Crowd Screen

Algorithms for filtering data with humans

Can humans filter information better?

How did we perform filtering?

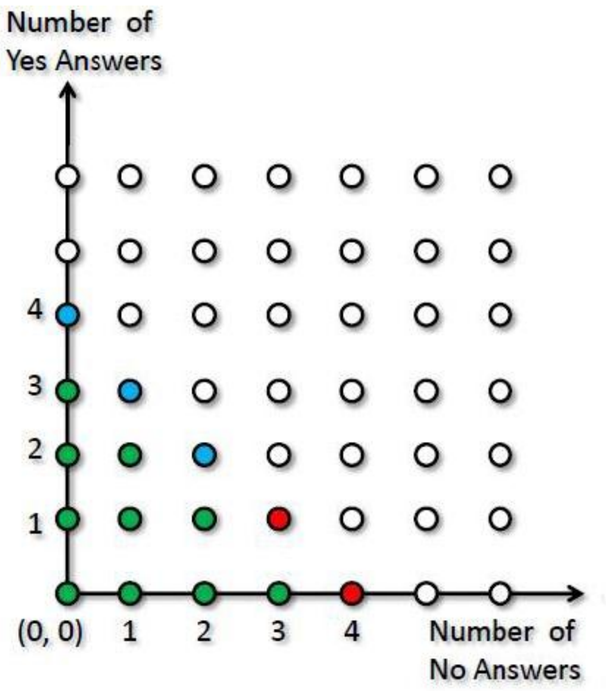
- Decided a question
- Repeat the question to a set of audience
- Combine the result
- How about the overall error?
- How about the overall cost?
- Is there a way to optimize(overall error and overall cost)

Preliminaries

- Several applications have come up with many filtering approaches
- The focus of the paper is to study how to implement optimal filtering strategies
- We consider predominantly single filter cases and determine how to find optimal strategy
- Also we assume our filters are binary(ie. They simply return YES or NO)

Strategy As Grid

- Strategy can be represented as 2D grid



Strategy As Grid

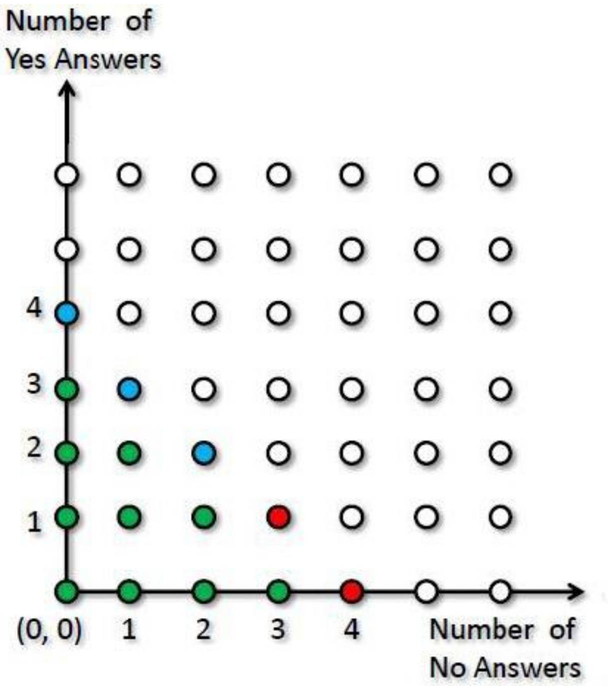
- Y axis represents number of YES answers
- X axis represents number of NO answers
- Blue grid point represents strategy output “Pass” at this point
- Red grid point represents strategy output “Fail” at this point
- Green points represents continue points(ie: no decision is made and we continue to ask questions)

Strategy

- A strategy is a computer algorithm that takes one item as input and asks one or more humans questions on the same item and outputs either “Pass” or “Fail”
- A “Pass” output represents item satisfies the filter
- A “Fail” output represents item not satisfying the filter
- Strategy can be visualized by a 2-D grid

