ECCO-A Framework for Ecological Data Collection and Management Involving Human Workers

Senjuti Basu Roy, Sihem Amer-Yahia, Lucas Joppa

ECCO- A Framework for Ecological Data Collection and Management Involving Human Workers

Senjuti Basu Roy[†], Sihem Amer-Yahia[⋄], Lucas Joppa[††].
[†]UW Tacoma, [⋄] CNRS, LIG, [††] Microsoft Research
senjutib@uw.edu,sihem.amer-yahia@imag.fr,lujoppa@microsoft.com

ABSTRACT

Scientific and ecological data collection in today's world is primarily driven by citizen-based observation networks to gather information on a diverse array of species and natural processes. Such efforts leverage the contributions of a broad recruitment of human observers to collect data and use Machine Learning algorithms to process the collected data leading to a computational power that far exceeds the sum of the individual parts. Instead of organic group formation and collaboration, our vision is the need to formalize collaboration and rethink the components of a data management system to ensure its sustainability in such human-intensive applications. The enabler of collaboration is the notion of *a user group* that implies different behaviors and interactions between its members. We advocate the design of new components of a data management system that deliberately acknowledge the uncertainty and dynamicity of *human behavior* by capturing the *human factors* that characterize group members. We describe ECCO, a framework that contains two generic components: *adaptive collaborative human factors learning* and *adaptive human-centric optimization*. Those are the core components that support the fundamental functionalities of a wide range of human-intensive applications. ECCO components rely on two optimization engines, namely *task assignment* and *human data management engine*. An additional challenge in designing the components of ECCO is the need to support adaptive and incremental computation. We discuss the modeling, learning, and computational challenges of designing the components of ECCO and propose a roadmap of future directions of this vision.

1. INTRODUCTION

Achieving insight about ecological patterns often requires the study of natural systems at large scales. An emerging focus therefore is to build an infrastructure for data synthesis and analysis that allows data collection and organization across the continent and perform large scale analyses over it. While new technologies are gradually emerging to leverage autonomous sensor networks for such data collection, the state of the art techniques still can not identify organisms to species, they serve to gather information on the variables that influence species occurrence. Therefore, most data on species-level occurrence still must be gathered by humans [10], necessitating innovative programs for wide-scale data collection and analysis. In particular, the ultimate objective of such effort is to build a hybrid human machine computational power to solve complex problems. We advocate the design of a new framework ECCO to that end with the vision to formalize collaboration and rethink the components of a data management system to ensure quality and sustainability in such human-intensive applications.

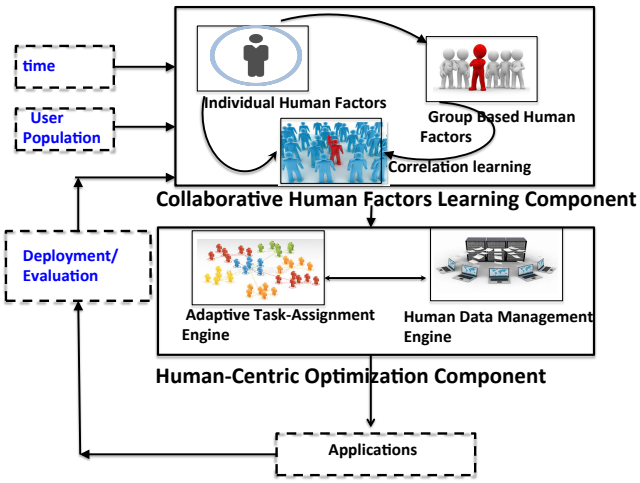

Figure 1: High Level Design of ECCO

Applications: Several leading efforts of citizen science are being carried out nationally and internationally. For example, the US National Phenology Network[1] conducts Project Budburst, a citizen-based effort to report phenological events such as first leafing, first flowering, and first fruit ripening for a variety of plant species in order to better understand the broad scale effects of climate change. The Galaxy Zoo [2] project provides access to almost 250,000 images of galaxies and engages volunteers to classify them into shapes in order to better understand how galaxies are formed. In FoldIt [3] project, researchers attempt to predict

[1]http://www.usanpn.org/
[2]http://www.galaxyzoo.org/
[3]http://fold.it/portal/

10.5441/002/edbt.2015.68

the structure of a protein by taking advantage of puzzle solving abilities of the human. Another popular example is the e-Bird project [11, 4], which engages a vast network of human observers (citizen-scientists) to report bird observations using standardized protocols.

