OSS コミュニティおよびクラウドソーシングの統合による ソフトウェア開発者の能力予測

大澤　昇平† 　松尾　豊†

† 東京大学大学院工学系研究科　〒 113–8656 東京都文京区弥生 2-11-16
E-mail: †{ohsawa,matsuo}@weblab.t.u-tokyo.ac.jp

あらまし　ソフトウェア開発プロジェクトの成功は，開発チームに所属するソフトウェア開発者の持つ能力に依存することが知られている．しかし，一般にソフトウェア開発者のこうした能力の推定は難しいとされている．本研究ではこうしたソフトウェア開発者の能力を，オープンソースソフトウェア（以下，OSS）コミュニティとクラウドソーシングの情報を用いて予測することを試みる．提案手法では，OSS コミュニティである GitHub と，クラウドソーシングサービスである oDesk のデータを統合し，oDesk から得られる情報を教師データにした能力の予測モデルを構築する．構築した予測モデルを nDCG によって評価し，実際に両方のデータを統合したモデルが最も精度が高いことを示す．

キーワード　能力予測，情報統合，エキスパート検索，GitHub, oDesk

1. はじめに

近年，ソフトウェア開発プロジェクトにクラウドソーシングからエンジニアを起用するという動きが行われている．ソフトウェア開発プロジェクトの成功は，開発チームに所属するソフトウェア開発者の持つ能力に依存することが知られている．たとえば，Crowston は，ソフトウェア開発プロジェクトの成果物が多くの人にダウンロードされる要因の一つとして，ソフトウェア開発者のプログラミング言語の習得度を挙げている [6]．しかし，一般にソフトウェア開発者のこうした能力の推定は難しいとされている [18]．

クラウドソーシングでは一般に一つの案件に対して複数のエンジニアの応募があるが，採用できるのは原則として一人のみである．そのため，候補となるエンジニア全員が初対面である場合，一人を選択するために，エンジニアの過去の実績を参考にできると有用である．

本研究では，OSS コミュニティの情報に基づき，クラウドソーシングのエンジニアの評価値を予測する問題を解く．OSS コミュニティとクラウドソーシングのエンジニアとの統合を行った上で，統合した情報に基づき素性生成を行い，クラウドソーシングのユーザの能力を予測するモデルを構築する．

実験では，対象とするクラウドソーシングサービスとして oDesk (注1) を想定し，実績情報を用いる OSS コミュニティとして，エンジニアの実績が多く蓄積されている OSS コミュニティの一つである GitHub のデータを用いる．評価値予測の評価には，検索システムの評価指標を用い，提案する手法が最も高くエンジニアのランキングができることを示す．

(注1)：https://www.odesk.com/

2. 関連研究

本研究が関連しているエキスパート検索について取り上げる．

エキスパート検索 (expert search) は，組織の文書などを格納しているデータベースから，目的の専門性 (expertise) を有したエキスパートを検索する問題である．エキスパート検索に関する分野は，Future Challenges in Expertise Retrieval (fCHER) を筆頭とするワークショップが開かれたり，評価用のデータセットが Text REtrieval Conference (TREC) から提供されたりするなど [5]， 盛んに研究が行われている．なお，エキスパート検索と同様のタスクとして，専門性特定 (expertise idenfication) [4]，エキスパート配置 (expert location) [11]，エキスパート探索 (expert finding) [1] があるが，本研究ではこれらをすべてエキスパート検索の関連研究として扱う．

エキスパート検索に関する文献では，専門性をある対象 (subject) に対する高度な知識 [2] あるいはスキル [11] であるとしており，知識と能力の二つの意味が混在している．本研究では専門性とはスキルであるという立場を取る．専門性の具体的な定量化の方法に関しては合意のとられたものは存在せず，トピックとの関連度により定量化する方法 [1] や，人物にリファレンスを取ってスコアを得る [4] などの方法がとられている．

エキスパート検索の目的は大きく分けて二つある．一つは，ある対象が与えられたときにそれに関連する人物を検索することであり，もう一つは，ある人物が与えられたときに，その人物の専門性を列挙することである．いずれも，ある人物の専門性をどのように予測するのかという問題に帰着される．

