UMAP

富田　直輝

2019 年 5 月 17 日

1　はじめに

次元圧縮はデータの情報量を削減するのと同じことになっている．手法としては統計的手法である主成分分析や多様体学習である ISOMAP や t-SNE 等が挙げられている．特に t-SNE は kaggle でもよく使われている手法になっておりニューラルネットワークの初期値として与えられたりするなど汎用性は高い．しかし、今回は t-SNE より実行速度が速く，また kaggle でも注目されつつある UMAP について紹介したいと思う (前提知識が多いので手法の紹介だけになるが)．

2　データの多様体への変換と近似

データを多様体の空間として考えたい．多様体の空間は $\mathbb{R}^n$ として考える。そこでデータは式 (1) で表されることにする．

$$X = X_1, , X_N \tag{1}$$

3　空間表現

局所的な空間を表現したい．

定義 1　圏有限群の $sets[n] = 1, \cdots, n$ を持つ．

定義 2　単体の集合は定義 1 によって与えられるマップからの関手である．

単体の集合は多様体空間の学習の組み合わせ方法で説明できる．対照的に距離空間を処理しと距離情報を持ちえる同様な構造を必要としている．しかし、これを表現する方法はすでに Spivak によって証明されている．

ここで I は $(0, a); a \in (0, 1]$ の間隔によって与えられる位相で $(0, 1] : 0 < a \leq 1\mathbb{R}$ の間隔である．オープンセットの圏はいくつかの順序集合のカテゴリーにとっての自然な方法でグロダンティーク位相になじませることができる．

定義 3　I における前層 $\mathcal{P}$ は集合への関手 I^{op} である．ファジー集合は I の層であるので全てのマップ $\mathcal{P}(ab)$ は単射である．

不完全な単調写像からまた層における限は定義されているが，直接限は定義されていない．そこで次の定義

を行う．

定義 4　ファジー集合における集合のファジーは I における層の部分集合である

また、ファジーの単体の集合の定義は次のように定義される

定義 5　ファジ-の単体の集合:$sFuzz$ は $\triangle^{op}$ から Fuzz への関手によって与えられるオブジェクトと，自然変換によって与えられる射の集合である．

ファジーの単体集合は密着空間として与えられ，そして $\triangle \times I$ は積位相を持つ層としてみられる．

定義 6　擬距離空間 (extended-pseudo-metric space) は次のように定義される．

1. $d(x,y) \geq 0 x = y, d(x,y) = 0$;
2. $d(x,y) = d(y,x)$;
3. $d(x,z) \leq d(x,y) + d(y,z) or d(x,z) = \infty$

これらの擬似空間距離の集合を EPMet と呼び擬似空間距離と収縮写像をもつ．また，有限擬似空間距離を FinEPMet と呼ぶ．

また，Spivak は随伴関手として　sFuzz と EPMet の集合の間である Real と Sing を定義した．Real は式 2 で定義される．

$$Real(\triangle^n_a) =^t riangle\{(t_0, \cdots t_n) \in \mathbb{R}^{n;1} | \sum_{i=0}^{n} t_i = -log(a), t_i \geq 0\} \tag{2}$$

また，$a \leq b$ なら，位相空間内の $\triangle^n_a \to \triangle^m_b$ と表され．$\triangle$morphonismσ : [n] $\to$ [m] として表される．

$$(x_0, x_1, \cdots x_n) \mapsto \frac{\log(b)}{\log(a)} \left(\sum_{i_0 \in \sigma^{-1}(0)} \sum_{i_0 \in \sigma^{-1}(1)} , \cdots , \sum_{i_0 \in \sigma^{-1}(m)} \right) \tag{3}$$

この写像は $0 \leq a \leq b$ における $\log(b)/\log(a) \leq 1$ であるときは非拡大的である．

$$\mathrm{Real(X)} =^{\triangle} \mathrm{colim}_{\triangle^{\mathrm{n}}_{<\mathrm{a}}} \to \mathrm{X} \mathrm{Real}(\triangle^{\mathrm{n}}_{<\mathrm{a}}) \tag{4}$$

集合の $\triangle \times I$ における擬似空間距離の随伴関数を Sing とする．

$$\mathrm{Sing}(Y) : ([n], [0,a] \mapsto \mathrm{hom}_{\mathrm{EPMet}}(\mathrm{Real}(\triangle^{\mathrm{n}}_{<\mathrm{a}}))) \tag{5}$$

また有限距離について，Fin-Fuzz と FinReal，FinSing を定義する．

定義 7　関手 FinReal の定義:Fin-sFuzz $\to$ FinEPMet

$$FinReal(\triangle^n_{<a}) = (x_1, x_2, \cdots, x_n, d_n) \tag{6}$$

このとき，

$$d_a(x_i, x_j) = \begin{cases} -log(a) & i \neq j \\ 0 & otherwise \end{cases} \tag{7}$$

