Софтвер: *Python* модул за интегративно кластеровање података

Одговорно лице: Сања Брдар

Аутори: Сања Брдар, Предраг Лугоња, Драган Летић, Владимир Црнојевић

Развијено: у оквиру пројекта ИИИ 44003 - Интегрисани систем за детекцију и естимацију развоја пожара праћењем критичних параметара у реалном времену (руководилац пројекта Владимир Црнојевић)

Година: 2013.

Примена од: 10.12.2013.

Кратак опис

Кластеровање података је поступак којим се подаци деле у групе према дефинисаном критеријуму сличности и спада у ненадгледани тип машинског учења. Обично је кластеровање први корак у анализи података и има за циљ да открије скривене структуре, објасни и сумира кључне карактеристике у подацима. Иза њега могу да следе друге методе машинског учења. Бројни фактори утичу на крајњи исход кластеровања. Тако резултат кластеровања може да зависи од иницијализације, одабране мере сличности или растојања, параметара алгоритма. За потребе побољшања робустности и тачности алгоритми кластеровања се допуњују делом за интеграцију. Резултати индивидуалних кластеровања се агрегирају у ансамбл за који се потом дефинише поступак добијања коначне партиције скупа података у кластере.

Техничко решење представља софтверски модул за интегративно кластеровање. Модул је имплементиран у програмском језику *Python* (http://www.python.org) и садржи пет различитих алгоритама за интеграцију кластера. Обухваћени алгоритми су *NMF* [1], *CONS* [2], *HGPA* [3], *MCLA* [3], *DICLENS* [4]. Овим софтверским решењем је омогућено једноставно поређење метода, и одабир најбоље за проблем од интереса. Модул је тестиран на разноврсним јавно доступним подацима.

Техничке могућности: Софтверски модул нуди избор алгоритма за интеграцију и начин на који се формира ансамбл кластеровања (из различитих извора, различитим алгоритмима или параметрима модела). Улаз у алгоритам су резултати појединачних кластеровања и листа објеката који се кластерују, а излаз коначна партиција објеката у кластере.

Техничке карактериситке: Софтверски модул је јединствено *Python* решење за интегративно кластеровање, имплементирано коришћењем додатних метода из библиотека *Numpy*, *Orange*, *Scikit-learn*. Једноставно се инсталира и користи.

Реализатори: ФТН - Нови Сад

Корисници: Катедра за Телекомуникације, Факултет техничких наука, Нови Сад

Подтип решења: Софтвер (M85)

Стање у свету

Интегративни приступи у машинском учењу су врло актуелни и развијају се из потребе да се унапреде постојећи алгоритми и испрати експоненцијални раст количине података. За њихову обраду није добро користити само један метод него је боље осмислити стратегију којом се више метода обједињује у коначно решење.

На подручју интеграције кластера разликују се два приступа: рана и касна интеграција. Рана интеграција подразумева фузију података или мера сличности пре формирања кластера. Касне прво формирају индивидуалне кастере па потом врше интеграцију. Истраживачи са Католичког универзитета у Лувену предложили су рану методу интеграције која користи фузију кернел функција на различитим улазним подацима [5]. Најпознатији метод касне интеграције кластера је*„Consensus Clustering“*, који су развили научници са америчког института *„The Broad Institute of MIT“*. Такође велику примену имају и два алгоритма базирани на хиперграфовима које су осмислили научници *Strehl* и *Ghosh* са универзитета у Тексасу. Примери успешне примене су интегративно партиционисање графа протеинских интеракција [6] и сегментација фрејмова у видео секвенци [7].

