Approximate Clustering in Very Large Relational Data

eNERF

Chris Leckie          Rao Kotagiri      Jackie Huband    Jim Bezdek       Rick Hathaway
RF CM  Relational Fuzzy (c-means)

NERFCM  Non Euclidean rFCM (aka NERF or LNERF)

kNERF  kernelized NERF: clustering in (badly messed up) relational data

eNERF  extended NERF for (VL=Unloadable) relational data
The eNERF family tree

FCM → RFCM → NERFCM → eNERF

Reformulation  β-spreading  PS & extension

same basic theory for the HCM and PCM cases

Reformulation has 2 objectives

Constrained optimization (FCM) becomes unconstrained (RFCM)

Reduction to the smallest possible number of problem variables
Eliminate $U$ variables (a cxn matrix) by substitution of the FONC for $U$ into $J_m(U,V)$, thereby obtaining $K(V)$, with $V$ a cxp matrix.

$$\min_{U \in M_{\text{fcn}}} \min_{V \in \mathbb{R}^{c \times p}} \left\{ J_m(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \left\| x_k - v_i \right\|^2_A \right\}$$

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}$$

$$d_{ikA} = \left\| x_k - v_i \right\|^2_A$$

$$u_{ik} = \left[ \frac{\sum_{j=1}^{c} \left( (d_{ikA})^2 / (d_{jkA})^2 \right)}{m - 1} \right]^{-1/(m-1)}$$

Distance coupling
The reformulated problem is RFCM

\[
\min_{V \in \mathbb{R}^{cn}} \left\{ K_m(V) = \sum_{k=1}^{n} \left[ \sum_{i=1}^{c} (d_{ikA})^{1/(1-m)} \right]^{1-m} \right\}
\]

\((\text{Equiv.) FONC}(V)\)

\begin{align*}
v_i &= \left( u_{i1}^m, \ldots, u_{in}^m \right)^T / \sum_{k=1}^{n} u_{ik}^m \\
\delta_{ikA}^2 &= \left( D v_i \right)_k - 0.5 \times v_i^T D v_i \\
D &= \left[ D_{jk} \right] = \left\| x_j - x_k \right\|_A^2
\end{align*}

\((\text{Equiv.) "distance" coupling})\)

\((\text{Same formula) FONC}(U)\)

\[
u_{ik} = \left[ \sum_{j=1}^{c} \left( \delta_{ikA} \right)^2 / \left( \delta_{jkA} \right)^2 \right]^{-1/(m-1)}
\]

\((\text{AO of } K_m)\)
Un-kernelized Duality Theory
for c-Means Clustering

Object Data

\[ X = \{ x_1, \ldots, x_n \} \]

Relational Data

\[ D = [d_{ij}] = [r(o_i, o_j)] \]

HCM \quad \leftrightarrow \quad \text{Hard} \quad \leftrightarrow \quad \text{RHCM}

FCM \quad \leftrightarrow \quad \text{Fuzzy} \quad \leftrightarrow \quad \text{RFCM}

PCM \quad \leftrightarrow \quad \text{Poss.} \quad \leftrightarrow \quad \text{RPCM}

Equivalent \iff D_{ij} = \left\| x_i - x_j \right\|_{\text{Eucl.}}^2

Yeow!
D is a Euclidean Matrix, X is a realization of D

\[ \exists X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^{n-1} \quad \text{s.t.} \quad D_{ij} = \sum_{k=1}^{n-1} (x_{ik} - x_{jk})^2 \]

\[ z^T D z \leq 0 \quad \forall z \in \mathbb{R}^n \quad \text{s.t.} \quad \sum z_i = 0 \]

\[ \text{RFCM} \iff \text{FCM} \iff D \text{ is Euclidean} \]

\[ \text{NERF extends RFCM for } D \neq D_{Eu} \]
The heart of NERF c-means

\[ \beta - \text{spread} \ldots D_\beta = D + \beta([1]_n - I_n) \]

\[ D_\beta \begin{cases} \text{Non-Euclidean} \\
\text{Realization in } \mathbb{R}^{n-1} \end{cases} \]

\[ D_\beta \begin{cases} \text{Euclidean} \\
\text{Realization in } \mathbb{R}^s \end{cases} \]

\[ \exists s, 1 \leq s \leq (n-2) \]
\( D = [D_{jk}] = \left\| x_j - x_k \right\|_A^2 \)

\( D_\beta = D + \beta ([1]_n - I_n) = D + \beta M \)

\( \delta_{ikA}^2 = \left( D_\beta v_i \right)_k - 0.5 * v_i^T D_\beta v_i \)

\( D, D_\beta \) are (nxn) matrices, soon to be associated with sample \( |X_n| = n \) when \( X_N \) is VL