Objective: The ultimate objective of such efforts is to employ a hybrid human and mechanical computation power to solve complex problems through active learning and feedback. The contributed large scale data is processed with Machine Learning algorithms for correlating species distributions with environmental covariates to identify the unlabeled points that when labeled would most rapidly decrease uncertainty in the model being deployed, or contain the highest amount of surprise versus expectation, or most likely result in a different model being proposed. To seamlessly enable such capabilities to the domain scientists, therefore, the challenge is to develop the appropriate data management and optimization framework that will allow effective group formation and large scale analyses on the collected data.

Current Practices and Shortcomings: Current citizen science practices primarily rely on *passive form of crowdsourcing*, i.e., forming human networks in a rather organic way. Naturally, this form of passive crowdsourcing leads to high latency, inaccuracy, with the potential of substantial noise in the overall outcome. We, on the other hand, propose `ECCO` to enable *active crowdsourcing* for such scientific data collection, where group formation is optimization guided and the framework constantly learns about the workers from the tasks they undertake and reuse this learning. We provide one such specific scenario next in the context of ecological data collection.

EXAMPLE 1. **Scientific Data Collection and Analyses - A Citizen Science application:** *Typical citizen science efforts take place in groups to reduce errors in observation, or even keep the citizen scientists vested and motivated in the task. Imagine a citizen scientist application needs to be designed to collect data that enables building accurate predictive models by correlating environmental covariates (e.g., elevation, soil type, and average precipitation) with the presence of a species. To formulate the predictive model, the following tasks are to be performed:*

- *Confirmed Absence of Species (subtask-1): Another group (sub-group 1) needs to be formed to confirm the absence of a species. This step is considered difficult and expert workers are required to be involved to carry out this step successfully.*
- *Confirmed Presence of Species (subtask-2): A third worker group (sub-group 2) is to be created to confirm the presence of a species.*
- *Co-variate Validation (subtask-3): A fourth group of workers (sub-group 3) is created to validate the model covariates (e.g., is the elevation really 100 m at this location like the current covariate dataset indicates?).*
- *Model Validation (subtask-4): A final group of workers (sub-group 4) is tasked to validate the model itself (e.g., the system recommends a particular location for sampling a species presence/absence and the group of workers are dedicated to validate that).*
- *This iterative process terminates when the resultant data has surpassed a certain benchmark in quality (for example, the built statistical model needs to reach 90% accuracy and 85% precision). The objective is to achieve the quality as quickly as possible, by spending the smallest cost.*

Group Interactions:

- *Intra-group: Workers in the same sub-group need to interact with each other to ensure that the collected observations are correct and consistent.*
- *Inter-group: The users in sub-groups 1 and 2 are required to interact with each other to confirm the absence and presence of the species.*

Workers' Skills: *Volunteers are likely to have multiple skills, e.g., skills in ecological assessment, field training, etc.*

We attempt to abstract the processes that are likely to take place in such active crowdsourcing application and formalize them and propose `ECCO` to achieve the desired outcome. We identify the following *core aspects to support such ecological applications.*

- Complex Tasks: A citizen science application such as the one above is an example of a complex task that is composed of sub-tasks. For example, each of the data collection step described above is a sub-task and the overall task is a composition of these steps in an appropriate sequence. The current practice is to identify these sub-tasks manually [6, 12]. Interestingly, while the overall goal of a complex task may be to surpass the quality benchmarks as quickly as possible, as stated in the example, each sub-task may have different goal(s). The overall execution flow is presented in Figure 2.
- Groups: Central to such collaborative human-intensive application is the notion of "group" which may further be decomposed into sub-groups, where a set of workers collaborate with each other to complete *tasks*. Example 1 requires 4 such sub-groups.
- Human Factors: A variety of individual and group based human characteristics are to be understood [8, 9]. For example, skill of the workers to identify experts, their incentives, motivation, or ability to collaborate with each other. Some relevant skills pertinent to the running example may be ecological assessment skill, field training, and so on.
- Primary Functionalities: (a) given a complex task and a worker pool, form group of sub-groups to assign to the sub-tasks; depending on the nature of the applications, a sub-group may undertake one or more subtasks, collaborate or compete with other sub-groups. (b) learn skills and other human factors of the workers that are either individual or group based.
- Scale: We envision the necessity for a generic system that can handle a wide variety of such applications. In such a system, hundreds and thousands of citizen science workers and tasks needs to be processed and assigned. Scalable solution design becomes the first class citizen in such settings.