文書に対して著者の専門性の推定を行っている研究がいくつかある．Balog [1] らは，研究者のホームページから専門性を推定する手法を提案している．彼らは，候補者 c のクエリ文字列 q に対する尤度 $p(q|c)$ を候補者の執筆した文書の単語頻度をベースにモデリングすることにより，候補者のスコアリングを行っ

ている．評価用のデータセットとして，TREC 2005 Enterprise Track [5] のデータセットを用いている．これは 1,092 人で構成される研究者に関するデータで，学会のワーキンググループに所属している人物は，その学会に関連する専門性を持っているという前提で構築されている．

ソーシャルグラフを用いることにより専門性を推定している研究がいくつかある．Campbell らの研究 [4] では，社内の電子メールの受送信ネットワークに基づき，社員の専門性を特定することを試みている．彼らは電子メールの送受信ネットワークにネットワーク中心性の計算アルゴリズムの一つである HITS アルゴリズム [10] を適用することで，専門性を抽出している．実験では 2 社を対象にし，60 人で構成される社員間の電子メール送受信ネットワークを対象にし，約 29,000 件のメッセージを対象にしている．評価では社員をよく知る人物に点数をつけてもらうことで行っている．Zhang [17] らの研究では，より一般的に，複数ラベルを持つソーシャルグラフを対象に中心性を計算し，専門性を予測する手法について提案している．実験では，情報抽出により得られた 240 万人の研究者間ネットワーク [16] を対象にし，共著関係，師弟関係，親子関係を考慮に入れたスコアリングを行っている．

これらの研究と本研究が異なる点はデータの量と質である．量として，ウェブから取れるデータと組織内の文書には分量の差がある．また，質として，ウェブからはエキスパートの開発物が取れるという点が異なる．これはシステムとのインタラクションによってユーザの開発物がデータベース上に蓄積されるようになって初めて実現したことである．

OSS コミュニティの発展により，ネットワーク構造を用いた専門性推定を OSS コミュニティに対して行っている研究もある．Ghosh らの研究 [7] では，Twitter のユーザを対象に専門性予測を行っている．具体的には Twitter のリストと呼ばれる機能を用いることで専門性を高い精度で予測できることを明らかにしている．

本研究はエキスパート検索の中では，ウェブ上の情報を統合することによりエキスパートの検索を行う手法と位置づけることができる．具体的には，GitHub とクラウドソーシングサービスの oDesk とを統合することにより，エキスパートの検索を実現している．評価は 550 人のユーザに対して人物による判断により行っている．本研究に近い研究として，同時期に行われた Bozzon らの研究 [3] が挙げられる，OSS コミュニティの情報を統合することによりエキスパート検索を実現している．彼らは Twitter, Facebook, LinkedIn の情報に基づき専門性の推定を行っている．しかし，彼らは実績情報をデータセットとして用いていない [3]．本研究ではこれをクラウドソーシングのデータを用いることにより解決している．

3. 提 案 手 法

図 1 に，問題設定および提案手法の概要を示す．発注者はクラウドソーシングに対し，クエリ q を入力する．これにはユーザの一覧を探すのにプログラミング言語を入力することや，ユーザを募集するプロジェクト概要書を提出する行動が該当する．システムは q に合致するユーザを，関連するか否かに応じて複数の候補に絞り込む．絞り込んだ結果に対して発注者は仕事を依頼する一人を選択する．

このとき，依頼する一人を選択するために，絞り込んだ複数のユーザに対してのランク付けを行う．ランク付けのためには，ユーザのスコアリングを行う必要が生じる．ユーザのスコアリングを行う際に，他の OSS コミュニティとの統合を行う．このために他の OSS コミュニティから得られる情報源を統合し，根拠になるデータとして用いる．

情報を統合したとき，それぞれのデータをどのように扱えばよいか考える．通常，二つのサービスは目的が異なるため，どの属性を扱えばよいかというのは決定しない．そこで，同じ意味を持つ関係や属性同士をマッチングすることによって，情報源の統合を行う．

