Anything You Can Do, I Can Do Better: Finding Expert Teams by CrewScout*

Naeemul Hassan [1], Huadong Feng [1], Ramesh Venkataraman [1]
Gautam Das [1], Chengkai Li [1], Nan Zhang [2]
[1]University of Texas at Arlington, [2]George Washington University

ABSTRACT

CrewScout is an expert-team finding system based on the concept of *skyline teams* and efficient algorithms for finding such teams. Given a set of experts, CrewScout finds all k-expert skyline teams, which are not dominated by any other k-expert teams. The dominance between teams is governed by comparing their aggregated expertise vectors. The need for finding expert teams prevails in applications such as question answering, crowdsourcing, panel selection, and project team formation. The new contributions of this paper include an end-to-end system with an interactive user interface that assists users in choosing teams and an demonstration of its application domains.

Categories and Subject Descriptors

H.2 [**DATABASE MANAGEMENT**]: Database applications

Keywords

Skyline Queries; Team Recommendation

1. OVERVIEW

We introduce CrewScout (http://idir.uta.edu/crewscout), a system for finding expert teams in accomplishing tasks. The underpinning concept of the system is *skyline teams* (called *skyline groups* in [2, 3]). The new contributions made in this paper include an end-to-end system with an interactive user interface that assists users in choosing teams among potentially many skyline teams and an extension of application and demonstration scenarios into more general areas ([2, 3] mostly focused on the application of forming teams in fantasy sports games.)

Consider a set D of n experts $t_1, \ldots, t_n$, modeled by m numeric attributes $A_1, \ldots, A_m$ that represent their skills and expertise. Any subset of k experts form a *k-expert team*. CrewScout finds, for a given k, all k-expert skyline teams, i.e., k-expert teams that are not *dominated* by any other k-expert teams. It further assists users in choosing among the skyline teams. The notion of dominance between teams is analogous to the dominance relation between tuples in skyline analysis [1]. CrewScout calculates for each team an aggregate vector of its experts' individual vectors. CrewScout provides efficient algorithms for four commonly used aggregate functions—AVG (i.e, SUM, since we only compare teams with equal size), MIN and MAX. [1] A team G_1 *dominates* another team G_2 (denoted $G_1 \succ G_2$), if and only if the aggregate value of G_1 on every attribute is better than or equal to the corresponding value of G_2 and G_1 has better value on at least one attribute.

	database	indexing
t_1	3	0
t_2	0	3
t_3	2	1
t_4	2	2
t_5	0	2

Table 1: Experts

team	AVG	MIN	MAX
$G_{1,2}$	$\langle 1.5, 1.5\rangle$	$\langle 0, 0\rangle$	$\langle$ **3, 3**$\rangle$
$G_{1,3}$	$\langle 2.5, 0.5\rangle$	$\langle 2, 0\rangle$	$\langle 3, 1\rangle$
$G_{1,4}$	$\langle$ **2.5, 1.0**$\rangle$	$\langle 2, 0\rangle$	$\langle 3, 2\rangle$
$G_{1,5}$	$\langle 1.5, 1.0\rangle$	$\langle 0, 0\rangle$	$\langle 3, 2\rangle$
$G_{2,3}$	$\langle 1.0, 2.0\rangle$	$\langle 0, 1\rangle$	$\langle 2, 3\rangle$
$G_{2,4}$	$\langle$ **1.0, 2.5**$\rangle$	$\langle$ **0, 2**$\rangle$	$\langle 2, 3\rangle$
$G_{2,5}$	$\langle 0, 2.5\rangle$	$\langle$ **0, 2**$\rangle$	$\langle 0, 3\rangle$
$G_{3,4}$	$\langle$ **2.0, 1.5**$\rangle$	$\langle$ **2, 1**$\rangle$	$\langle 2, 2\rangle$
$G_{3,5}$	$\langle 1.0, 1.5\rangle$	$\langle 0, 1\rangle$	$\langle 2, 2\rangle$
$G_{4,5}$	$\langle 1.0, 2.0\rangle$	$\langle$ **0, 2**$\rangle$	$\langle 2, 2\rangle$

Table 2: All possible 2-expert teams

The need for finding expert teams prevails in several application areas, including question answering, crowdsourcing, panel selection, project team formation, and so on. This is illustrated by the following motivating examples.