Конструкција и принцип рада

Интегративним техникама се основне методе кластеровања надограђују и од посебног су значаја у доменима где постоји велика количина података и/или више хетерогених извора података. Софтверски модул садржи имплементације добро познатих алгоритама *CONS, HGPA* и *MCLA* али и два новија *NMF* и *DICLENS.*

1. ***NMF*** - *Nonnegative Matrix Factorization: NMF* методом се резултати појединачних кластеровања интегришу у бинарну матрицу где врста означава кластер, а колона објекат. Јединице у матрици означавају припадност објекта одређеном кластеру. Потом се матрица факторише на две матрице које дају нову репрезентацију припадности објеката кластерима.
2. ***CONS*** - *Consensus clustering: CONS* методом резултати различитих кластеровања се обједињују у консензус матрицу која се потом интерпретира као матрица сличности и служи за крајњу поделу података у кластере.
3. ***HGPA*** - *HyperGraph Partitioning Algorithm: HGPA* метода репрезентује ансамбл кластеровања као хиперграф који се даље партиционише сечењем минималног броја хиперграна.
4. ***MCLA*** - *Meta-Clustering Algorithm: MCLA* метода прво кластеровањем хиперграфа креира мета-кластере, а потом техником гласања одређује припадност објекта мета-

кластеру.

5. ***DICLENS*** - *Divisive Clustering Ensemble: DICLENS* метода користи минимално разапињуће стабло где сваки чвор представља један кластер, а тежине грана су пропорционалне сличности између кластера. Функција за проналажење најбоље партиције сече стабло тако да максимизује компактност кластера.

Модул обухвата и део за процену квалитета добијених кластера. Процена се може вршити помоћу екстерних или интерних мера. Екстерне оцењују кластере помоћу лабела уколико су оне доступне, а интерне се заснивају на критеријума компактности кастера, добре раздвојености међу кластерима или неком другом критеријуму.
Модулом је понуђено је шест мера за евалуацију:

1. ***ARI*** *(Adjusted Rand Index),* екстерна мера
2. ***NMI*** *(Normalized Mutual Information),* екстерна мера
3. ***Silhouette index****,* интерна мера, за сферичне кластере
4. ***Dunns index****,* интерна мера, за сферичне кластере
5. ***Isolation index***, интерна мера, за кластере произвољног облика
6. ***Gap IC-av,*** *(Gap Statistic for average Intra-Cluster distance)* интерна мера, за кластере произвољног облика

ARI, NMI, Silhouette index мере за евалуацију кластера су доступне у пакету *Scikit-learn* и позивају се функцијама из *metrics* модула: *sklearn.metrics.adjusted_rand_score, sklearn.metrics.normalized_mutual_info_score, sklearn.metrics.silhouette_score.* Преостале три методе су имплементиране и саставни су део предложеног техничког решења. Позивају се функцијама из *evaluate* модула *evaluate.dunns_index, evaluate.isolation_index, evaluate.ic_index*

Софтверски модул је тестиран на разноврсним подацима:

1. *UCI Machine Learning* репозиторијум (http://archive.ics.uci.edu/ml/)
2. Синтетички подаци (http://cs.joensuu.fi/sipu/datasets/)
3. Биолошки подаци (http://bioinformatics.rutgers.edu/Static/Supplements/CompCancer/datasets.htm)

На слици 1. дати су примери за позивање свих метода за интегративно кластеровање. Прво је потребно урадити увоз модула са имплементацијама метода: *import nmf, cons, hgpa, mcla, diclens.* Такође наопходан је модул *transform* који је саставни део техничког решења и садржи функције за трансформацију матрица и репрезентација кластера. Следећи корак је креирање ансамбла од индивидуалних кластеровања. Пример показује случај када се користи *K-means* алгоритам. У најједноставнијем случају се анасамбл креира само различитим иницијализацијама. Ансамбл може да се креира и различитим подскуповима обележја или у случају да се користи *Kernel K-means* као основни алгоритам могу се употребити различите кернел функције. Пре позива интегративних метода потребно је резултате индивидуалних спојити у одговарајућу матрицу.