Desirable characteristics of ECCO: Several key aspects are to be appropriately unfolded: (1) uncertainty in human characteristics, namely **human factors** [8, 9] are to be understood. More importantly, we need to identify both individual human factors and those that impact group dynamism; For example, individual human factors, such as a user skill may impact how much leadership she has in a group; (2) **designing declarative primitives** which allow domain experts to easily accomplish the required functionalities; (3) **designing relevant mathematical models** to capture the appropriate optimization objectives; (4) **designing appropriate algorithms and data management techniques** for effective task assignment and human factors learning. (5) finally, **developing actual systems** or platforms that can integrate the components of ECCO.

Proposed components: ECCO consists of two primary components which are both required to be adaptive.

- *Collaborative Human Factors Learning Component:* This component first formalizes human factors - some of these characterize individuals, such as, their respective skills in different domains, their incentives (e.g., wage), motivation, as well as describe group characteristics, affinity between the workers, trust, leadership, or even application characteristics, such as, critical mass (a socio-psychological concept that describes how large a group can be for effective collaboration) [5]; These factors are then leveraged within this component to accomplish different learning, as described in Section 2. This module also exploits a "feedback" loop to enrich its learning by ingesting data coming from the deployment platform. This very aspect of users evaluating other users is a clear departure of ECCO from any existing system.
- *Human-Centric Optimization Component:* This component consists of two different optimization engines and heavily interacts with the human factors learning module to appropriately incorporate human factors in the design. (a) *Adaptive Task-Assignment Engine* is in charge of building a set of *homophilous, diverse, or complementary* groups by enabling different *interaction patterns among group members and accounting for appropriate human factors* to optimize certain outcomes of a given task. Furthermore, as stated in the running example, different groups may have different interaction pattern with each other. (b) *Human Data Management Engine*, on the other hand, manages the data learned from human factors learning component, as well as the data generated by the task assignment engine. The overall objective of this engine is to store, index, and effectively retrieve the collected data over the time. (c) Last, but not least, we wish to support *adaptivity and incremental* computation in both those engines, as human factors change over time. Figure 1 describes a high level architecture and interactivity among the different components inside ECCO.

Team formation [1] in online social networks has been the subject of some recent works which bears resemblance to the task-assignment problem. What differentiates us, is the *time-variance property and significant interoperability between different components, by deliberately acknowledging a wide variety of human factors.* Obviously, no attempt has been made to incorporate human factors learning for collaborative applications, or to effectively manage data generated by human workers.

Sections 2 and 3 contain further details. Our goal is to ensure scalability, as well as allow incremental and adaptive learning and computation. In addition to benefiting ecological and environmental science, we envision that ECCO will transpire many data management, index design, algorithmic, machine learning, and social science research problems and foster synergy across these disciplines.

2. ECCO

Individual users and applications which consist of tasks are integral part of ECCO. ECCO works in conjunction with an evaluation environment where tasks get evaluated by a human machine computation model. ECCO's components are:

2.1 Collaborative Human Factors Learning

Different collaborative applications rely on capturing and including individual human factors such as skill, motivation, acceptance ratio (describing how likely an individual will contribute) [9], or expected wage. Similarly, group human factors, such as, affinity, leadership, influence, group size (referred to as critical mass [5]) are also to be factored in. Moreover, while new users may join, existing ones may leave. Interestingly, human factors are dynamic - i.e, they change over time, and depend on the context. Human factors are also *correlated* (e.g., a highly skilled individual may be more influential, or higher rewards lead to higher acceptance ratio), and sometimes probabilistic (e.g., acceptance ratio).

