BitBlaster: querying metric spaces with bit operations

Alan Dearle & Richard Connor
outline of talk

1. Setting the Scene
2. The bit blaster algorithm
3. Some results
4. Future work
http://mir-flickr-near-duplicates.appspot.com
https://press.liacs.nl/mirflickr/

1000s of features
why similarity search?

• We need to find records that are similar to each other in order to create pedigrees

• We are matching text fields on records, looking for similar pictures, could be molecules, X-rays etc.

• As with other similarity problems we seek to find records that are close to query records

• What does this mean?...
similarity operations

• Range query – find points within range $r$
similarity operations

- Range query – find points within range $r$
similarity operations

- Nearest neighbour query
similarity operations

- K Nearest neighbours query
- K=3
metric spaces

- A metric space is an ordered pair \((M,d)\) where \(M\) is a set and \(d\) is a metric on \(M\), i.e., a function \(d : M \times M \rightarrow \mathbb{R}\), such that for any \(x, y, z\) in \(M\), the following holds:

  1. \(d(x,y) \geq 0\) \hspace{1cm} \textit{non-negativity}
  2. \(d(x,y)=0 \iff x=y\) \hspace{1cm} \textit{identity of indiscernibles}
  3. \(d(x,y) = d(y,x)\) \hspace{1cm} \textit{symmetry}
  4. \(d(x,z) \leq d(x,y) + d(y,z)\) \hspace{1cm} \textit{triangle inequality}
typical approaches: M Tree
failure to narrow search
P1 = duncan*mckinnon|peggy+mckinnon
P2 = niel*mcgillivray|marion+anderson
d = 0.85777833938599
abstract

• Metric search techniques can be usefully characterised by the time at which distance calculations are performed during a query.

• Most exact search mechanisms use a “just-in-time” approach where distances are calculated as part of a navigational strategy.

• An alternative is to use a “one-time” approach, where distances to a fixed set of reference objects are calculated at the start of each query.

• These distances are typically used to re-cast data and queries into a different space where querying is more efficient, allowing an approximate solution to be obtained.

• Here we use a “one-time” approach for an exact search mechanism.

• A fixed set of reference objects is used to define a large set of regions within the original space, and each query is assessed with respect to the definition of these regions.

• Data is then accessed if, and only if, it is useful for the calculation of the query solution.

• As dimensionality increases, the number of defined regions must increase, but the memory required for the exclusion calculation does not.

• We show that the technique gives excellent performance over the SISAP benchmark data sets, and most interestingly we show how increases in dimensionality may be countered by relatively modest increases in the number of reference objects used.
BitBlaster: mechanism

- We characterise metric space as a fixed list of (overlapping, e.g. semispace) regions called exclusion zones (EZs) defined only by reference points and distances

- We store each EZ as list of booleans according to containment, represented as bitmaps

- For each query, for each reference point, we calculate for each EZ one of three possibilities:
  - solution set must be contained in region
  - solution set can’t overlap with region
  - solutions may, or may not, be within region

- From this information calculate regional requirements for solution set as conjunctive normal form logical expression

- Query filtering can then be achieved using logical bitmap comparisons
Data points

<table>
<thead>
<tr>
<th>Data points</th>
<th>EZ1</th>
<th>EZ2</th>
<th>EZ3</th>
<th>EZ4</th>
<th>EZ5</th>
<th>EZ6</th>
<th>EZ7</th>
<th>EZN</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>d6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>d9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>dn</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Exclusion Zones

based on inside or outside of ball and closer to Ro1 than Ro2 in a hyper plane
Query

- 3 Phase query:

1. The distance from the query $q$ to each reference object is measured. We establish “must be” and “can’t be” bitmaps for regions that can contain and certainly do not contain the query - $B_{in}$ and $B_{out}$.

   \[ |d(p_i, q) - \mu| \leq t \]

   For a ball region defined by reference object $p_i$ and a radius $m$, then the condition for intersection is

   \[ \frac{|d(p_i, q)^2 - d(p_j, q)^2|}{2d(p_i, p_j)} < t \]

   For a sheet region, if the the metric has the supermetric property then:

   \[ \frac{|d(p_i, q) - d(p_j, q)|}{2} < t \]

   otherwise.
3 Phase query:

1. The distance from the query q to each reference object is measured. We establish “must be” and “can’t be” bitmaps for regions that can contain and certainly do not contain the query - $B_{in}$ and $B_{out}$.