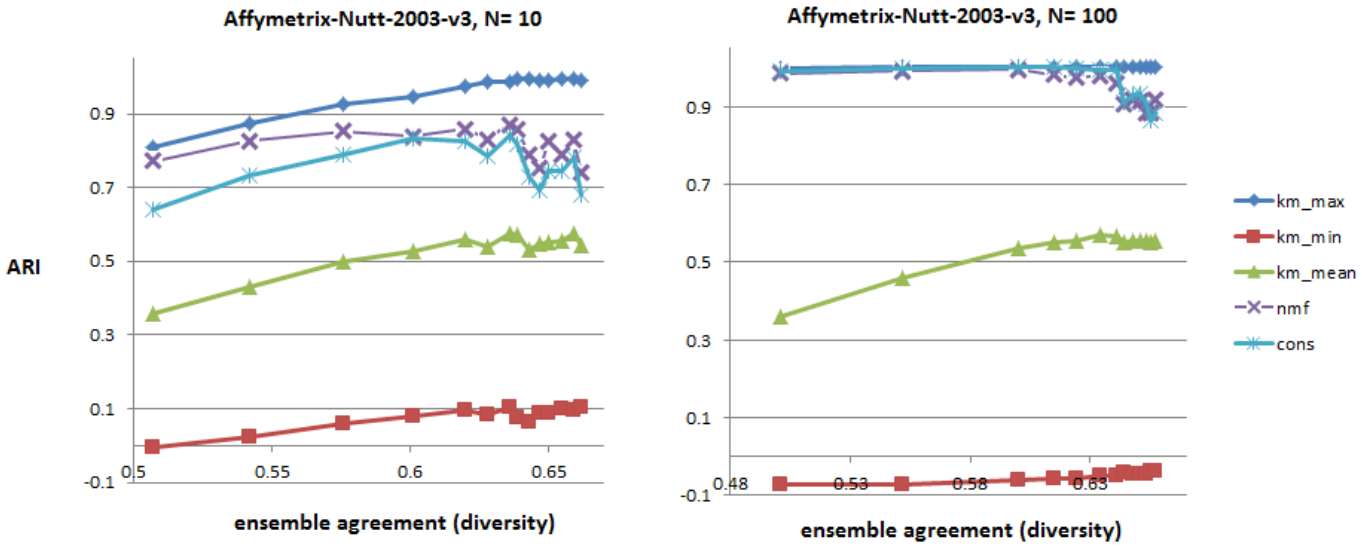
```
# Import methods for integrative clustering
import nmf, cons, hgpa, mcla, diclens
import transform
import Orange
import sklearn
import numpy

data_set  = "iris.tab"
K = 3 # K - number of clusters
clusters_ensemble, merged_clusters = [],[]
connectivity_mat, indicator_mat = numpy.mat(np.zeros((N, N))), np.mat(np.zeros((N, N)))
for data in data_ensemble:
    km = Orange.clustering.kmeans.Clustering(data, K)
    clusters_ensemble.append(km.clusters)
    # Prepare matrix for NMF, MCLA and DICLENS
    merged_clusters += transform.list2matrix(km.clusters)
    # Prepare matrix for Consensus
    connectivity_mat += cons.create_connectivity_mat(objs, transform.list2clusters(km.clusters))
    indicator_mat += cons.create_indicator_mat(objs, transform.list2clusters(km.clusters))
    # Prepare files for HGPA
    graph_file = data_set[:-4] + '.hgr'
    hgpa.append_gfile(graph_file, km.clusters)
# NMF clusters
M = nmf.Nmf(numpy.matrix(merged_clusters), components = K, iterations=100)
cls = nmf.create_clusters(M.W, M.H, objs, thr = 0.0, ctype = 'exclusive')
# CONS clusters
M = connectivity_mat/indicator_mat
cls = cons.create_clusters(M, objs, K)
# HGPA clusters
hgpa.adjust_gfile(graph_file_hgpa, merged_clusters)
cls = hgpa.hgpa(graph_file_hgpa, K, 3)
# MCLA clusters
mcla.create_gfile(graph_file_mcla, merged_clusters)
meta_clusters = mcla.mcla(graph_file_mcla, K)
cls = mcla.create_clusters(meta_clusters, merged_clusters)
# DICLENS clusters
cls = diclens.diclens(merged_clusters, k=K)
```