While prior work [9] has acknowledged human factors, no further attempt has been made to *learn and incorporate* them in human-intensive applications in a principled manner. On the contrary, the collaborative human factors learning component is considered as one of the most fundamental contributions of ECCO, designed with the overall objective to learn the collaborative human factors [8, 9] adaptively. It proposes a set of declarative primitives to learn the (1) individual human factors, (2) group based human factors, (3) correlation among different human-factors, (4) most importantly adaptive and incremental learning of these factors, considering the achieved quality of the group based tasks. Recall the feedback loop in Figure 1 that comes from external evaluation to this component. For our citizen science example, the evaluation is performed with a hybrid human and machine intelligence. In particular, the evaluation of the completed tasks could be *precision, recall, accuracy, sensitivity*, etc. The corresponding vector in *evalmatrix* may look like, $precision = 0.8, recall = 0.6, accuracy = 0.5, specificity = 0.5$. Function `relearn` is designed to relearn how to obtain the skills of the workers (ecological assessment knowledge, field training, etc) from these evaluation values. Some example primitives are provided in Table 1.

2.2 Human-Centric Optimization

This component consists of two optimization engines.

Adaptive Task-Assignment Engine: Inputs to this engine are the user population and the tasks (or a set of subtasks), and the output are the *groups that are best suited to undertake the tasks.* Primitive `Form-Grp(`t,$\mathcal{U}$`)` is designed for this purpose, which is further explained in Table 1. No-

Primitive	Description
`human-factor-ind(u)`	output the individual human factors of user u.
`human-factor-grp(g)`	output the group based human factors of g.
`cor-human-factor(`$\{X\}, k$`)`	output the correlation among the human factors in set $\{X\}$.
`relearn(`$\{X\}', T, evalmatrix$`)`	relearn the human factors in the set $\{X\}'$, considering T tasks and the *evalmatrix*.
`Form-Grp(`$t, \mathcal{U}$`)`	output the assignment of a set of workers from the available worker pool $\mathcal{U}$ to task t.
`Form-group(L, clique,` $th, \mathcal{U}$`)`	output a clique of L workers whose aggregated all-pair affinity is th or beyond.
`add-worker(u), delete-worker(u)`	adds and deletes worker u to/from the available tasks.
`find-worker(human-factor)`	find worker with a given human factor.
`find-top-worker(s)`	Find highest skilled worker for skill s.
`find-top-worker(s,k,time-period)`	Find k highest skilled worker for skill s for a given time period.
`add-eval(`g, t^1`)`	add evaluation score of subtask-1 of task t completed by group g.

Table 1: Example Primitives Descriptions Supported by ECCO.

tice that, the group formation problem for us, is *optimization mediated, instead of organic*. Needless to say that each user is described by a set of human factors that are learned from the *collaborative human factors learning* component. The criteria of the *best outcome* (i.e., optimization objectives) is domain-specific to say the least, and to be left for the domain experts to decide. We, however, provide the mechanism to incorporate these criteria into a set of well formulated optimization models.

As a simple example, given a sub-task, such as finding the workers group that can collect initial data to build the statistical model (subtask-1), the group should be formed such that the users *collectively* have the expertise to collect both positive and negative labels for the statistical model, *their wages do not surpass the cost budget of the task*, and group should be designed such that it brings forth the *maximum collaborative synergy*. On the contrary, given a complex task with a set of sub-tasks (such as one described in the running example) and described in Figure 2, we need to form a group of groups, where workers inside the same sub-group must be highly collaborative, and workers across some sub-groups need to interact with each other as well (for example, sub-group-1 and sub-group-2 in the running example). This engine is responsible to analyze the desiderata of task-assignment and form groups to enable the desirable outcome.

Naturally, even the simplest settings for such problems give rise to complex mathematical formulations having multiple-objectives to optimize. Then, the interaction pattern gives rise to constructing graphs involving the workers, where the topology of the graph should conform a specific interaction pattern, as described by the domain experts. For the example task stated in the running example, each group interaction translates to forming a clique to execute a sub-task and the interaction between the sub-groups give rise to forming a connected topology among the cliques with highest affinity, as described by Figure 2.

Finally, how big a subgroup should be is application-specific many times. We envision that ECCO would support a variety of such applications, where the group formation is premeditated and guided by a well-defined optimization objective.

