Реализация алгоритма синтеза минимального графа смежности[1]

Максимов А. Г., студент 3 курса СПбГУ, мл. научн. сотрудник лаборатории ТиМПИ СПИИРАН, maksimov.20.43@gmail.com

Аннотация

В работе представлена реализация алгоритма построения одного из минимальных графов смежности алгебраической байесовской сети по заданной первичной структуре.

Введение

Вероятностные графические модели относятся к классу моделей машинного обучения, при этом обучение в них отлично от обучения в нейронных сетях. Алгебраические байесовские сети относятся к классу вероятностных графических моделей. В рамках глобального обучения алгебраической байесовской сети ставится цель синтеза глобальной структуры сети [1], ее описание приводится в данной статье. Сам алгоритм синтеза был подробно описан в [2].

Алгебраические байесовские сети и графы смежности

Алгебраическая байесовская сеть представляет собой ненапрвленный граф, в вершины которого помещены фрагменты знаний, представляющие из себя идеалы конъюнктов с оценками вероятности их истинности [3].

Граф называется графом смежности, если удовлетворяет следующим условиям[4]:

1. между любыми двумя вершинами, веса которых имеют непустое пересечение, существует путь
2. вес любой вершины этого пути содержит пересечение весов стартовой и конечной вершин
3. вес никакой вершины не входит полностью в вес никакой другой

[1]Работа выполнена в рамках проекта по государственному заданию СПИИРАН № 0073-2019-0003, при финансовой поддержке РФФИ, проекты №18-01-00626 и №18-37-00323.

Особенности реализации

В данной реализации можно выделить следующие особенности:

- используемый язык программирования – C++
- реализация не выходит за рамки библиотеки std
- функция component(G, v, q) возвращает компоненту связности сужения графа G на вес q, содержащую вершину v, реализована как поиск в глубину. Пометки о посещении вершин при поиске в глубину удаляются дополнительной функцией dfs_cleaning()
- операции над множествами – объединение и пересечение – реализованы методом small to large в функциях intersection(S_1, S_2) и sunion(S_1, S_2)
- функция delegate(S) возвращает минимальный элемент множества S из-за особенностей контейнера std::set
- из-за особенности функции delegate(S) результаты работы программы не меняются при многократном запуске
- можно добиться построения псевдослучайного минимального графа смежности, модифицировав функцию delegate(S), чтобы она возвращала случайный элемент множества S
- функция output() записывает в файл описание полученного графа на языке Dot
- преобразовать Dot-описание в изображение можно, например, при помощи [5]