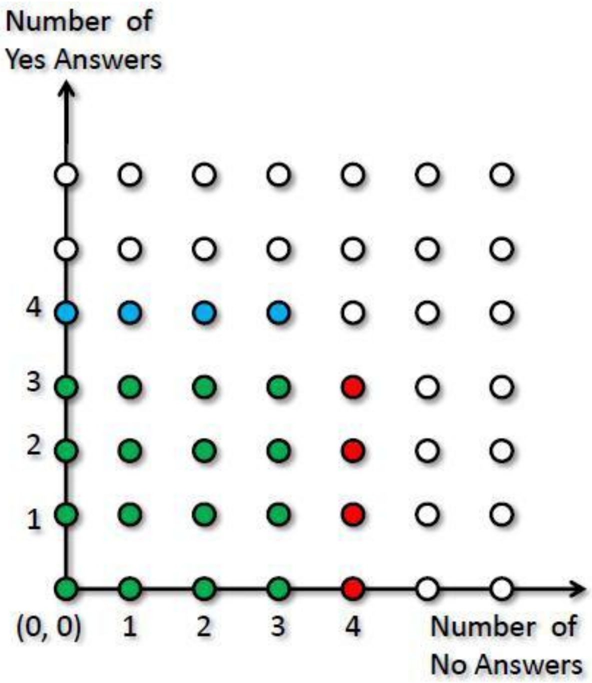
Common Strategies

- Triangular strategy
- Always ask constant number of people
- Eg: 4 persons for item



Common Strategies

- Rectangular Strategy
- The process stops after receiving constant number of YES or NO



Properties of Strategy

- Strategy is nothing but processing sequence of “YES” or “NO” answers
- Strategy tells us what to do at each reachable point
- We want our strategies to be terminating
- Batch of questions can be asked. In that case if decision is reached before receiving all answers, the outstanding questions can be cancelled
- The triangular and rectangular strategies are deterministic
- In deterministic strategy output is same for the same sequence of answers

Formal Definitions

- Set of items D, where |D|=n
- Random variable V (V=1 item satisfies filter , V=0 item does not satisfies filter) and selectivity of the filter ‘s’
- False positive rate pr[answer is YES| V=0]=e0
- False negative rate pr[answer is NO| V=1]=e1

Error

- $E(x,y) = p0(x,y)/[p0(x,y)+p1(x,y)]$ if $Pass(x,y)$
- $E(x,y) = p1(x,y)/[p0(x,y)+p1(x,y)]$ if $Fail(x,y)$
- $E(x,y) = 0$ else

Error across all termination point is given by

- $E = \sum(x,y)E(x,y) \times [p0(x,y)+p1(x,y)]$

P0 and P1

- To determine the best strategy we need error and cost
- p1(x , y) - probability that strategy reaches point(x , y) and the item satisfies the filter(V=1)
- P0(x , y) - probability that strategy reaches point(x , y) and the item does not satisfy the filter(V=0)

$$p_0(x,y) = \begin{cases} p_0(x-1,y)(1-e_0) + p_0(x,y-1)e_0 & \text{if } \neg\text{Term}(x,y-1) \wedge \neg\text{Term}(x-1,y) \\ p_0(x,y-1)e_0 & \text{if } \neg\text{Term}(x,y-1) \wedge \text{Term}(x-1,y) \\ p_0(x-1,y)(1-e_0) & \text{if Term}(x,y-1) \wedge \neg\text{Term}(x-1,y) \\ 0 & \text{if Term}(x,y-1) \wedge \text{Term}(x-1,y) \end{cases}$$

$$p_1(x,y) = \begin{cases} p_1(x,y-1)(1-e_1) + p_1(x-1,y)e_1 & \text{if } \neg\text{Term}(x,y-1) \wedge \neg\text{Term}(x-1,y) \\ p_1(x,y-1)(1-e_1) & \text{if } \neg\text{Term}(x,y-1) \wedge \text{Term}(x-1,y) \\ p_1(x-1,y)e_1 & \text{if Term}(x,y-1) \wedge \neg\text{Term}(x-1,y) \\ 0 & \text{if Term}(x,y-1) \wedge \text{Term}(x-1,y) \end{cases}$$

Cost

- C(x,y) is the number of questions used to reach decision at (x,y)
- We consider cost to be zero at all non termination points
- The expected cost across all termination point is given by