2. The solutions may be found from

$$\left( \bigwedge_{b \in B_{in}} b \right) \land \left( \neg \bigvee_{b \in B_{out}} b \right)$$

3. Filtering the result sets derived in phase 2 against the original space and distance metric in order to produce an exact solution to the query.
balancing
balancing
balancing
balancing
balancing

- We have tried both balanced and unbalanced bitsets

- Balancing can be achieved by selecting a set of witness objects from the finite space $S$ and finding a median distance or offset for these, so that the regional boundary divides the finite set into two equal parts

- For ball partitions, the median distance to the centre is used; for sheet partitions, an offset can be selected left or right of the central hyperplane

- for supermetric spaces, the XY plane can also be rotated to maximise the spread of values
performance

3 Point Distances per query vs reference objects

4 Point (balanced) Distances per query vs reference objects

4 Point (unbalanced) Distances per query vs reference objects

Colors Thresholds:
- 0.051768
- 0.082514
- 0.131163

NAGA Thresholds:
- 0.12
- 0.265
- 0.53
### Performance

<table>
<thead>
<tr>
<th></th>
<th>colors</th>
<th></th>
<th></th>
<th>nasa</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>t0</td>
<td>t1</td>
<td>t2</td>
<td></td>
<td>t0</td>
<td>t1</td>
<td>t2</td>
</tr>
<tr>
<td>DiSAT metric</td>
<td>4049</td>
<td>9112</td>
<td>19745</td>
<td>554</td>
<td>2176</td>
<td>6448</td>
<td></td>
</tr>
<tr>
<td>DiSAT supermetric</td>
<td>2015</td>
<td>5737</td>
<td>16199</td>
<td>320</td>
<td>1300</td>
<td>5444</td>
<td></td>
</tr>
<tr>
<td>BB: metric unbalanced</td>
<td>4207</td>
<td>14930</td>
<td>42139</td>
<td>14</td>
<td>1120</td>
<td>9202</td>
<td></td>
</tr>
<tr>
<td>BB metric balanced</td>
<td>2246</td>
<td>9610</td>
<td>30250</td>
<td>11</td>
<td>822</td>
<td>7671</td>
<td></td>
</tr>
<tr>
<td>BB supermetric unbalanced</td>
<td>544</td>
<td>3114</td>
<td>13045</td>
<td>3</td>
<td>296</td>
<td>4122</td>
<td></td>
</tr>
<tr>
<td>BB supermetric balanced</td>
<td><strong>518</strong></td>
<td>3259</td>
<td>17512</td>
<td><strong>1</strong></td>
<td><strong>91</strong></td>
<td><strong>2204</strong></td>
<td></td>
</tr>
</tbody>
</table>

Residual distance calculations required when 60 reference objects are used. The numbers reported do not include the 60 distance calculations required for these. The top two rows give comparable figures for the state-of-the-art Distal SAT. Nasa thresholds of $t_0 = 0.12$, $t_1 = 0.285$, and $t_2 = 0.53$ Colors thresholds of $t_0 = 0.052$, $t_1 = 0.083$, $t_2 = 0.13$. 
dealing with dimensionality
tradeoffs

- The computation is $O(n\log n)$ in complexity
  - instead of $O(\log n)$ for a perfect metric index
- However: $n\log n$ bits!
  - so eg $10^{10}$ objects is no problem!
- $O(n\log n)$ part of computation is inherently decomposable and parallelisable
  - and bit computations go really, really fast on modern hardware
- The mechanism scales for dimensionality, if not size
  - increasing dimension adds fixed cost to computation
  - at least, a bit - we’re not sure how far this stretches!
future work

• We are currently investigating:
  • Reference object selection
  • The orthogonality of regions
  • Parallel implementations
  • Amongst others
Time for Euc20 over 1M nodes and 1000 queries 50 ROs T=0.6017
Manifesto E5-2630 6 cores, support for 12 hyperthreads