Реализация

```
#include<iostream>
#include<vector>
#include<set>
#include<algorithm>
using namespace std;
int const MAX = 100;
int used[MAX]; int n;
vector<int> g[100]; set<int> w[MAX];
set<int> comp; set<set<int>> Q; set<int> s;
```

```
void dfs_cleaning()
{
  for (int i = 0; i < MAX; ++i)
    used[i] = 0;
}
set<int> intersection(set<int> a, set<int> b)
{
  set <int> c;
  if (a.size() < b.size())
  {
    for (auto i : a)
    if (b.count(i))
      c.insert(i);
  }
  else
  {
    for (auto i : b)
    if (a.count(i))
      c.insert(i);
  }
  return c;
}
int delegate(set<int> s)
{
  auto i = s.begin();
  return *i;
}
set<int> sunion(set <int> a, set<int> b)
{
  if (a.size() < b.size())
  {
    for (auto i : a)
      b.insert(i);
    return b;
  }
  else
  {
    for (auto i : b)
      a.insert(i);
    return a;
  }
}
void component(set<int> q, int v)
{
  used[v] = 1; comp.insert(v);
  for (auto i = g[v].begin(); i != g[v].end(); ++i)
  if (!used[*i])
  if (intersection(q, w[*i]) == q)
    component(q, *i);
```

```
}
void qinitial(int n)
{
  set<int> t;
  for (int i = 1; i <= n; ++i)
  for (int j = i + 1; j <= n; ++j)
  {
    t = intersection(w[i], w[j]);
    if (!t.empty())
      Q.insert(t);
  }
}
void input()
{
  int m; int t;
  for (int i = 1; i <= n; ++i)
  {
    cin >> m;
    for (int j = 0; j < m; ++j)
    {
      cin >> t;
      w[i].insert(t);
    }
  }
}
void output()
{
  cout << "graph g {" << endl;
  for (int i = 1; i <= n; ++i)
  for (int j = 0; j < g[i].size(); ++j)
  if (i < g[i][j])
  {
    cout << '"';
    for (auto k : w[i])
      cout << k << " ";
    cout << '"' << " -- " << '"';
    for (auto k : w[g[i][j]])
      cout << k << " ";
    cout << '"' << " ";
    cout << "[label=" << '"';
    for (auto k : intersection(w[i], w[g[i][j]]))
      cout << k << " ";
    cout << '"' << "]" << endl;
  }
  cout << "}" << endl;
}



int main()
```

```
{

  freopen("Text.txt", "r", stdin);
  freopen("Text1.txt", "w", stdout);
  cin >> n;
  input();


  qinitial(n);
  for (auto q : Q)
  {
    s.clear();
    for (int v = 1; v <= n; ++v)
    if (intersection(q, w[v]) == q && (!s.count(v)))
    {
      comp.clear();
      dfs_cleaning();
      component(q, v);
      if (!s.empty())
      {
        int d = delegate(s);
        g[d].push_back(v);
        g[v].push_back(d);
      }
      s = sunion(s, comp);
    }
  }

  output();
}
```

Заключение

В документе была представлена реализация алгоритма синтеза графа смежности на языке C++, позволяющая, в числе прочего, ускорить существующие решения. В дальнейшем алгебраические байесовские сети в рамках исследований социоинженерных атак [6, 7, 8].

Литература

[1] А. Л. Тулупьев, Д.М. Столяров, М.В. Ментюков, “Представление локальной и глобальной структуры алгебраической байесовской сети в Java-приложениях” // Тр. СПИИРАН, 5 (2007), 71–99

[2] В. В. Опарин, А. Л. Тулупьев, “Синтез графа смежности с минимальным числом ребер: формализация алгоритма и анализ его корректности” // Тр. СПИИРАН, 11 (2009), 142–157

[3] Тулупьев А. Л. Алгебраические байесовские сети: локальный логико-вероятностный вывод : Учеб. пособие // СПб.: СПбГУ; Издательство «Анатолия», 2007. 80 с.

[4] Тулупьев А. Л. Алгебраические байесовские сети: глобальный логико-вероятностный вывод в деревьях смежности: Учеб. пособие // СПб.: СПбГУ; Издательство «Анатолия», 2007. 40 с. (Элементы мягких вычислений).

[5] http://www.webgraphviz.com/

[6] Абрамов М. В., Тулупьев А. Л., Сулейманов А. А. Задачи анализа защищенности пользователей от социоинженерных атак: построение социального графа по сведениям из социальных сетей //Научно-технический вестник информационных технологий, механики и оптики. 2018. Т. 18. №. 2.

[7] Багрецов Г. И., Шиндарев Н. А., Абрамов М. В., Тулупьева Т. В.Подходы к автоматизации сбора, структурирования и анализа информации о сотрудниках компании на основе данных социальной сети //Нечеткие системы, мягкие вычисления и интеллектуальные технология (НСМВИТ-2017): труды VII всероссийской научной-практической конференции. Т.1, с. 9-16. (2017)

[8] Хлобыстова А. О., Абрамов М.В., Тулупьев А.Л., Золотин А. А. Поиск кратчайшей траектории социоинженерной атаки между парой пользователей в графе с вероятностями переходов // Информационно-управляющие системы – 2018. – Т. 97. – №. 6.