This makes extension more difficult than for eFFCM and geFFCM.
Relational Clustering with NERF aka “LNERF”

<table>
<thead>
<tr>
<th>Input</th>
<th>nxn Dissimilarity Matrix $D_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints</td>
<td>$d_{ij} \geq 0$; $d_{ij}=d_{ji}$; $d_{ii}=0$</td>
</tr>
<tr>
<td>Choose</td>
<td>$M=[1]_n-I_n$</td>
</tr>
</tbody>
</table>

$c = \# \text{ of clusters, } 2 \leq c \leq n$

Reformat this like the others effcm, etc

- $\varepsilon_L$ = termination criterion
- $Q_M =$ max. # of iterations
- termination norm $\left\| U(q) - U(q-1) \right\|$
- $q = 0; \beta = 0$  $U_{diff} = 2^*\varepsilon_L$  $U^{(0)} \in M_{fcn}$
While \( U_{\text{diff}} > \varepsilon_L \) and \( q < Q_M \)

\[
v_i(q) = \left( \left( u_1^m(q) \right)^m, \left( u_2^m(q) \right)^m, \ldots, \left( u_n^m(q) \right)^m \right)^T \sum_{j=1}^n \left( u_j^m(q) \right)^m
\]

\[
\delta_{ik} = \left( (D + \beta M) v_i(q) \right)_k - \left( (D + \beta M) v_i(q) \right)^T / 2
\]

If any \( \delta_{ik} < 0 \):

\[
\Delta \beta = \max_{i,k} \left\{ -2 \delta_{ik} \left\| v_i(q) - e_k \right\|^2 \right\}
\]

\[
\delta_{ik} = \delta_{ik} + (\Delta \beta / 2) \cdot \left\| v_i(q) - e_k \right\|^2
\]

\[
\beta = \beta + \Delta \beta
\]

If \( \delta_{ik} > 0 \) (else usual):

\[
u_{ik}^{(q+1)} = \frac{1}{\sum_{j=1}^c \left( \delta_{ik} / \delta_{jk} \right)^{1/(m-1)}}
\]

\[q = q + 1\]

\[
U_{\text{diff}} = \left\| U^{(q)} - U^{(q-1)} \right\|
\]

\[\text{AO of } K_m\]

Works fine ...

if \( D \) is small!

but what if \( D \) is un-loadable (VL)?
Bird's eye view of eNERF Architecture

1. DF
get h features (indices)

2. PS
get n samples (indices)

3. LNERF
literal clusters in D_n

4. xNERF
fuzzy labels in D_{N-n}
1. Algorithm DF: find distinguished features (DFs)

Input

\[ \text{VD } D_{N \times N} : D_{ij} \geq 0 ; D_{ii} = 0 : D = D^T \]

Choose

\[ H = \text{# of candidate rows in } D_N \]
\[ h = \text{# of DFs to select : } h \leq H \]
\[ m_1 = 1 \quad (\text{object}) \quad o_1 = 1^{\text{st}} \text{ DF} \]
Get indices \{m_i\} of (h) DFs

DF_1 = o_1 (m_1 = 1)

DF_2 = o_2 (m_2 = j)

DF_3 = o_j (m_3 = k)

DF Selection Rule: GET DF_t = o_t as far from \{o_1, ..., o_{m_t-1}\} as possible by finding maximin of accumulated distance tuples.
$D_H$ contains (c) CS (Dunn) clusters \( \{U_j : 1 \leq j \leq c \leq h\} \),

\[ \Downarrow \]

DF selects \( o_j \in U_j \) for \( j = 1, 2, \ldots, c \)
aside

What are CS clusters?

\[ \min_{1 \leq i \leq c} \left\{ \min_{1 \leq j \leq c, j \neq i} \frac{\text{dis}(X_i, X_j)}{\max_{1 \leq k \leq c} \{ \text{dia}(X_k) \}} \right\} \]

X has CS clusters \[ \iff \max_{U \in M_{\text{hcn}}} \left\{ V_{\text{Dunn}}(U) \right\} > 1 \]

\[ U \in M_{\text{hcn}} \leftrightarrow \{X_1, \ldots, X_c\} \text{ is any crisp partition of } X \]
2. Algorithm PS: Relational Progressive Sampling

**Input**

\( D_{hxN} \) : The DFs identify \( h \) rows of \( D_N \)

**Pick**

\( b = \# \) of EC histogram intervals

\( p = \) initial sample \% (of \( N \))

\( \Delta p = \) percentage increment

\( \varepsilon_{ps} = \) termination threshold

**Compute**

Initial \# of samples

\[ n = \left\lfloor \frac{(pN)}{100} \right\rfloor \]

