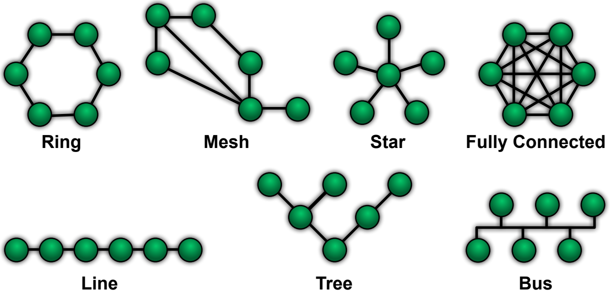
BETTI NUMBERS AND ALPHA COMPLEXES

March 23, 2015

A topological space may be defined as a set of points, along with a set of neighbourhoods for each point, that satisfy a set of axioms relating points and neighbourhoods.



Betti numbers are used to distinguish topological spaces based on the connectivity of n-dimensional simplicial complexes. The k'th Betti number refers to the number of k-dimensional holes on a topological surface.

- b_0 is the number of connected components
- b_1 is the number of one-dimensional or "circular" holes
- b_2 is the number of two-dimensional "voids" or "cavities"

BETTI NUMBERS BY EXAMPLE

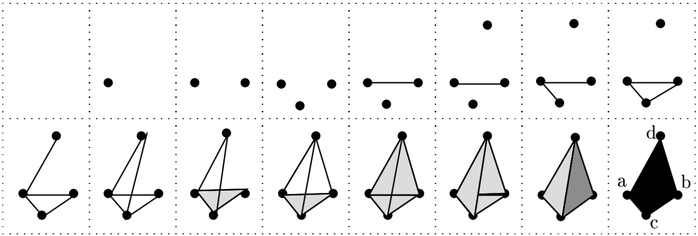
Examples of Betti numbers for a complex.

$\mathcal{K}^{(0)}$	$\mathcal{K}^{(1)}$	$\mathcal{K}^{(2)}$	$\mathcal{K}^{(3)}$
$\beta_0 = 4$	$\beta_0 = 1$	$\beta_0 = 1$	$\beta_0 = 1$
$\beta_1 = 0$	$\beta_1 = 3$	$\beta_1 = 0$	$\beta_1 = 0$
$\beta_2 = 0$	$\beta_2 = 0$	$\beta_2 = 1$	$\beta_2 = 0$

AN ALGORITHM FOR BETTI NUMBERS

Incremental algorithm, that uses filtration.

Filtrations. A *filtration* is a sequence of simplicial complexes, where every complex is a proper subcomplex of its successor. If $\sigma_1, \sigma_2, \ldots, \sigma_n$ is a sequence of simplices, then $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_n$, with $\mathcal{K}_i = \{\sigma_1, \sigma_2, \ldots, \sigma_i\}$, is a filtration, provided each $\mathcal{K}_i$ is a genuine simplicial complex. We call the sequence of simplices the *filter* of the filtration. Figure 2 illustrates an example filtration.



The incremental algorithm:

for $\ell := 0$ **to** d **do** $b_\ell := 0$ **endfor**;
for $i := 0$ **to** m **do**
  $k := \dim \sigma_i$;
  if σ_i belongs to a k-cycle of $\mathcal{K}_i$ **then** $b_k := b_k + 1$ **else** $b_{k-1} := b_{k-1} - 1$ **endif**
endfor.

Lets do a small example.

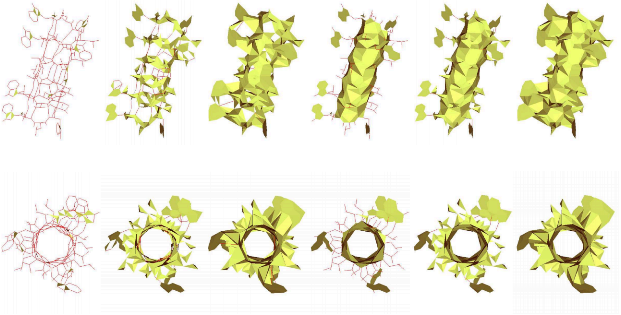
In practice cycles are checked by using sets.
Running time is $O(\alpha(n)n)$, where $\alpha(n)$ is the inverse ackermann function. and n is the sum of k-simplices.

ALPHA COMPLEXES

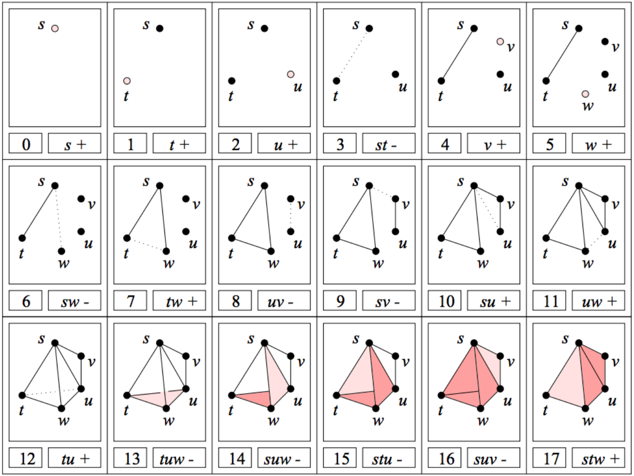
Definition goes here.

PERSISTENCE AND SIMPLIFICATION

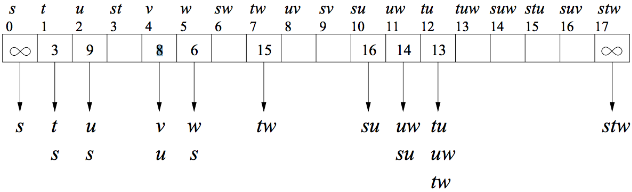
explain



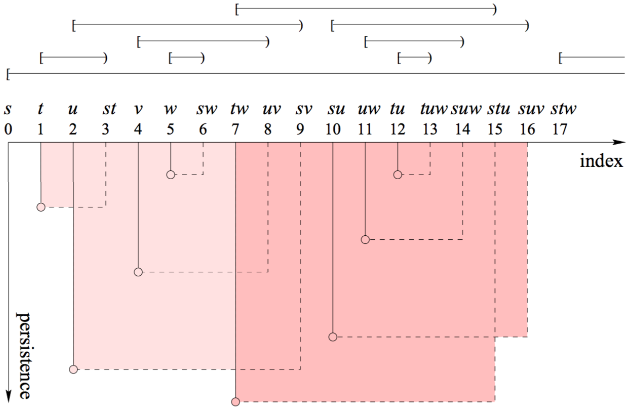
explain


0
s +
1
t +
2
u +
3
st -
4
v +
5
w +
6
sw -
7
tw +
8
uv -
9
sv -
10
su +
11
uw +
12
tu +
13
tuw -
14
suw -
15
stu -
16
suv -
17
stw +

explain


s t u st v w sw tw uv sv su uw tu tuw suw stu suv stw
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
∞ 3 9 8 6 15 16 14 13 ∞
s t u v w tw su uw tu stw
s s u s su uw
tw

explain


s t u st v w sw tw uv sv su uw tu tuw suw stu suv stw
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
index
persistence