Human Data Management Engine: The human factors learning component constantly generates data involving individual human workers and group of workers over the time that the task assignment engine needs to tap into to enable effective assignment of worker groups to the sub-

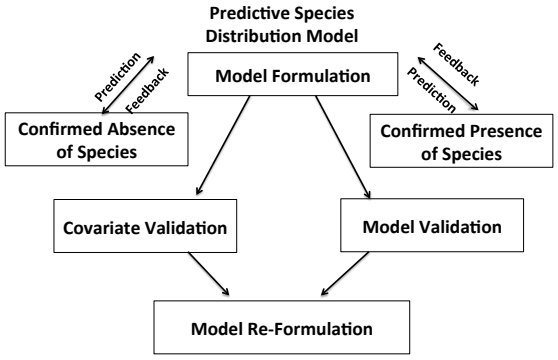

Figure 2: A Complex Task with Sub-tasks Using Example 1.

tasks. Not only that, the evaluation component generates the evaluation of the completed tasks. Interestingly, this data is temporal [3] and is associated with time stamps. Human Data Management engine provides effective management over this dataset to enable effective storage and retrieval over the time. We explain some of such functionalities next.

This engine will be enabled with the traditional database primitives, such as, add, delete, or find a user with a given human factor value, as explained in Table 1. Additionally, the domain experts or the data analysts may be interested to perform simple statistical analyses on this collected data, for example, finding the highest skill worker for a given skill s, or finding the top-k highest skilled workers, and so on. The corresponding primitives are described in Table 1. In an earlier work, we have proposed an effective indexing technique to cluster the workers based on skills and wages [9].

Recall Example 1 and notice that the complex tasks consists of a set of sub-tasks. We propose primitives to add evaluation score of a completed task (e.g., sub-task-1 in Example 1), completed by a group. Similarly, for efficient task assignment, ECCO will leverage this engine to quickly find a group of workers who are most skilled to undertake a given task (e.g., finding the best set of workers for each of the sub-tasks in Example 1). In an recent work, we propose the notion of `virtual worker`, an effective indexing technique to cluster the workers based on skills and wages for effective worker to task assignment [9]. Additionally, our proposed human data management engine is empowered to retrieve worker groups that will optimize a particular interaction pattern. For example, we will design primitives to

retrieve a clique of L workers with affinity more than some threshold th. We refer back to the Table 1 again for the exact definition of the primitives.

3. CHALLENGES & DIRECTIONS

We describe some of the major challenges in realizing ECCO and our proposed directions.

(a) Identifying Relevant Factors: One significant challenge is to identify a wide variety of human factors and other necessary semantics that are needed in such applications. We consider a platform with a set of n workers, $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$[4] and l tasks, $\mathcal{T} = \{t_1, t_2, \ldots, t_l\}$. A task may be *indivisible* or may be decomposed to a set of sub-tasks. A task requires multiple skills over m different domains, $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$.
(1) Human Factors: Our initial effort has identified the following factors that potentially have a dramatic impact on the ecosystem. *(a) Skills:* The skill of a worker u is expressed as a vector $s_u^1, s_u^2, \ldots, s_u^m$ over $\mathcal{S}$, where each skill is quantified in a continuous scale between $[0, 1]$, where a value of 0 reflects no expertise for that skill. *(b) Wage/Cost:* For many collaborative tasks, explicit monetary remuneration may need to be offered to the workers. w_u represents the amount of money a worker u is willing to accept to complete a task. *(c) Acceptance Ratio:* Acceptance ratio $p_u \in [0, 1]$ of a worker u is the probability at which u accepts a task. *(d) Worker affinity:* A key to successful collaboration is the affinity among the individuals. At the atomic level, affinity is defined between a pair of workers u, u', i.e., $Aff(u, u')$ denotes how well these pair of workers work with each other.
(2) Task Quality Metrics: A task t may consist of a set $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_r\}$ of quality metrics (as described in the running example, such as precision, recall, etc). We envision that the relevant metrics to estimate the task quality would be domain specific and the *value* of a quality metric may be an *additive, multiplicative, or a more complex function* of individual worker's skills or other human factors.
(3) Constraints in Collaborative Tasks : A task t may have a budget (cost) threshold of C_t. To be considered successful, it may also have a set of quality metrics threshold Q_t^i ($i \in \{1 \ldots r\}$) with the total expenses less than C_t. Global constraints should also be considered. For example, each workers should neither be under nor be over-utilized, by assigning a lower (X_l) and an upper (X_h) limit on the number of tasks she can be assigned to.