また，次のように定義することもできる．

$$\mathrm{FinReal(X)} =^{\triangle} \mathrm{colim}_{\triangle^{\mathrm{n}}_{<\mathrm{a}} \to \mathrm{X}} \mathrm{FinReal}(\triangle^{\mathrm{n}}_{<\mathrm{a}}) \tag{8}$$

定義 8　関手 FinSing:FinEPMet → Fin-sFuzz は次のように定義される．

$$FinSing(Y) : ([n], [0, a]) \mapsto hom_{\mathrm{FinEPMet}(FinReal(\triangle^n_a)Y)} \tag{9}$$

FinReal と FinSing は随伴関手として与えられる．

定義 9　$X = X_1, \ldots, X_N \in \mathbb{R}^n$

$\{(X, d_i)_{i=1\ldots N}\}$ は擬似距離空間の一部であり，式 (10) で表される

$$d_i(X_i, X, j) = \begin{cases} d_{\mathcal{R}}(X_i, X_k) - \rho & \text{if } i = j \text{ or } i = k \\ \infty & otherwise \end{cases} \tag{10}$$

このとき ρ は X_i の近傍 (Nearest Neigbor) を示し，$d_{\mathcal{M}}$ は多様体における曲線距離 (geodesic distance) を示す．X のファジートポロジカルの表現は式 ()**??**) で示される．

$$\mathbf{??} \bigcup_{i=1}^{n} FinSing((X, d_i)) \tag{11}$$

このファジー集合によってグローバルな構造とローカルな構造を示していることになる．

4　低次元表現の最適化

定義 10　(A, μ) と (A, ν) によって与えられるクロスエントロピーは次のように定義される．

$$C((A, \mu), (A, \nu)) \stackrel{\triangle}{=} \sum_{a \in A} (\mu(a) \log \frac{\mu(a)}{\nu(a)} + (1 - \mu(a)) log \frac{1 - \mu(a)}{1 - \nu(a)}) \tag{12}$$

これは，t-SNE の最適化と似ているところがある最終的な誤差関数は次のように表される．

$$C_l(X, Y) = \sum_{i=1}^{l} \lambda_i C(X_i, Y_i) \tag{13}$$

5　UMAP の可視化

本手法 g は k 近傍方を基にした手法となっている．

段階は 2 つに分かれており，1 段階目においては重み付きの k-neigber グラフを作成する．2 段階目においては低次元における最適化を行う．

5.1　グラフの初期化

可視化を実装するためにまず，重み付き k 近傍グラフを用いる．ここで次のデータセットを $X = x_1, , x_N$，距離を $d : X \times X$ で定義する．ハイパーパラメータ k は d 距離 d における近傍を決めるための値である．ここで ρ_i と σ_i について定義する．

$$\rho_i = \min\{d(x_i, x_i j) | 1 \geq j \geq k, d(x_i, x_i j) > 0\} \tag{14}$$

$$\sum_{j=1}^{k} \exp\left(\frac{\max(0, d(x_i, x_j) - \rho_i)}{\sigma_i}\right) = \log_2(k) \tag{15}$$

ここで重み付き有効非巡回グラフ $\bar{G} = (V, E, w)$ を定義する．頂点集合 V は X の単純集合である．またエッジ E は (x_i, x_{ij}) で表され，重み w は式 (16) で定義される．

$$w((x_i, x_{ij})) = \exp\left(\frac{\max(0, d(x_i, x_j) - \rho_i)}{\sigma_i}\right) \tag{16}$$

隣接行列は式 16 によって定義される．また，対称行列は次で定義される．ここで，$\circ$ はアダマール行列を示す．これにより無向巡回グラフを生成することができる．

$$B = A + A^{\mathsf{T}} - A \circ A^{\mathsf{T}} \tag{17}$$

5.2　グラフの配置

t-SNE の時と同じく斥力と引力を考えてグラフの配置を行っていく．

引力は式 18 で示される．

$$\frac{-2ab\|y_i - y_j\|_2^{2(b-1)}}{1 + \|y_i - y_j\|_2^2} w((x_i, x_j))(y_i - y_j) \tag{18}$$

ここでの a，b はハイパーパラメータとなっている．

斥力は式 (19) で表されることにする．

$$\frac{b}{(\epsilon + \|y_i - y_j\|_2^2)(1 + \|y_i - y_j\|_2^2)} (1 - w((x_i, x_j)))(y_i - y_j) \tag{19}$$

ϵ は分裂を防ぐためのごく小さい数字となっている．

6　実装とハイパーパラメータ

UMAP のアルゴリズムの全容は Algorithm1 で示される．

次に Algorithm2 を示す．これは次元圧縮後の値の初期化を示す．

$\log(n)$ に最適な ω を求める方法を Algorithm3 に示す．

グローバル的な構造を保つための処理を重み付きグラフとラプラシアン行列を用いて表現する．これは Algorithm4 で示される．

Algorithm 1 UMAP Algorithm

function UMAP(X, n, d, min-dist, n-epochs)

for all $x \in X$ **do**

Fs-set[x] $\leftarrow$ LocalFuzzySimpricalSet(X,x,n)

top-rep $\leftarrow \cup_{x \in X}$ fs-set[x]