$C=\sum_{(x,y)}C(x,y) \times [p0(x,y)+p1(x,y)]$

- The cost for evaluating n items is given as nC

Problems

- Problem 1: Given an error threshold τ and a budget threshold per item m, find a strategy that minimizes C under the constraint $E < \tau$ and $\forall (x, y)\ C(x, y) < m$
- Problem 2: Given an error threshold τ and a budget threshold per item m, find a strategy that minimizes C under the constraint $\forall (x, y)\ E(x, y) < \tau$ and $C(x, y) < m$
- We will be focusing on problem 1 to explain our strategies

Brute force

- **Theorem** : The expected cost and error of a strategy can be computed in time proportional to the number of reachable grid points
- Brute force algorithm to find the best deterministic strategy involves examining strategies corresponding to all possible assignments of “Pass”, “Fail” or “Continue” points
- Naive3 finds the best strategy in $O(3^m m^2)$

Path Principle

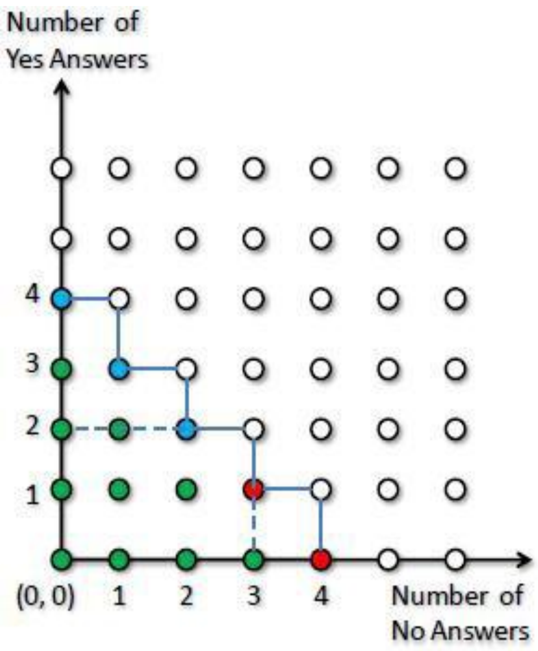
- **Theorem:** Given s,e1,e0 for every point(x , y), the function R(x , y) = p0(x , y)/(p0(x , y)+p1(x , y)) is a function of (x , y) , independent of the particular (deter or prob) strategy
- Intuitively the theorem holds because the strategy only changes the number of paths leading to the point but the characteristic of the point stay the same
- **Theorem :** For every optimal strategy ,for every point(x , y) ,if Term(x , y) holds then

  If R(x , y) > ½,then fail(x , y)

  If R(x,y) < ½ ,then pass(x,y)

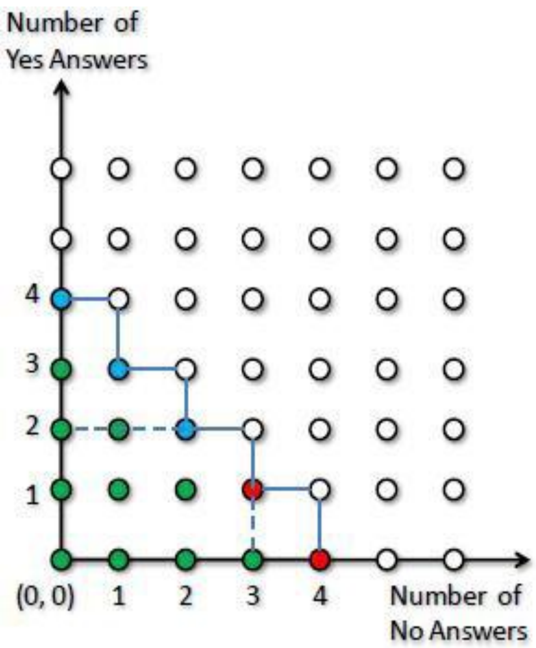
Naive2: The best strategy for problem 1 using naïve 2 can be found in $O(2^m m^2)$