(b) Modeling: Appropriate incorporation of the human factors is one of the foundational steps in the successful development of the *Collaborative Human Factors Learning Component*. Similarly, the *Human-Centric Optimization Component* have to formalize complex optimization problems with multiple objectives and constraints. In our initial direction, we realize that it is only realistic to collaborate with the domain experts to understand and appropriately incorporate the application specific human factors. As an example, for the species data collection task described in the running example, human factors could be ecological assessment ability, field training, workers' affinity with each other, explicit monetary incentives, etc. After that, a mathematical model is to be formalized that incorporates those factors appropriately, where it would maximize some of the factors, and use the rest as constraints. A simple mathematical model may intend to form a group $\mathcal{G}$ which maximizes the aggregated expertise Σu_{d_i} of the users as well as their collaboration affinity $\Sigma \text{Aff}(u_i, u_j)$, while keeping the total cost under a certain threshold $\Sigma w_u \leq C$, such as:

$$\text{Maximize } \Sigma_{\forall_{u_i, u_j \in \mathcal{G}}} \text{Aff}(u_i, u_j) +$$
$$\Sigma_{\forall_{u_i \in \mathcal{G}}} u_{d_i}, \Sigma_{\forall_{u \in \mathcal{G}}} w_u \leq C$$

On the contrary, if the task-assignment optimization is performed globally, we also need to add the load balancing constraints. However, the question remains, how to acknowledge in the modeling that not all users will perform according to their expertise, or a group may have to be partitioned into sub-groups if it violates the critical mass constraint. Similarly, *task creation engine* is also designed to optimize outcomes (such as minimizing latency), while satisfying the constraints provided by the domain experts.

(c) Learning: The *collaborative human factor learning* component hinges on automated learning techniques to uncover the correlation among the human factors. Several interesting and challenging problems surface that involve designing learning algorithms. For example, what makes individuals or a group remain motivated, or how to learn worker skills for collaborative tasks? Such problems are likely to give rise to novel supervised or unsupervised machine learning solutions.

Let us consider a simple illustration of the function `relearn`$(\{X\}', T, evalmatrix)$ in Section 2.1, where we are given a matrix *evalmatrix* that provides how each task t_i is evaluated based on various task quality metrics. Learning worker skills could be posed as a matrix tri-factorization problem, where *evalmatrix* is factorized as, - i.e. $evalmatrix \approx FX'G^T$ where the approximation accuracy is measured based on the norm $||evalmatrix - FX'G^T||$. Matrix $\overline{F}$ is a Boolean matrix and has the assignment of workers to groups in different tasks. X' denotes the worker to human factor matrix. The final matrix G measures the impact of human factors to task quality metrics, specifying that metric G_i as a linear combination of human factors. This tri-factorization [7] is heavily constrained using non-negativity, sparsity, row/column stochasticity, or other *marginal* constraints.

(d) Adaptivity and Incrementality: Adapativity is essential for the survival of ECCO from several perspectives - with changing time and context, individual and group human factors, as well as their correlation will vary. This not only requires the two of the first three challenges (i.e., modeling and learning) to be time-aware, but also to be adaptive in nature. For the *Human Factors Learning* component, this means that ECCO should be able to adaptively learn the human factors, as they perform more actions in the system. For the *task assignment engine*, it would mean that the system would be able to incrementally form groups as more users join or existing ones leave the platform. To enable adaptive and incremental computation the *human data management engine* needs to be sensitive to the footprints of a workers' activity in a temporal fashion. Understandably, incremental computation may introduce approximations in the results. In a recent work of ours, we have proposed how to perform adaptive task assignment by marginally solving the problem

[4]Although new workers could join and existing ones could leave any time.

and our experimental results demonstrate that our proposed solutions are both effective as well as efficient [9]. An interesting study would be to investigate the approximation factors of the proposed algorithms theoretically.

