MA-INF 4306 Lab Development and Application of Data Mining and Learning Systems: Big Data
A measure for (weak) convexity

First Progress Meeting

Author: Timon Oerder, Mat.-Nr. 2679722

Institut für Informatik Uni Bonn

s6tioerd@uni-bonn.de

Table of contents

The top-level road-map

1. Define a measure for (weak) convexity

The top-level road-map

1. Define a measure for (weak) convexity
2. Test that measure on artificial data

The top-level road-map

1. Define a measure for (weak) convexity
2. Test that measure on artificial data
3. Test the measure on real-world data

The road so far

Define a measure for (weak) convexity

1. Understand the paper [SHW21] (✓)

The road so far

Define a measure for (weak) convexity

1. Understand the paper [SHW21] (✓)
2. Implement the convex hull algorithm (Algo 1) (✓)

The road so far

Define a measure for (weak) convexity

1. Understand the paper [SHW21] (✓)
2. Implement the convex hull algorithm (Algo 1) (✓)
3. How to store data? ✓

The road so far

Define a measure for (weak) convexity

1. Understand the paper [SHW21] (✓)
2. Implement the convex hull algorithm (Algo 1) (✓)
3. How to store data? ✓
4. How to generate artificial data? ✓

The road so far

Define a measure for (weak) convexity

1. Understand the paper [SHW21] (✓)
2. Implement the convex hull algorithm (Algo 1) (✓)
3. How to store data? ✓
4. How to generate artificial data? ✓
5. What does literature say to the problem ((✓))

The road so far

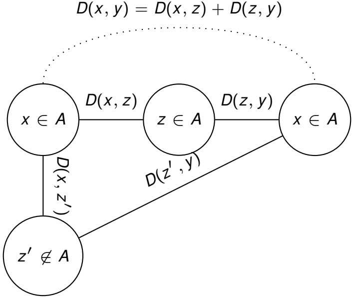
Define a measure for (weak) convexity

1. Understand the paper [SHW21] (✓)
2. Implement the convex hull algorithm (Algo 1) (✓)
3. How to store data? ✓
4. How to generate artificial data? ✓
5. What does literature say to the problem ((✓))
6. Implement sanity measures for convexity ✓

Weak convexity definition [SHW21]

Let (X, D) be a metric space and $\theta \geq 0$. A set $A \subseteq X$ is θ-convex if $\forall x, y \in A$ and $z \in X$ it holds that $z \in A$ whenever $D(x, y) \leq \theta$ and $z \in \Delta_{=}(x, y)$, where

$$\Delta_{=}(x, y) = \{x \in X : D(x, z) + D(z, y) = D(x, y)\}$$



Proposed relaxation (I)

back to Feature List In the paper [SHW21] Definition 1 says:

Let (X, D) be a metric space and $\theta \geq 0$. A set $A \subseteq X$ is θ-convex if $\forall x, y \in A$ and $z \in X$ it holds that $z \in A$ whenever $D(x, y) \leq \theta$ and $z \in \Delta_{=}(x, y)$, where

$$\Delta_{=}(x, y) = \{x \in X : D(x, z) + D(z, y) = D(x, y)\}$$

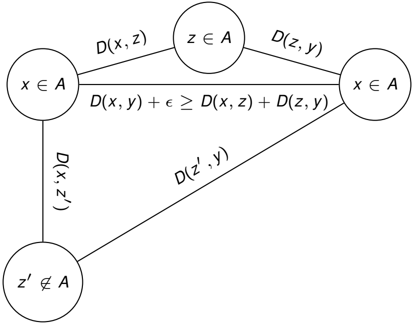
This last part, the triangle inequality is supposed to be relaxed, creating the notion of (θ, ϵ)-convex by defining

$$\Delta_{\epsilon}(x, y) = \{x \in X : D(x, z) + D(z, y) \leq D(x, y) + \epsilon\}$$

Proposed relaxation (II)

back to Feature List

$$\Delta_\epsilon(x,y) = \{x \in X : D(x,z) + D(z,y) \leq D(x,y) + \epsilon\}$$



The area of points in A spanned by x, y, is now shaped like an ellipsis.

