LDA を含むと 3 種のクエリ情報取得方法について

木村　健

平成 30 年 10 月 24 日

1　このドキュメントについて

本ドキュメントでは現状で使用可能なデータ（D 系データとする）について、クエリ情報と個人識別情報から共通する「クラスタ」[*1]を抽出する手法について叩き台としての案を 3 種述べる。これらの方法のうち一つは LDA(Latent Dirichlet Allocations) を用いた今までの調査の延長の方法である。これに対しもう一つの方法は NN を用いた、やや連想ネットワーク的な使い方を取り入れて無理やり教師なし学習を教師あり学習に持っていくタイプの手法である。そして最後に FSA(Finite State Automata) を用いたやや古典的な手法について提案してみる。旧来のコンピュータサイエンスっぽいやり方とも言える。

三つの方法は同じタスクをこなすことに関して共通しており、入力と出力はほぼ一緒である。また互いに手法の一部を使わせてあげたり、使わせてもらったりして、複合的な手法へと変化する可能性を秘めていると筆者は思っている。

LDA を使うもの以外は筆者の貧困な発想もありイマイチだと自分でも感じているので、忌憚なきアドバイスをいただけたらと思っている。

想定する読者としては過去の LDA.py についてのドキュメントをある程度（さらりとでも）読んでいて、LDA とはなんぞや、くらいの知識がある人を想定している。

2　3 種の方法

まず 3 種の問題に先立って入力と出力を定めたい。

入力　個人匿名化された個人認識 id と発されたクエリーの文字列の対の膨大な集合

出力　個人認識 id のリストの集合と、各リストに付随する連想単語などの付随情報（例えば将棋が好きな人なら「将棋」と言う単語）

いかなる方法を使ってもいいので、このクエリーと個人の関係性から複数の話題に敏感な特定層を抽出したい。それが本タスクの目的である。

2.1　LDA を用いた方法（Method-I）

以前から使っている持橋先生の LDA.py を活用する。特に入出力を作り変えるわけではないが、特に入力を工夫して LDA.py の入力方法にすると同時に出力のうち特定の行列を使い、クラスタを抽出する。

[*1] ここでクラスタというのは、将棋クラスタとか、アニメクラスタとか言う、Twitter などでの同じ趣味を持つ人の集まりを想定している。

2.1.1 入力

まず、クエリーを定式化する。

$$q_i = \cup_{j=1}^{L_i} w_{ij}$$

$$Q = \cup_{i=1}^{N} (q_i, u_i)$$

この時 u_i にはユーザー識別 id が入っているわけだが、そのユーザーが発したクエリーと Python 的言い方をすればタプルになったものの集合と思っていただければいい。

$$U = 1, 2, \ldots, U_{max}$$

今ユーザーが U_{max} しかいないとした時に、クエリーの集合 Q は U により切断されてユーザーごとのクエリー部分集合となる。

$$q_j = \cup(q_i, u_i) | \text{where: } u_i = j$$

全てのユーザ U について q_j を並べたものを LDA の疎行列（WD 行列）とする。ここで WD のうち W は各クエリに属する単語、D はドキュメントであるが、ここではユーザーとする。

直感的にユーザー u のクエリ集合に対応する単語頻度ベクトルであることがわかると思う。各クエリの単語頻度ベクトル（単語が一つなら one hot vector）について、これをベクトル和を取ったものとなる。

2.1.2 出力

LDA の出力は通常はトピックに属する単語群の集合とすることが多い。ただ LDA では様々な情報が付随して計算され、例えば各 LDA において $k = 100$ をトピック数とすれば、100 個のトピックそれぞれについて各ドキュメント（＝ユーザ）においてどのような分布を持つか、などの情報もある。

- ND $= R^{D \times K}$ （ドキュメント数 x トピック数）

ここからユーザ集合（最初の意味でのクラスタ）を抽出する方法は何通りかあると思うが、ここでは各ユーザについて上位 n の頻度の高いトピックを抽出しておき ($n = 5$ なら $c_{u_1}, c_{u_2}, \ldots, c_{u_5}$ をユーザ u に対して数え上げる) 全ての c_{ui} についてその単語クラスタに属するユーザ u のリストを生成する。

ここでクラスタは最初の意味での共通の意味を持つユーザの集合として、また LDA を計算するときの単位はトピックと言う単語を使いこの二つを識別することにする。

2.2 NN を用いた方法（Method-II）

LDA を用いた方法が割と直感的にクラスタとトピックに属するユーザの関係として綺麗に紐づくのに対し、NN での解析方法というと、通常はそれを教師あり学習に落とし込んで解析しなければならず、あまり良い案が思い浮かばなかった。

思いついた手法としては、LSTM による系列データの入力に対し、系列データの出力を答えとするような DNN を考え、これに対して同じユーザのクエリーをお互いに連想できるように学習する。実際のクラスタの取得に際しては単語を順に LSTM に与えて、連想する単語を芋づる式に取り出してきて、その単語を発しているユーザの集合からクラスタを計算する。

ただし、データに対して指摘があったように複数クエリーを発しているユーザは少なく、また複数単語のクエリーを発しているユーザも少ないため、単語のリンクを辿りながら、あらかじめ用意してある『単語ーユーザ群』の集合を拡張する形で共通の興味をもつクラスタを取得する。

2.2.1 あるいは大胆な案

入力としてユーザ全員の one hot vector を入力とし、クラスタの出現確率を出力とする LSTM RNN ネットワークを考えて、不特定の組み合わせで入力を発火し、出力となる確度についてその入力の組み合わせをクラスタとする場合どのくらいの確率でクラスタが形成されるかを考える。不特定の組み合わせを試し、確率の高い組み合わせを n-best で抽出すれば良いと言うアイデアだが、この確度は 1.0（まさにクラスタである）しか学習データがない気がする。

2.3 FSA を用いた方法（Method-III）

少し確率的現象の把握を離れて、もっとグラフ理論的にクラスタを求めることはできないだろうか、みたいなことを少し考えていて、例えばクエリーの単語 w をノードとし、その間を単語の進行方向にエッジを張り最終的に全クエリーで一種の蜘蛛の巣のようなグラフ構造を作り、各単語のノードにその単語を発したユーザ群をぶら下げておいて、このグラフのエッジが双方向とした場合の disjoint sets についてクラスタとして抽出する。

```
WORD_PRE-----WORD------WORD_POST
             U0
             U1
             U2
             ...
             UN
```

参考文献

[1] Kevin P. Murphy: Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series) The MIT Press, 2012.

[2] Blei, David M. and Ng, Andrew Y. and Jordan, Michael I.: Latent Dirichlet Allocation, J. Mach. Learn. Res. 3/1, volume 3, 2003.