CrowdSourcing Consider forming a team of Wikipedia editors to write a new Wikipedia article related to "database" and "indexing". Table 1 shows all relevant editors $t_1, \ldots, t_5$ and their expertise on the two topics. We want to assign the task to a team of 2 editors. Table 2 shows the aggregate vectors under AVG, MIN and MAX, for all possible 2-expert teams where $G_{i,j}$ stands for a team of experts t_i and t_j. A simple scheme such as picking top editors on individual topics does not work. For example, $G_{1,2}$ consists of the top editor on each topic and has an aggregated vector $\langle 1.5, 1.5\rangle$ with regard to AVG. $G_{3,4}$, with vector $\langle 2.0, 1.5\rangle$, dominates $G_{1,2}$ (denoted $G_{3,4} \succ G_{1,2}$) under AVG. Hence, $G_{3,4}$ is a better team in terms of collective expertise. In fact, $G_{3,4}$ is a 2-expert skyline team, since no other team dominates it under AVG. Table 2 highlights all 2-expert skyline teams for every aggregate function.

Questing Answering Consider a question-answering platform such as Quora.com. A question is displayed to users who might answer it. The question asker can also explicitly solicit answers from certain users, oftentimes by offering rewards. To receive quality answers, it is necessary to intelligently post the question to users with proper expertise. More often than not, a question requires expertise on several aspects that cannot be fulfilled by any single user, needing attention from a diverse team of experts who collectively

*Das is supported in part by NSF grant 1018865 and grants from Microsoft Research. Hassan and Li are partially supported by NSF grants 1018865, 1117369, 1408928, and the National Natural Science Foundation of China Grant 61370-019. Zhang is supported in part by NSF grants 0852674, 0915834, 1117297, and 1343976. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agencies. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notice herein.

CIKM'14, November 3–7, 2014, Shanghai, China.
ACM 978-1-4503-2598-1/14/11.
http://dx.doi.org/10.1145/2661829.2661839.

[1]While the concept allows arbitrary aggregate functions, efficient algorithms for less common functions remain an open problem.

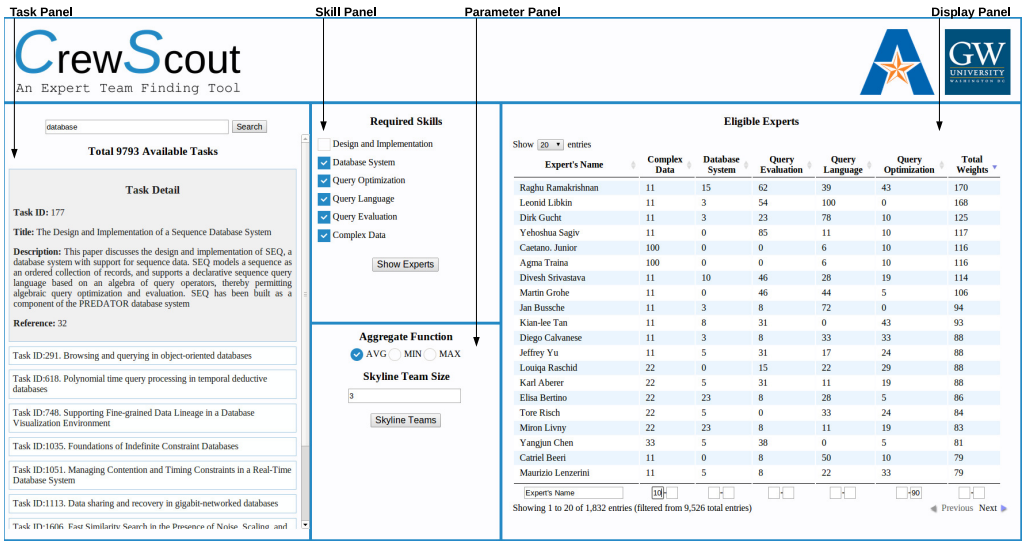


Figure 1: CrewScout **Interface**

excel. For instance, consider question "Is C or Python better for high-performance computing?" To get a comprehensive answer, we need experts in "high performance computing", "C", and so on.

Other Motivating Applications The need for finding expert teams arises in several other applications. **1)** Consider the task of choosing a panel of experts to evaluate a research paper or a grant proposal. An expert can be modeled as a tuple in the multi-dimensional space defined by the paper's topics, to reflect the expert's strength on these topics. The collective expertise of a panel is modeled as the aggregate vector of the corresponding tuples. **2)** Forming collaborative teams for a software development project can be viewed as finding programmers who are collectively strong in the multi-dimensional space of desired skills for the project. **3)** In a variety of applications we look for "teams" in more general sense, such as bundles of products, reviews, stocks, and so on. For instance, to summarize a product's many customer reviews, choosing a set of diverse reviews is forming a "team" of reviews, where the reviews are modeled by attributes such as "sentiment", "length", "quality", etc. Another example is online fantasy sports where gamers compete by forming and managing team rosters of real-world athletes, aiming at outperforming other gamers' teams. The teams are compared by aggregated statistics (e.g., "points", "rebounds", "assists" in basketball games) of the athletes in real games.