Shape



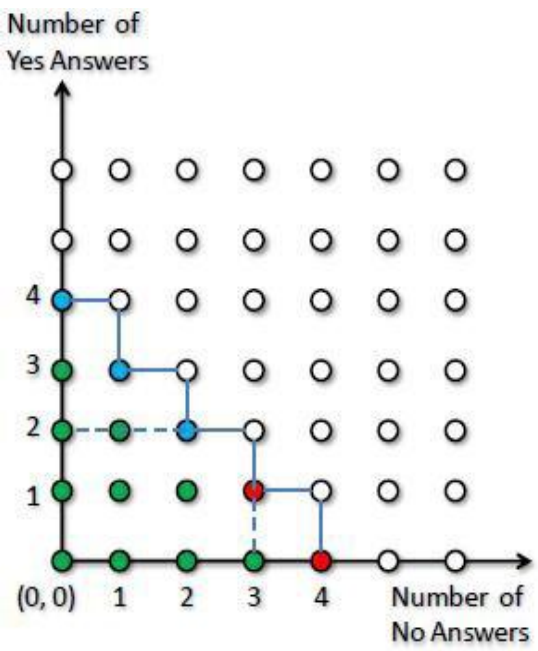
- In Practice considering all 2^m strategies is feasible only for small m
- A shape is defined by connected sequence of segments on the grid beginning at a point on y-axis and ending at a point on x-axis along with a special point called decision point

Ladder



- From all strategies that correspond to shapes , if we wanted to find best strategy we only need to consider the subset of shapes that we call ladder shapes
- Ladder is a connected sequence of segments connecting grid points such that flat lines go right(ie from a smaller x value to a larger x value) and vertical lines go up(ie from smaller y value to larger y value)

Converting shapes to ladder



- From the definition ,the shape is not ladder
- In the strategy note that asking questions at point above y=2 ,we always reach "Pass"(redundant questions)
- Similarly asking questions on right of line x=3 is redundant

Probabilistic Strategies

- Each point is represented in a grid with triple(r1,r2,r3) corresponding to the probability of returning continue, pass or fail
- The probabilistic strategy is also called ‘linear’ program

$$E = \sum_{(x,y);x+y\leq m} tPath(x,y) \times \min(S_0(x,y), S_1(x,y))$$

$$C = \sum_{(x,y);x+y\leq m} tPath(x,y)(x+y)(S_0(x,y) + S_1(x,y))$$

Growth

- Growth : The greedy algorithm “grows” a strategy until the constraints are met
- It begins with null strategy (0,0) and each iteration ,the algorithm “pushes the boundary ahead) (ie .. (x+1,y) or (x,y+1))
- The ratio of change in cost to the change in error is computed
- The algorithm moves the termination point towards the smallest increase in ratio
- The pushing continues until the error constraint is satisfied

Shrink and point

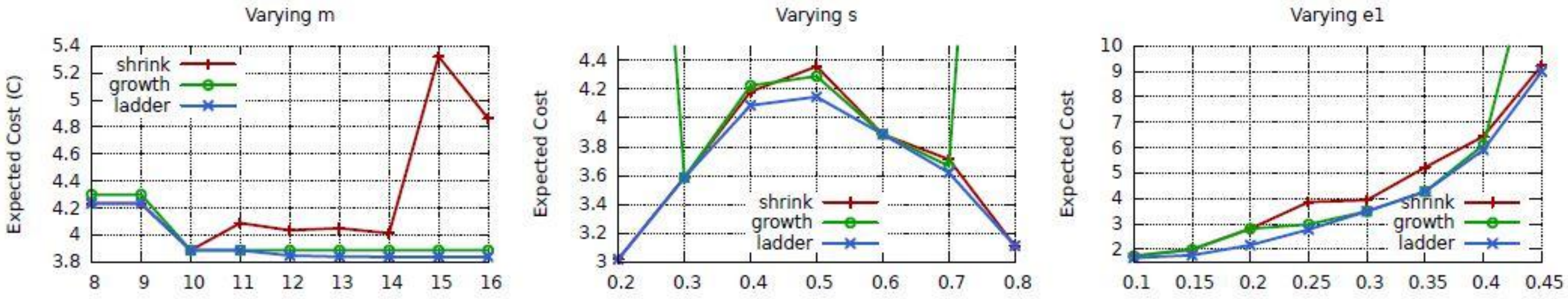
- Works opposite in compare to “growth”
- It begins with m questions and continue to choose between (x,y-1) or (x-1,y) based on the ratio of change in cost to change in error
- It stops to shrink when the error constraint is satisfied
- Point algorithm: The algorithm ensures that at every termination point , $E(x,y) < \tau$