Convex hull algorithm

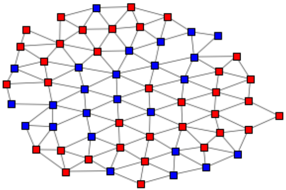
(a) Before application

(b) After application

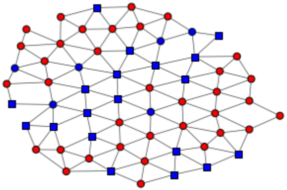
Figure: Application with $\theta = 0.7$, $\epsilon = 0$ (no relaxation) and geodesic distance; class 0 is red; class 1 is blue; assigned class 0 are circles

Implementation

Convex hull algorithm (with relaxation)



(a) Before application



(b) After application

Figure: Application with $\theta = 0.2$, $\epsilon = 0.005$ and euclidean distance; class 0 is red; class 1 is blue; assigned class 0 are circles

▸ Implementation

Convexity sanity measures

For a set of points `A` and the convex hull of this set `conv(A)`.

Convexity sanity measures

For a set of points `A` and the convex hull of this set `conv(A)`.

Measure 1: $convSanity1(A) = \frac{|A|}{|conv(A)|}$

Implementation

Convexity sanity measures

For a set of points `A` and the convex hull of this set `conv(A)`.

Measure 1: $convSanity1(A) = \frac{|A|}{|conv(A)|}$

Implementation

Measure 2: For all (or some sampled) pairs $x, y \in A x \neq y$ we calculate the set of points in between these and calculate a coefficient of points in A and not in A.

Convexity sanity measures

For a set of points `A` and the convex hull of this set `conv(A)`.

Measure 1: $convSanity1(A) = \frac{|A|}{|conv(A)|}$

Implementation

Measure 2: For all (or some sampled) pairs $x, y \in Ax \neq y$ we calculate the set of points in between these and calculate a coefficient of points in A and not in A.

$$convSanity2(A, \epsilon) = 1 - \frac{|(\{\Delta_{(x,y),\epsilon} | x,y \in A, x \neq y\} \cap conv(A)) \setminus A|}{|conv(A)|}$$

Implementation

Current state of programming

- usage of the package `igraph`.

Current state of programming

- usage of the package `igraph`.
- data of `igraph` graphs is stored as `pickle`

Current state of programming

- usage of the package `igraph`.
- data of `igraph` graphs is stored as `pickle`
- random graphs are generated by calculating random points with coordinates, applying a Delauny triangulation [Del34] on them

Current state of programming

- usage of the package `igraph`.
- data of `igraph` graphs is stored as `pickle`
- random graphs are generated by calculating random points with coordinates, applying a Delauny triangulation [Del34] on them
- the proposed relaxation to (θ, ϵ)-convexity is already implemented

Current state of programming

- usage of the package `igraph`.
- data of `igraph` graphs is stored as `pickle`
- random graphs are generated by calculating random points with coordinates, applying a Delauny triangulation [Del34] on them
- the proposed relaxation to (θ, ϵ)-convexity is already implemented
- the code is generic, such that several distance measures can be used and are implemented
  - Minkowski Distances (Euclidean, Manhattan, etc.)
  - Cosine Similarity
  - Geodesic Distance
  - Common Neighbors, Jaccard Distance

Next steps

- Implement the convex hull algorithm ✓
- Generate artificial data and test the algorithm ✓
- Implement sanity and distance measures ✓

Next steps

- Implement the convex hull algorithm ✓
- Generate artificial data and test the algorithm ✓
- Implement sanity and distance measures ✓
- Literature research: What are other measures for compactness, clustering scores?

Next steps

- Implement the convex hull algorithm ✓
- Generate artificial data and test the algorithm ✓
- Implement sanity and distance measures ✓
- Literature research: What are other measures for compactness, clustering scores?
- Can ϵ and θ be modified as a measure?

Next steps

- Implement the convex hull algorithm ✓
- Generate artificial data and test the algorithm ✓
- Implement sanity and distance measures ✓
- Literature research: What are other measures for compactness, clustering scores?
- Can ϵ and θ be modified as a measure?
- Define a measure or some measures to test

Next steps

- Implement the convex hull algorithm ✓
- Generate artificial data and test the algorithm ✓
- Implement sanity and distance measures ✓
- Literature research: What are other measures for compactness, clustering scores?
- Can ϵ and θ be modified as a measure?
- Define a measure or some measures to test
- Find repository data sets to test the measure(s) on

Next steps

- Implement the convex hull algorithm ✓
- Generate artificial data and test the algorithm ✓
- Implement sanity and distance measures ✓
- Literature research: What are other measures for compactness, clustering scores?
- Can ϵ and θ be modified as a measure?
- Define a measure or some measures to test
- Find repository data sets to test the measure(s) on
- Test the measure(s)

Next steps

- Implement the convex hull algorithm ✓
- Generate artificial data and test the algorithm ✓
- Implement sanity and distance measures ✓
- Literature research: What are other measures for compactness, clustering scores?
- Can ϵ and θ be modified as a measure?
- Define a measure or some measures to test
- Find repository data sets to test the measure(s) on
- Test the measure(s)

Thank you for your attention!