本研究では，ユーザと意味的関係にあるエンティティ群に着目する．こうしたエンティティとしては，友人や作品が考えられる．たとえば，OSS コミュニティにおいてエンジニアは何らかの作品を残しており，これらに対する評価をフィードバックとして得ている．これらのフィードバックを集約することで，エンジニアの能力の予測を行う (注2)．

3.1 素 性 生 成

素性生成では，エンジニア候補集合 U に含まれるエンジニア u に対して，素性ベクトル $\psi : u \to \mathbf{R}$ を生成する．このとき，エンジニアが他の OSS コミュニティで生成したエンティティ群の評価値を集約することでエンジニアの評価を行う．集約方法には様々な方法が考えられるが，意味的関係にあるエンティティの質と量をなるべく公平に反映できる手法が望ましい．

単純な手法がいくつか考えられるが，どれも公平な評価は難しい．第一に，エンティティの評価値平均を用いる方法が考えられる．評価値平均はプロジェクトの質を評価できる一方で，量的な情報を含んでいない．第二に，エンティティ数を用いる方法が考えられる．この方法は量的な情報を含んでいるが，質的な情報を含んでいない．第三に，エンティティの評価値を合計したものが考えられる．これは量的な情報，質的な情報ともに多く含んでいるが，一定の成果を連続して出している人と，一回しか成功せずその後継続して成果を続けていない人を同程度に評価してしまう性質がある．

一方で，研究者の評価も論文の引用数によって行われることが多いため，アカデミアからのアナロジーによりエンジニアの評価を行うことを考える．研究者の評価に **h-index** [8] が使われるが，これは引用数を集約する関数であるとみなすことができる．h-index は降順に整列され論文に対して，評価値が h 以上の論文が h 本存在するような最大の h を指し，上記で挙げた集約関数の問題を解決している．

本研究もこの h-index の類推により **s-index** を提案する．s-index はユーザが OSS コミュニティ上に残した作品に対して，評価値が s 以上の作品が s 個存在するような最大の s を指す．

(注2)：成果物に対するユーザからのフィードバックという同一の意味を用いることにより，情報源同士を統合していると言える．

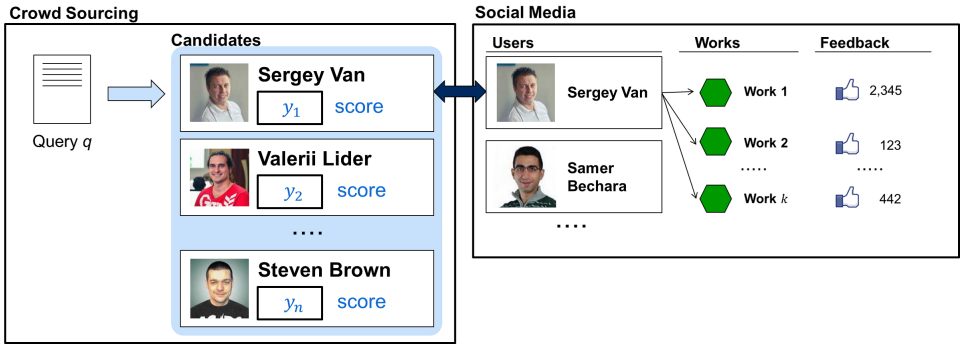

図 1　問題設定および提案手法の概要

また，本研究では s-index に加えユーザのクラウドソーシング，OSS コミュニティ上のプロフィールから得られる素性を併せて用いる．

3.2　予測モデル

ここでは，得られた素性に基づき，予測モデルの構築を行う．

本研究の目標は，対象ユーザ集合 U に含まれる任意のユーザ $u \in U$ に対し，u の能力値（専門性，信頼性，総合的能力）を予測することである．そのために，U の部分集合 $\tilde{U}$ とその能力値の集合 $\tilde{L} \subset \tilde{U} \times \mathrm{R}$ が予め教師データとして与えられることを前提に，$\tilde{L}$ を線形回帰モデル $f(a) \equiv \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\psi}(u) + c$ を用いて，一般の U に対して汎化する．