Experiments

- The parameters chosen are m,e0,e1,τ,s
- In some cases ,the values are manually selected
- In other cases values are selected over a range to explore average behavior
- Experiments on the following deterministic and probabilistic algorithms
- naive3,naive2,ladder,growth,shrink,rect,linear,point

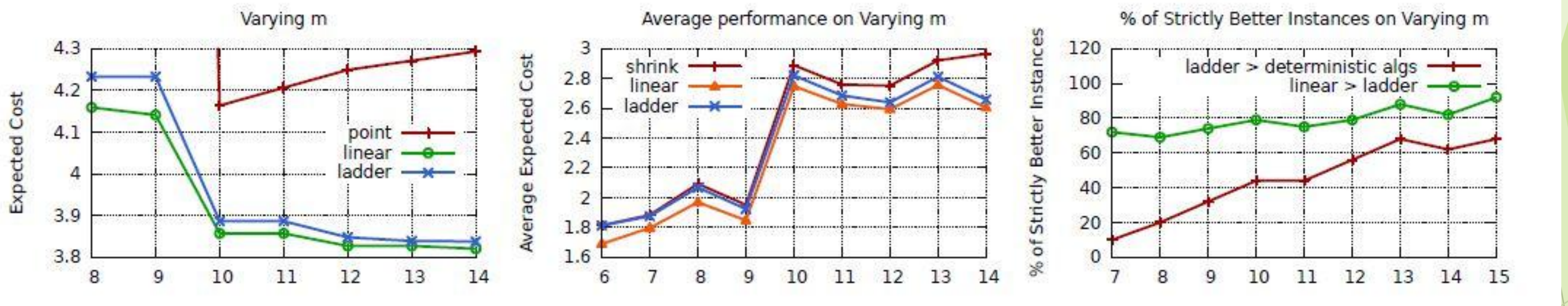
Compare-Deterministic algo

- The parameters s , e0 , e1 and τ are kept constant and m is varied
- The parameter e0, e1, τ, m are kept constant and s is varied
- The parameters e0,s, τ,m are kept constant and e1 is varied

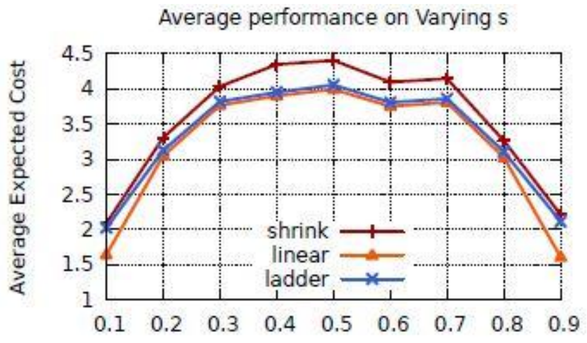


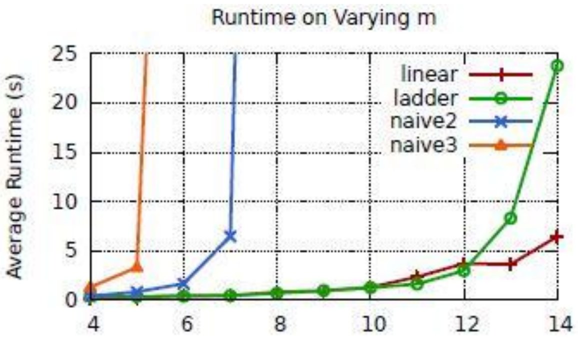
Ladder vs point vs linear

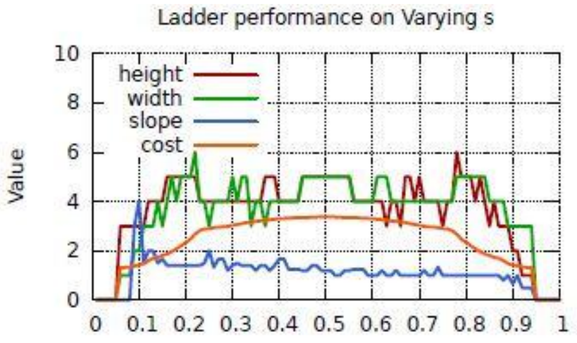
- Linear yields better than ladder in majority of scenarios and ladder outperforms rest of the deterministic algorithms in a substantial number o scenarios