$Y \leftarrow$ SpectralEmbedding(top-rep,d)

$Y \leftarrow$ OptimizeEmbedding(top-rep,Y,min-dist,n-epochs)

return Y

Algorithm 2 Constructing a local fuzzy simplical set

function LocalfuzzySimplicalSet(X,x,n)

knn,knn-dists $\leftarrow$ ApproxNearestNeighbors(X, x, n)

$\rho \leftarrow$ knn-dists[1] $\triangleright$ Distance to nearest neighbor

$\sigma \leftarrow$ SmoothKNNDist(knn-dists, n, σ) $\triangleright$ Smooth approximator to knn-distance

fs-set$_0 \leftarrow X$

fs-set$_1 \leftarrow \{([x, y,]0) | y \in X\}$

for all $y \in$ knn **do**

$d_{x,y} \leftarrow 0, dist(x, y) - \rho/\sigma$

fs-set$_1 \leftarrow$ fs-set$_1 \cup ([x, y], exp(-d_{x,y}))$

return fs-set

Algorithm 3 Compute the normalizing for distance σ

function SmoothKNNDist(knn-dists,n,ρ)

Binary search for σ such that $\sum_{i=1}^{n} \exp(-knn - dists_i - \rho)/\sigma) = \log_2(n)$

return σ

Algorithm 4 Spectral OptimizeEmbedding for initialization

function SpectralEmbedding(top-rep,d)

$A \leftarrow$ 1-skeleton of top-rep expressed as a weighted adjacency matrix

$D \leftarrow$ degree matrix for the graph A

$L \leftarrow D^{\frac{1}{2}}(D - A)D^{\frac{1}{2}}$

evec $\leftarrow$ Eigenvectors of L (Sorted)

$Y \leftarrow$ evec[$1d + 1$]

return Y

最後に最適化関数を与える．ここでは交差情報量を使うことによって与えられる．

$$C((A, \mu)(A, \nu)) = \sum_{a \in A} \mu(a) \log(\frac{\mu(a)}{\nu(a)}) + (1 - \mu(a)) \log(\frac{1 - \mu(a)}{1 - \nu(a)}) \tag{20}$$

$$= \sum_{a \in A} \mu(a) \log(\mu(a)) + (1 - \mu(a)) \log(1 - \mu(a)) - \sum_{a \in A} \mu(a) \log(\nu(a)) + (1 - \mu(a)) \log(1 - \nu(a)) \tag{21}$$

ここで最小化したい式は式で与えられる．

$$P(x_i) = \frac{\sum_{a \in d_0(a)=x_i} 1 - \mu(a)}{\sum_{b \in d_0(b)=x_i} 1 - \mu(b)} \tag{22}$$

ここで $\mu(\alpha)$ を $\nu(\alpha)$ にしたがって更新したい

定義 11　定義 $\phi : \mathbb{R}^d \times \mathbb{R}^d \leftarrow [0,1]$ $\mathbb{R}$ の 2 点間における 1 次元の単体のメンバーシップ関数の近似は式 23 で与えられる．

$$\phi(x,y) = (1 + a(\|x-y\|_2^2)^b)^{-1} \tag{23}$$

また，Ψ は非線形最小二乗法によって与えられる．

$$\Psi(x,y) = \begin{cases} 1 & if \|x-y\| \leq mindist \\ \exp(-(\|x-y\|_2 - mindist)) & otherwise \end{cases} \tag{24}$$

これによって埋め込みのための最適化は Algorithm5 で与えられる．

Algorithm 5 Optimizing the embeding

function OPTIMIZEEMBEDDING(top-rep,Y,min-dist,n-epochs)
  $\alpha \leftarrow 1.0$
  Fit ϕ from Φ defined by min-dist
  for $e \leftarrow$ 1,n-epochs **do**
    for all $([a,b],p) \in top - rep_1$ **do**
      if Random() $\leq$ **then**
        $y_a \leftarrow y_a + \alpha \dot{\triangle}(\log(\phi))(y_a, y_b)$
        for $i \leftarrow$ 1, ,n-neg-samples **do**
          $c \leftarrow$random sample from Y
          $y_a y_a + \alpha \dot{\triangle}(\log(1-\phi))(y_a, y_c)$
    $\alpha \leftarrow 1.0 - e/$n-epochs
  return Y

参考ページ

原論文:https://arxiv.org/pdf/1802.03426.pdf
日本語訳:http://cympfh.cc/paper/UMAP.html
グラフ行列の説明:https://mathtrain.jp/graphmatrix
UMAP のドキュメント:https://umap-learn.readthedocs.io/en/latest/