\( n/b = \) initial \# of samples per bin
PS1. Randomly choose (w/o repl.) \( n \) columns \( \{c_1, \ldots, c_n\} \) of \( D_{hxN} \)

PS2. For each DF (k=1 to h)
   Define EC histogram bins of unequal widths for DF \( m_k \) using \( \{d_{mk^c_j}\} \)
   
   - Sort initial sample \( \{d_{mk^c_j}\} \)
   - Construct \( b \) \([\ast, \ast]\) bins with order statistics

\[
\begin{bmatrix}
0, d\left(1 + \left[\frac{n}{b}\right]\right) \\
 d\left(1 + \left[\frac{n}{b}\right]\right), d\left(1 + \left[\frac{2n}{b}\right]\right) \\
 \vdots \\
 d\left(1 + \left[\frac{(b-1)n}{b}\right]\right), \infty
\end{bmatrix}
\]

- Many narrow bins in dense data areas
- Fewer (wider) bins in sparse data areas
- Endpoints vary from feature to feature
- Endpoints depend only on original n-sample
PS3. For each bin \((i = 1 \text{ to } b)\)
   For each DF \((k=1 \text{ to } h)\)
   - Get full count for bin \((i, m_k)\) using \(\{d_{mk}\}\)
   - Get sample count for bin \((i, m_k)\) using \(\{d_{mkc_j}\}\)

\[
\text{div}_k = n \sum_{i=1}^{b} \left( \frac{N_i^k}{N} - \frac{n_i^k}{n} \right) \log\left( \frac{nN_i^k}{Nn_i^k} \right)
\]
PS5. WHILE $\exists k [ \text{div}_k > F^{-1}(1 - \varepsilon_{PS}) ]$

$\Delta n = \min \{ N-n, (\Delta pN/100) \}$

$n = n + \Delta n$

Randomly choose $\Delta D = \Delta n$ unused columns of $D_{hxN}$

$D_n = D_n + \Delta D$

Return to PS3
4. **XNERF**: Why extension is more difficult

### eFFCM and geFFCM

Run LFCM on Sample $X_n$ to get prototypes $V_n$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_j$</th>
<th>$x_g$</th>
<th>$x_m$</th>
<th>$x_n$</th>
<th>$x_p$</th>
<th>$x_q$</th>
<th>$x_N$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1.  
2.  
$p$

Use FONC for $U$ in FCM to label remaining points

$$U(0) = G(V_n, x_k)$$
4. **XNERF**: Why extension is more difficult

**eNERF**

Run LNERF on Sample $D_n$ to get "prototypes" $V_n$

$D_N = \bullet \quad D_n = \bullet$

$U_n$ and $V_n$ are **fixed**, but here is what's left - now what?

Which (n+1) of the N values for $o_j$ to use, and how?
4. **XNERF** : The general idea

\[ \beta_n = \text{final value of } \beta \]

\[ \beta_n = (\beta_n, \ldots, \beta_n)^T \]

\[ D_{\beta_n} = D_n + \beta_n M \]

\[ z_j = (d_j + \beta_n) \]

\[ D_j = \begin{bmatrix} D_{\beta_n} & z_j \\ z_j^T & 0 \end{bmatrix}_{(n+1) \times (n+1)} \]

Get new \( V_{n+1} \), new \( \{\delta_{ik}\} \), and labels for \( o_j \). I can’t explain all the math left in the time I have today .... so
4. XNERF in brief

Inputs

- From PS: \( D_n \)
- From LNERF on \( D_n \): \( \beta_n, U_n, V_n \)

Choose

\[ c ; m ; \varepsilon_x ; \left\| U^{(q)} - U^{(q-1)} \right\| \]

Do

xNERF iteration for \( j=n+1 \) to \( N \)

Inefficient - but works!

Output

\[ U_{x\text{NERF}} \mapsto U_{\text{app}} = [U_{\text{LNERF}} \mid U_{x\text{NERF}}] \text{ for } D_N \]
Ex.1 Gene Product Data (GPD194)

GPD194 extracted from "ENSEMBL"

ENSEMBL has gene product Clusters discovered by clustering sequence similarity data with a Markov clustering algorithm (MCL)

## The GPD194_{12.10.03} data set of c=3 MCL Clusters

<table>
<thead>
<tr>
<th>Ensembl ID</th>
<th>Protein Family</th>
<th>Biological Functionality</th>
<th># of genes In Family</th>
<th># of Individual Human GPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENSF00000000339</td>
<td>myotubularin dephosphorylation</td>
<td>7</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>ENSF0000000073</td>
<td>receptor precursor cell division and cell differentiation</td>
<td>7</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>ENSF0000000042</td>
<td>collagen alpha chain strength and structure of connective tissue</td>
<td>13</td>
<td>86</td>
<td></td>
</tr>
</tbody>
</table>