What are your questions?

References

B. Delaunay.
Sur la sphère vide. a la mémoire de georges voronoi.
In *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na, No. 6*, pages 793–800, 1934.

Eike Stadtländer, Tamás Horváth, and Stefan Wrobel.
Learning weakly convex sets in metric spaces.
In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II*, page 200–216, Berlin, Heidelberg, 2021. Springer-Verlag.

Weakly convex hull algorithm implementation

' ▸ back

```
def ExtensionalWeaklyConvexHull(g, vertexset, theta, epsilon=0.0):
    g.vs['mark'] = 0
    C = set()
    E = set()
    Q = queue.Queue()
    for vertex in vertexset:
        vertex['mark']=1
        Q.put(vertex)
    while not Q.empty():
        el = Q.get()
        C.add(el)
        thetanh = set(g.vs.select(dnhfactory(g, el, theta)))
        for nhel in C.intersection(thetanh):
            E.add((el, nhel))
            distElNhel = distance_GraphObj(g, el, nhel)
            nhdistel = set(g.vs.select(dnhfactory(g, el, distElNhel)))
            nhdistnhel = set(g.vs.select(dnhfactory(g, nhel, distElNhel)))
            intersect = nhdistel.intersection(nhdistnhel)
            for z in intersect:
                if not z['mark']==1 and z in Delta(g, el, nhel, epsilon=epsilon):
                    z['mark']=1
                    Q.put(z)
```

$N_\theta(x)$ and $\Delta_{=(x,y)}$

back To calculate $N_\theta(x)$ (`thetanh`) as well as $N_{D(x,y)}(x)$ (`nhdistel`) and $N_{D(x,y)}(y)$ (`nhdistnhel`) the following code is used:

```
def dnhfactory(g, vertex, d):
    def rtfn(v):
        return distance_GraphObj(g, vertex, v) < d
    return rtfn
```

To calculate $\Delta_{=(x,y)}$(`Delta`) the following code is used:

```
def triangleDeltaFactory(g, vx, vy, epsilon=0.0):
    def rtfn(v):
        if epsilon == 0.0:
            return distance_GraphObj(g, v, vx) + distance_GraphObj(g, v, vy) == distance_GraphObj(g, vx, v
        return distance_GraphObj(g, v, vx) + distance_GraphObj(g, v, vy) <= distance_GraphObj(g, vx, vy) +
    return rtfn

def Delta(g,x,y, epsilon=0.1):
    return set(g.vs.select(triangleDeltaFactory(g, x, y, epsilon=epsilon)))
```

*SanityMeasure*1

back

```
def convSanity1(A, convexhullA):
    if len(convexhullA) == 0:
        return 0
    return len(A) / len(convexhullA)
```

SanityMeasure2

back

```
def convSanity2sampled(g, A, convexhullA, epsilon=0.0, samplepercentage=0.3):
    if len(convexhullA) == 0:
        return 0
    g.vs['markwp'] = False
    pairslist = []
    numberofsamples = math.floor(len(A)*samplepercentage)
    drawA = [(index, element) for index, element in enumerate(list(A)+list(A))]
    while numberofsamples > 0:
        v1, v2 = RNG.choice(drawA), RNG.choice(drawA)
        if v1[1] == v2[1] or (v1[0]-v2[0]) % len(A) == 0:
            continue
        pairslist.append((v1[1],v2[1]))
        drawA = list(filter(lambda item: item[0]!=v1[0] and item[0]!=v2[0],drawA))
        numberofsamples -= 1
    numberofwrongpoints = 0
    for pair in pairslist:
        x,y = pair
        for v in Delta(g,x,y, epsilon=epsilon).intersection(convexhullA):
            if v not in A and not v['markwp']:
                v['markwp'] = True
                numberofwrongpoints += 1
    return 1 - numberofwrongpoints / len(convexhullA)
```