An attractive characteristic of a skyline team is that no other team of equal size can dominate it. In contrast, given a non-skyline team, there is always a better skyline team. This property distinguishes CrewScout from other team recommendation techniques. The skyline teams consist of the teams that are worth recommending. They become the input to further manual or automated post-processing that eventually finds one team. Admittedly, determining the "best" team is a complex task that may involve more factors than what skyline teams can capture—e.g., which experts are available for a task, whether they have good relationship to work together, and so on. The post-processing is thus crucial. Examples of such post-processing include eye-balling the skyline teams, filtering and ranking them by user preferences, and browsing and visualization of the skyline teams. Particularly, CrewScout provides an interactive tool to assist a human user in exploring and choosing skyline teams.

2. USER INTERFACE

Figure 1 shows the GUI of CrewScout, which is comprised of a *task panel*, a *skill panel*, a *parameter panel*, and a *display panel*.

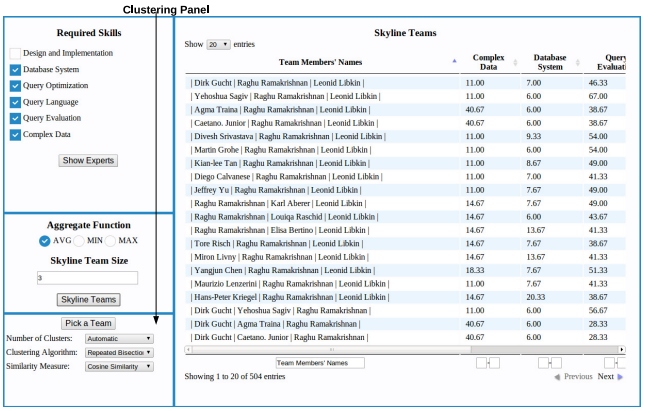
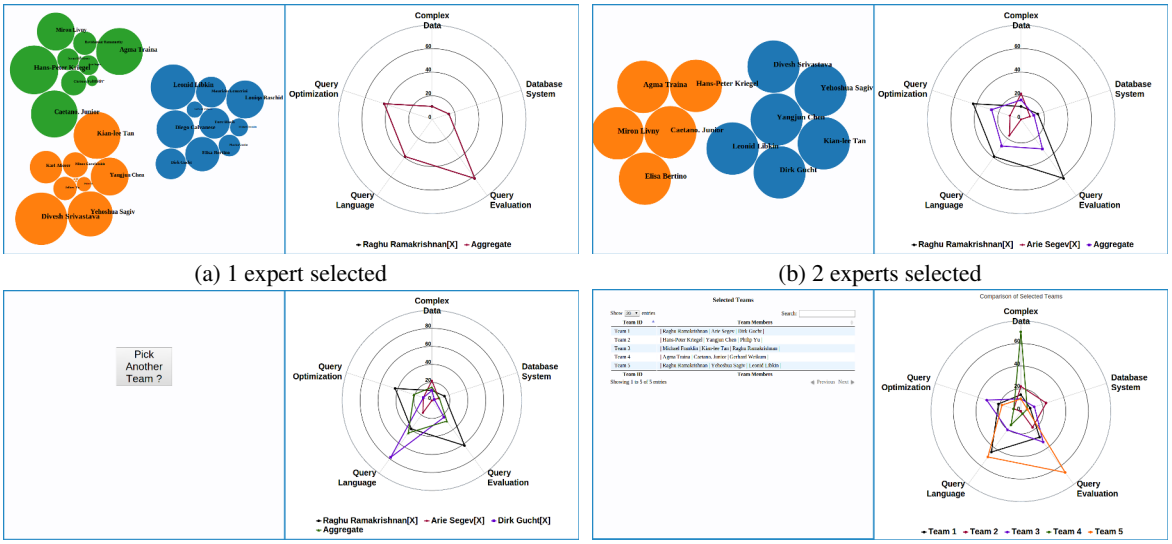


Figure 2: Display Panel Showing Skyline Teams

The task panel presents a list of available tasks. When a user clicks a task, CrewScout provides more details about it. CrewScout also provides a keyword search box at the top of this panel for searching available tasks. The skill panel presents the skills required for completing the selected task. It shows a checkbox for each skill. By default, all the checkboxes are checked. The user can check/uncheck some of them according to their preference. When the user clicks the "Show Experts" button, the display panel presents a paginated list of all experts who have expertise in at least one checked skill. If the user further checks/unchecks some skills, the expert list is automatically refreshed to reflect the change. Experts are ordered by summations of their expertise in all selected skills in the current implementation. In the expert list, a filter is provided for each skill. The user can filter the experts by setting the minimum and maximum expertise for one or more skills. The user can also filter the experts by their names through partial string matching.

