:

- hard or fuzzy
- dynamic - points can move between clusters
- distance measure can be chosen to find clusters of desired size and shape
- susceptible to local minima of its objective function
- sensitive to noise and outliers
- usual complexity $O(N)$

Hierarchical clustering (2)

Beside partitonal methods, we have *hierarchical clustering* ..

- consists of a sequence of partitions, usually represented by a tree - dendrogram
- agglomerative or divisive techniques
- less sensitive to outliers
- no need to specify number of clusters
- overlapping clusters cannot be separated
- typical complexity $O(N^2)$ (divisive clustering is more expensive)

Hierarchical clustering (3)

.. and *density-based* clustering:

- groups objects based on density conditions
- complexity of $O(N^2)$
- typical algorithm: DBSCAN

Hierarchical clustering (4)

Various inter-cluster distance measures:

- *single-linkage* - distance calculated from the two closest points in different clusters
- *complete-linkage* - using the farthest distance between points in different clusters
- other: group-average-linkage, median-linkage, centroid-linkage

Hierarchical clustering (5)

Agglomerative clustering:

- starts with N clusters, one point in each cluster
- proceeds by merging the closest clusters in each step
- examples: minimum spanning tree (MST) method, BIRCH method, CURE method, CHAMELEON method

Constructive clustering (1)

- tries to solve the problem of pre-selection of the number of clusters
- simple strategy - perform clustering for various number of nodes, select the one that minimizes a cluster validity measure

Growing cell structures (GCS):

- modification of SOM with node adding/pruning
- each node has a signal counter, increased when the node wins the competition, all counters decay at each step
- new node inserted after a fixed number of steps to the neighborhood of the node with the highest counter value
- nodes with too low counters are pruned

Constructive clustering (2)

Growing grid network:

- similar to GCS, but with rectangular topology

Growing neural gas:

- based on GCS and NG
- generates and removes neurons dynamically
- uses the Hebbian learning rule for adding/removing of lateral connections

Constructive clustering (3)

Other models:

- dynamic cell structures (DCS)
- DCS-GCS algorithm (results similar to GNG)
- life-long learning cell structures (LLCS) - strategy similar to ART
- self-splitting competitive learning (SSCL)

Miscellaneous clustering methods

Expectation-maximization (EM) clustering:

- each cluster is represented by a probability distribution
- can be treated as a fuzzy clustering technique
- maximizes the log likelihood of the probability density function of the mixture model

Kernel based clustering:

- maps patterns into high-dimensional feature space, where the clustering is then performed
- example: support vector clustering

Cluster validity

Criteria based on maximal *compactness* and maximal *separation* of clusters

- for minimalization - depending on in-cluster scatter and between-cluster distance
- example: entropy cluster validity measures

Criteria based on minimal *hypervolume* and maximal *density* of clusters

- fuzzy hypervolume criterion - sum of volumes of all clusters

The end

Thank you for your attention

mkukacka@gmail.com