Other experiments







Related work

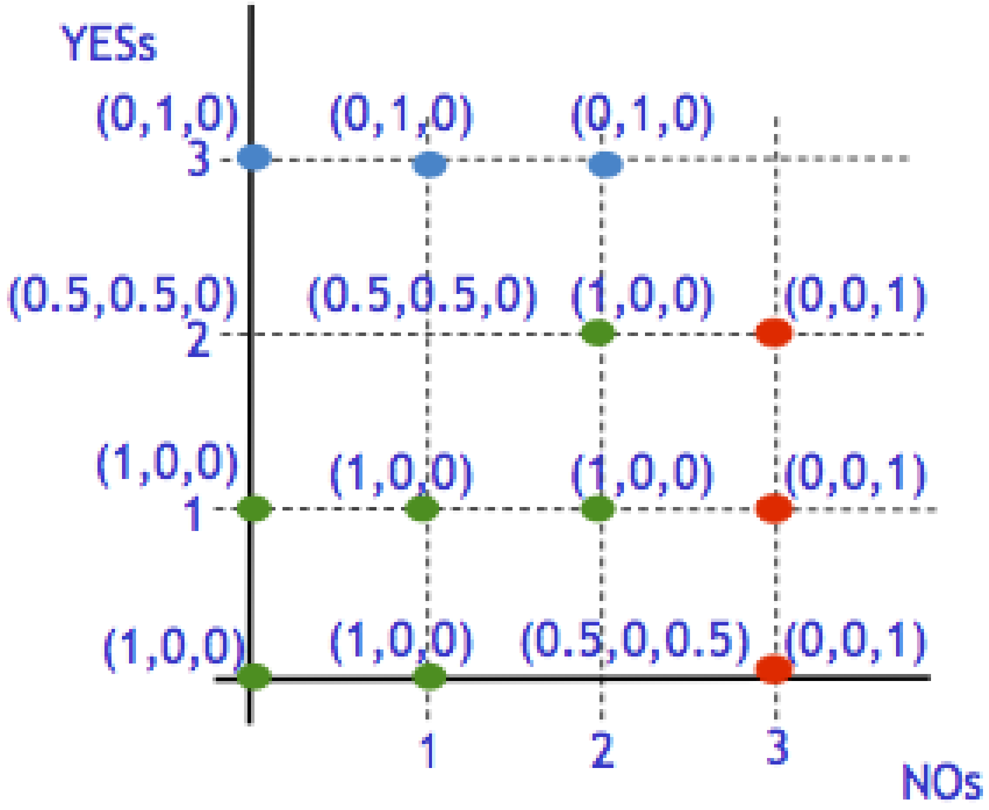
- Machines schema in online communities – considers the problem of using crowd sourcing for schema matching at least v1 questions and stop either when the number of “Yes” and “No” answers reach a threshold ∂
- Active Learning Literature surveys - actively selecting the training data set to ask an “oracle” which would help in training the classifier with least error
- The other works such as “All of statistics” for hypothesis testing and “cheap and fast – but is good” for filtering applications are some of good works in the respective fields

Future work

- Handling correlation between filters in the multiple-filters case
- Extending the approach for categorization and classification problems
- Resolving the open question of whether shapes are optimal for deterministic strategies

Thank you

Grid points



YESs
(0,1,0)
(0,1,0)
(0,1,0)
3
(0.5,0.5,0)
(0.5,0.5,0)
(1,0,0)
(0,0,1)
2
(1,0,0)
(1,0,0)
(1,0,0)
(0,0,1)
1
(1,0,0)
(1,0,0)
(0.5,0,0.5)
(0,0,1)
1
2
3
NOs