(e) Scalability: The task assignment problem is shown to be NP-hard[2], even in the simplest scenarios [9] even without considering affinity. Similarly, matrix factorization problem is inherently NP-hard [7]. Therefore, all the components of ECCO must ensure efficient algorithms for human factors learning or task assignment. Proposed human data management engine therefore needs to be appropriately designed to ensure efficient computations. When affinity is considered in the modeling, the very simple task assignment formulation described itself gives rise to complex graph partitioning formulation. At the same time, we intend to stay as principled as possible. Traditional modeling and learning algorithms that are typically inefficient may come largely inappropriate in our settings, thereby requiring to generate new theory and techniques. We foresee that the scalability challenges of ECCO will nurture engagement and collaboration across the theory, database, and machine learning community. Efficient approximation algorithms with theoretical guarantees will be proposed, or we expect to see innovative pre-computation or index design solutions to enable real time response. A very few existing research efforts [9, 1] superficially investigates some of these scalability issues for the task assignment problem. How to design the human data management engine effectively to enable efficient task assignment and human factors learning remains to be an open problem.

(f) Platform Design: ECCO would not be possible without the ability to conduct comprehensive experiments and validate the outcomes. Note that, finding appropriate datasets that represent the real world is one of the toughest barriers that we yet have to surpass. Most of the existing platforms, commercial or academic, such as Amazon Mechanical Turk (www.mturk.com), CrowdDB, Qurk, Deco, do not naturally support collaborative tasks. To go beyond theoretical analyses, the community needs to have access to one or more platforms that support collaboration and group formation, where the experiments and analyses can be conducted systematically. We expect that ECCO will transpire enough system research to build platforms and propose declarative languages to support collaborative human-intensive applications. Without appropriate evaluation strategies, indeed, the effectiveness of ECCO will only be partially explored.

4. CONCLUSION

We propose the vision of ECCO, a framework that supports data management and analyses for ecological data by leveraging the innate characteristics of individuals. We outline the two core components of ECCO - 1) Collaborative Human Factors Learning Component, 2) Human-Centric Optimization Component. The first component is designed to learn and characterize individual and group behaviors over time, their interdependence, which is designed to closely work with the evaluation or the deployment environment. The latter is an optimization component which interacts with the former to leverage human factors in the modeling and computations. This component is intended to automate *worker to task assignment* which are largely manual (or self meditated) and painfully slow till date. ECCO warrants adaptivity and scalability - to support that we propose the necessity to design an appropriate *human data management engine* that will collect and manage data coming from the human factors learning component and use that in task assignment. We intend to design principled solutions that are effective as well as efficient. We summarize the challenges of ECCO and propose initial directions.

5. REFERENCES

[1] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi. Online team formation in social networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 839–848. ACM, 2012.

[2] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[3] C. S. Jensen and R. Snodgrass. Temporal data management. *Knowledge and Data Engineering, IEEE Transactions on*, 11(1):36–44, Jan 1999.

[4] S. Kelling, J. Gerbracht, D. Fink, C. Lagoze, W. Wong, J. Yu, T. Damoulas, and C. P. Gomes. ebird: A human/computer learning network for biodiversity conservation and research. In *Proceedings of the Twenty-Fourth Conference on Innovative Applications of Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*, 2012.

[5] R. Kenna and B. Berche. Managing research quality: critical mass and optimal academic research group size. *IMA Journal of Management Mathematics*, 23(2):195–207, 2012.

[6] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut. Crowdforge: crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 43–52, New York, NY, USA, 2011. ACM.

[7] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[8] S. B. Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Crowds, not drones: Modeling human factors in interactive crowdsourcing. In *DBCrowd*, 2013.

[9] S. B. Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Optimization in knowledge-intensive crowdsourcing. *CoRR*, abs/1401.1302, 2014.

[10] J. Soberón and T. Peterson. Biodiversity informatics: managing and applying primary biodiversity data. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 359(1444):689–698, 2004.

[11] B. Sullivan, W. Christopher, M. J. Iliff, M. J. Bonney, D. Fink, and S. Kelling. ebird: A citizen-based bird observation network in the biological sciences. In *Elsevier, Biological Conservation*, 2009.

[12] H. Zhang, P. André, L. B. Chilton, J. Kim, S. P. Dow, R. C. Miller, W. E. Mackay, and M. Beaudouin-Lafon. Cobi: communitysourcing large-scale conference scheduling. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 3011–3014. ACM, 2013.