The extraction date (12.10.03) is important because the GO is updated every 30 minutes!
**"Benchmark" partition : Crisp $U_{\text{MCL}}$ for GPD194**

<table>
<thead>
<tr>
<th>Protein</th>
<th>1</th>
<th>21</th>
<th>22</th>
<th>108</th>
<th>109</th>
<th>194</th>
</tr>
</thead>
</table>
| $U_{\text{MCL}}$ | $\begin{bmatrix}
1 & \ldots & 0 \\
0 & \ldots & 1 \\
0 & \ldots & 0
\end{bmatrix}$ & $\begin{bmatrix}
0 & \ldots & 1 \\
1 & \ldots & 0 \\
0 & \ldots & 0
\end{bmatrix}$ & $\begin{bmatrix}
0 & \ldots & 0 \\
0 & \ldots & 1 \\
1 & \ldots & 1
\end{bmatrix}$ |

- $F_1 = \text{myotubularin}$ (muscle tissue)
- $F_2 = \text{receptor precursor}$ (cell division)
- $F_3 = \text{collagen alpha chain}$ (connective tissue)
Dissimilarity Images for GPD

LOS4 Input Data

Scrambled indices

c = 3 clusters visually apparent

Randomly sampled to test xNERF (omit PS here)
Fuzzy partition Images for GPD

\[ e\text{NERF}(D_{\text{scram}}) = U_{\text{scram}} \rightarrow U_{\text{unscram}} \rightarrow I(U) = [1] - (U^T U / \max\{U^T U\}) \]
These clusters are very compact and well-separated.
D = D_{40,000 \times 40,000} is the (Euclidean) distance matrix for X \times X

Storage of D = 12.2 GB vs MATLAB limit 250 MB
... so... D is unloadable (VL) for this platform

Progressive sampling

\[
\begin{align*}
H &= 10 \text{ DF candidates} \\
h &= 1 \text{ DF} \implies m_1 = 1 \forall H \\
b &= 10 \text{ EC bins} \\
p &= \Delta p = 1\% \implies n = \Delta n = 400 \\
\varepsilon_{PS} &= 0.80
\end{align*}
\]

Terminated after 1 increment \implies n_{\text{final}} = 800

LNERF on D_{800 \times 800}

\[
\begin{align*}
m &= 2 \\
c &= 8 \\
\varepsilon_L &= 10^{-5}
\end{align*}
\]

LNERF terminated after 3 iterations

xNERF on (D_N - D_n)

\[
\begin{align*}
m &= 2 \\
c &= 8 \\
\varepsilon_x &= 10^{-3}
\end{align*}
\]

49 successive calls on 800x800 chunks
Approximation error in (derived) crisp labels

$$E_{app} = 0.5 \cdot \sum_{i=1}^{8} \sum_{k=1}^{40000} \left| H(U_{lit})_{ik} - H(U_{app})_{ik} \right| / 40000 = 0$$

Training error in (derived) crisp labels

$$E_{tr} = 0.5 \cdot \sum_{i=1}^{8} \sum_{k=1}^{40000} \left| (U_{true})_{ik} - H(U_{app})_{ik} \right| / 40000 = 0$$

Error of eNERF fuzzy labels as approximation to LNERF fuzzy labels

$$\left\| U_{lit} - U_{app} \right\|_F = 0.2548$$

This is the TOTAL error for 8x40,000=320,000 memberships
Summary and Conclusions: NERF Family

**NERF**
- **Input**: nxn Dissimilarity Matrix $D_n$
  - (neither transitive nor Euclidean)
- **Output**: Fuzzy Clusters in $D_n$
- **Limit**: Storage required for $D_n$

**eNERF**
- **Input/Output/Limit**: (SAME as NERF)
- **Does “feature extraction” in relational data**
  - ✅ VAT $\Rightarrow$ good value for [Gaussian K] ($\sigma$)
  - ✅ VAT $\Rightarrow$ good guess about c (in X or D)
  - $k\text{NERF} \rightarrow \text{NERF as } \sigma \rightarrow \infty$

**kNERF**
- **Input**: VL NxN Dissimilarity Matrix $D_N$
- **Output**: Approx NERF Clusters in $D_N$
- **Limit**: No size limit
  - ✅ Easily kernelized to get VL $k\text{NERF}$
Yet to do ...

1. CLUSTER VALIDITY (how many clusters to seek) in VL data?

2. Need a more efficient EXTENSION procedure than xNERF

3. NERF algorithm for RECTANGULAR (MxN) dissimilarity data

4. Perform eNERF testing on REAL VL data (useful applications)
Thanks mates!

G’Day