Сл. 1. Позив метода за интегративно кластеровање

Примена

Тестирањем софтверског модула на различитим типовима података потврђена је могућност широке примене алгоритама за интегративно кластеровање. Само неке од примена су сегментација слике, кластеровање текстуалних фајлова, груписање профила експресије гена. За потребе илустрације одабран је *Affymetrix-Nutt-2003* из скупа билошких података. *Affymetrix-Nutt-2003* садржи експресије гена у случају тумора мозга, где сваки профил експресије одговара једној од две подгрупе: класични или специфични тумор. Алгоритам треба да идентификује два кластера. Графички приказ евалуације примењених кластеровања дат је на слици 1. Евалуација је вршена једном од шест понуђених мера за валидацију - *ARI (Adjusted Rand Index),* која мери слагање између лабела идентификованих кластеровањем и стварних лабела. У датом примеру индивидуална кластеровања добијена су *K-means* алгоритмом. Диверзитет индивидуалних кластеровања остварен је различитим иницијализацијама и подскуповима простора обележја. У случају са подскуповима простора обележја диверзитет се повећава (смањује слагање између индивидуалних кластеровања) тако што се на случајан начин бирају обележја. На примеру се уочава да повећање диверерзитета побољшава резултат до одређене границе. То је потврђено и другим студијама које показују да се тако елиминише шум у улазним подацима и побољшава робустност и тачност у подели података на кластере. Приказани су експериментални резултати са интеграцијом 10 и 100 индивидуалних кластера. У оба случаја интеграцијом се добијају решења боља од просечног у ансамблу. У примеру са величином ансамбла 100 резултати имају веће *ARI* вредности.



Сл. 2. Евалуација *K-means*, *NMF* и *CONS* кластеровања

Неке од могућности које пружа описани софтверски модул:

- Већа поузданост и робустност кластеровања података
- Могућност бирања методе интеграције
- Различите технике за постизање диверзитета индивидуалних кластеровања
- Евалуација по више критеријума

Техничке карактеристике

- Програмски језик ***Python***
- Потребне библиотеке:
  - ***Numpy*** *(Package for scientific computing in Python)* **http://www.scipy.org**
  - ***Orange*** *(Data Mining Toolbox for Python)* **http://orange.biolab.si/**
  - ***sciket learn*** *(Machine Learning in Python)* **http://scikit-learn.org**
- ***Python*** скрипте са методома за интегративно кластеровање (*nmf, cons, mcla, hgpa, disclens*), трансформацију података (*transform*) и евалуацију кластера (*evaluate*)

ЛИТЕРАТУРА

[1] Sanja Brdar, Vladimir Crnojević, Blaž Zupan „Integrative clustering by non-negative matrix factorization can reveal coherent functional groups from gene profile data“, (under review, submitted August 2013)

[2] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub, “Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data”, *Machine Learning*, vol. pp. 91-118, 2003.

[3] Alexander Strehl, and Joydeep Ghosh, ”Cluster ensembles - a knowledge reuse framework for combining multiple partitions” *The Journal of Machine Learning Research*, vol 3., pp. 583-617 2003.

[4] Selim Mimaroglu, and Emin Aksehirli, Diclens: “Divisive clustering ensemble with automatic cluster number”, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9. pp. 408-420, 2012.

[5] Shi Yu, L-C Tranchevent, Xinhai Liu, Wolfgang Glanzel, Johan AK Suykens, Bart De Moor, and Yves Moreau, “Optimized data fusion for kernel K-means clustering”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1031-1039, 2012.

[6] Sitaram Asur, Duygu Ucar and Srinivasan Parthasarathy, “An ensemble framework for clustering protein--protein interaction networks”, vol. 23, pp., i29--i40, 2007.

[7] Luciano Silva, and Jacob Scharcanski, “Video segmentation based on motion coherence of particles in a video sequence”, *IEEE Transactions on Image Processing*, vol. 19. pp. 1036-1049, 2010.