A New Distance Based Method to Detect Outliers

Submitted By:-

Prayank Mathur
2012JE0719
VI Semester

Nishant Raj
2012JE0847
VI Semester

Dept. Of Computer Science & Engg.

Acknowledgements

It is my great privilege to express my deepest and most sincere thanks to **Dr. Rajendra Pamula** , for his invaluable suggestions and guidance during the course of this project,without which the project would not have been successful.

I am also grateful to my our friends with whom we were able to discuss my doubts.

INTRODUCTION

Continuing our work on the distance based outliers , we suggest a new way to calculate outliers. The method is based on the simple fact that nearby objects from infinity seem pretty close to each other. Also the angle subtended by a close points on infinity is very less. Thus , the two ideas combined can give a set of potential outliers from a given dataset.

Now, given 'n' points with their distances from a very faraway point, all the points that are close to one another and have approximately the same distance from the far point, lie on a circle(in 2-D) , or on a sphere in 3-D .

Now the points that are close to one another subtend equal (or rather with difference very less) angles from that point because it is really faraway. Thus we can screen the points on the above two criteria and choose our oultiers.

ALGORITHM

Algorithm for the above method.

1. Calculate the Distances of all the points (total N) from a fixed predefined point that is far away.(inf = 999999999).

2. Sort the points on the basis of distance calculated in step 1.

3. Group the points into clusters on the basis that their distance (step 1) should not be more than 0.00000001% of the cluster's mean distance.

4. For each cluster formed do the following,
(i). Calculate the angle subtended by each of them from inf.
(ii). Again form a cluster out of the above cluster on the basis that all the points that subtend angles that lie within 0.5 degree of the new cluster's average mean should lie in that group else a new group should be formed for them
(iii). If the number of points in a particular group are less than 0.5% of the total points N then declare all the points in the group as outliers.

COMPLEXITY

The complexity for the above algorithm is O(nlogn) in average case and O(nlogn) in the worst case.

RESULTS

The results of the above algorithm on standard dataset revealed quite good results and with an efficiency of more than 95%.
On a dataset having 30 dimensions the results were following.

Cluster 1 -> 333 332 35 356 68 166 14 169 351

Cluster 2 -> 4 197 20 48 347 220 307 69 247 240 94 16 15 84 237 273 349 18 46 322 159 318 308 13 85 153 33 12 350 42 98 342 37 45 355 121 2 208 281 184 187 279 182 248 133 130 103 38 114 127 143 321 106 113 171 261 344 134 341 155 124 353 345 331 343 198 282 340 178 270 330 83 59 242 231 284 112 249 157 119 259 89 250 109 310 82 76

Cluster 3 -> 58 218 144 193 266 24 262 280 319 62 7 219 265 17 334 213 285 156 60 126 39 327 315 236 56 185 151 141 88 251 43 32 243 212 191 239 211 289 10 102 41 138 235 245 162 57 77 194 154 170 295 63 135 70 91 253 27 125 275 186 75 346 309 190 297 179 79 221 132 312 36 148 290 230 328 90 226 47 195 192 216 9 176 101 224 180 188 225 306 140 234 72 40 92 300 199 34 201 214 54 165 298 26 252 139 168 3 96 174 19 173 326 5 167 74 81 129 49 149 147 229 21 87 172 303 204 241 95 304 118 164 181 227 286 324 142 268

Cluster 4 -> 93 348 233 67 145 337 269 272 128 255 200 354 111 11 271 99 317 301 316 120 325 228 177 44 116 23 25 86 209 278 22 122 97 6 123 277 338 223 320 339 1 61 244 31 115 274 352 51 53 215 161 246 137 257 71 335 293 8 64 117 158 196 29 107 28 105 260 256 206 336 276 323 222 203 210 0 292 287 258 150 136 302 66 52 146 65 267 238 288 217 104 160 80 50 264 131 254 305 78 110 175 108 291 313 183 152 296 314 263 329 30 163 189 205

Cluster 5 -> 294 283
Cluster 6 -> 73 207 55 202 232 311 100

Cluster 7 -> 299
Cluster 8 -> 357
Cluster 9 -> 359
Cluster 10 -> 358
Cluster 11 -> 360
Cluster 12 -> 361