The parameter panel allows the user to set parameters for skyline team computation. It includes a textbox for specifying the skyline team size and radio buttons for choosing an aggregate function (AVG, MIN, or MAX). Once the user clicks the "Skyline Teams" button, CrewScout calculates all skyline teams (considering all experts satisfying the aforementioned filters) and shows them in the display panel (Figure 2). Similar to the filters on experts, CrewScout also provides filters for the skyline team list, including filters on team members' names and minimum/maximum aggregated exper-



(a) 1 expert selected

(b) 2 experts selected

(c) 3 experts selected

(d) Selected skyline teams

Figure 3: Selecting and Comparing Teams

tise on individual skills. The teams satisfying the conditions are called the *qualifying skyline teams*. When the display panel exhibits the skyline teams, a *clustering panel* is added below the parameter panel. It provides a "Pick a Team" button and three drop-down lists that allow the user to choose a clustering algorithm (e.g., *K-means*), a similarity/distance function (e.g., Euclidean distance) for the clustering algorithm, and the number of clusters. When the user clicks the button, CrewScout will display below the current panels a visualization interface (Figure 3) that clusters the experts in the qualifying skyline teams and assists the user in exploring and choosing teams.

The visualization interface has two panels. The left panel visualizes the clusters. Each expert that belongs to at least one qualifying skyline team is represented as a circle. Circles in the same cluster are annotated with the same color. Their positions are automatically determined by the *multi-foci force layout* (https://github.com/mbostock/d3/wiki/Force-Layout). The size of a circle is proportional to the number of qualifying skyline teams containing the corresponding expert. At the beginning, only the labels of big circles (containing information of corresponding experts) are visible. When the user hovers the mouse over a circle, the expert's profile (including the name and the number of skyline teams containing the expert) is displayed in a small pop-up window. The user can gradually zoom in to see the labels of smaller circles. The user can iteratively select k experts. Whenever the user selects an expert, CrewScout removes those circles whose corresponding experts do not belong to any skyline teams with all selected experts so far. The remaining circles are re-clustered and resized, based on only qualifying skyline teams containing the selected experts. The right panel presents a polar-chart. Each polygon in the polar-chart represents a selected expert's expertise on the chosen skills. The aggregated expertise of the selected experts is also represented by a polygon. The selected experts are listed under the polar-chart. The user can remove any expert by clicking the cross sign beside it, and the clusters of circles are refreshed accordingly. Once k experts are selected, a skyline team is chosen. A "Pick Another Team?" button appears in the left panel. If the user clicks it, two more panels are added to the lower portion of the visualization interface—the left one lists all selected teams and the right one presents another polar-chart that compares them.

3. DEMONSTRATION PLAN

An online demonstration of CrewScout is hosted at http://idir.uta.edu/crewscout. Its front-end UI is developed in PHP+JavaScript. The system demonstrates three application scenarios, including paper reviewer selection, question answering, and team formation, on a 900K-publication dataset collected through Microsoft Academic Search API, a stackoverflow.com dataset and an NBA dataset from databasebasketball.com, respectively. It also supports user-uploaded datasets. Below we describe the demonstration steps for the reviewer selection scenario, with an imaginary user Amy.

(1) Amy searches for, say "database", and matching publications are displayed in the task panel. A default publication is highlighted.
(2) Amy clicks a publication to show or hide its abstract, depending on its status. When a publication is selected, the skill and display panels are refreshed with the corresponding required expertise and qualifying reviewers, respectively. Amy checks/unchecks one or more skills, the qualifying reviewers are automatically refreshed. Amy filters the reviewers by setting minimum and/or maximum thresholds on one or more skills. (Figure 1)
(3) Amy specifies an aggregate function and a skyline team size in the parameter panel. Once Amy clicks the "Skyline Teams" button, the display panel shows the skyline teams (Figure 2). Amy can filter them by reviewer name and thresholds on aggregated skills.
(4) After choosing clustering parameters, Amy clicks the "Pick a Team" button and the visualization interface presents the reviewer clusters (Figure 3).
(5) Amy moves the mouse over the circles to see the reviewers' profiles and she also zooms in and out. When Amy selects a reviewer, the corresponding expertise polygon is inserted into the polar-chart. Amy repeats this step multiple times until a k-reviewer team is formed.
(6) Amy clicks the "Pick Another Team?" button to select another team. In this way, Amy chooses multiple teams and compares them in a polar-chart.

4. REFERENCES

[1] S. Borzsony, D. Kossmann, and K. Stocker. The skyline operator. In *ICDE*, pages 421–430, 2001.
[2] C. Li, N. Zhang, N. Hassan, S. Rajasekaran, and G. Das. On skyline groups. In *CIKM*, pages 2119–2123, 2012.
[3] N. Zhang, C. Li, N. Hassan, S. Rajasekaran, and G. Das. On skyline groups. *TKDE*, 26(4):942–956, April 2014.