$\boldsymbol{w} \in \mathbf{R}^d$ は重みベクトル，$c \in \mathbf{R}$ はバイアスである．なお，$\left|\tilde{U}\right|$ は $|U|$ に対して十分小さいものとする．$\boldsymbol{w}, c$ の最適化には**サポートベクター回帰** (support vector regression; SVR) を用いる．SVR とは**サポートベクターマシン** (support vector machine; SVM) を用いた回帰手法の一つである．SVM は学習器の一つであり，入力される複数の素性ベクトルの中から予測に貢献する一部（サポートベクター）を一定の基準で選定し，新たに到着する素性ベクトルに対して，サポートベクターとの間の類似度の線形結合により予測を行う．入力のすべてを用いないことでスパースなモデルを実現し，過学習を回避している．

SVM では，**カーネル** (kernel) と呼ばれる素性ベクトル間の類似度を計算する関数を指定する必要がある．これには，線形カーネル，動径基底関数カーネル，多項式カーネルなどが存在する．本研究では，入出力間に線形性が成立する**線形カーネル** (linear kernel) を利用する．これは，線形なモデルを用いることで，モデルの解析を容易にするためである．

3.3　教師データ

教師データとしてエンジニアのスコアを計算する関数をどのように定義するかについて議論する．クラウドソーシングにはエンジニアの成果物に関して評価する仕組みがあるが，oDesk において多くのクラウドソーシングにおいて多くのユーザは経験数が少なく，正しい評価が得られていない．

具体的にクラウドソーシング上でユーザの評価を行う単純な方法には，平均フィードバック，プロジェクト数，平均時給が挙げられる．平均フィードバックは，顧客満足度を勘案できるフィードバックである．しかし，大体は満点を獲得できることや，安いプロジェクトの方が高いフィードバックを取りやすいこと，10 回満点を取ったのと 1 回満点を取ったのが同じ評価になるなどの問題がある．プロジェクト数は，経験の厚みを評価できる一方で，顧客が満足していないプロジェクトを連発しても意味がないこと，平均フィードバックと同様に時給の情報が含まれていないことが問題である．最後に，平均時給は高い時給のプロジェクトを連発している人を評価できるという長所がある一方で，顧客満足度がわからないことや，一回だけ高い時給のプロジェクトを行っただけの人を高く評価してしまうことがある．

以上の議論を踏まえ，フィードバック，プロジェクト数，時給の 3 つの情報を用いる手法をスコアとして提案する．本研究では，クラウドソーシングの時間単価とフィードバックに基づく指標を評価データとして用いるため，次式で定義される**補正フィードバック比重平均時給** (adjusted feedback-weighted average rate; AFWAR) を用いる．

$$\tilde{r}_u \equiv \left(1 + \frac{\lambda}{|P_u|}\right)\left(\frac{1}{|P_u|}\sum\nolimits_{p \in P_u} f_p r_p\right)$$

$\tilde{r}_u$ は u の AFWAR，P_u は u のすべての成果物の集合，f_p は p のフィードバックを $[0,1]$ の範囲に正規化したもの，r_p は p を生産する仕事に関して支払われる報酬を時給換算したもの，λ はスムージング係数である．右辺の変数はすべてクラウドソーシングから得ることができる．

4.　実　　験

実験では具体的に，クラウドソーシングとして，ユーザ数が現在世界最大である oDesk を用い，エンジニアの評価を行う．統合する OSS コミュニティとして，OSS コミュニティの一つである GitHub を用いる．GitHub は OSS の開発を行っているユーザがそれぞれのシステムを公開している．このシステムを作品群とみなし，それぞれのシステムが Fork （再利用）された回数を評価として用いる．

4.1　データセット

4.1.1　GitHub

ここではエンジニアの情報を取得するために GitHub へのアクセスを行う．GitHub は第三者向けに API を提供しており，本研究でもこの API を対象に通信を行う．GitHub API はユーザの名前や所属といった基本的な属性に加え，これまでどのようなオープンソースプロジェクトに貢献してきたかとい

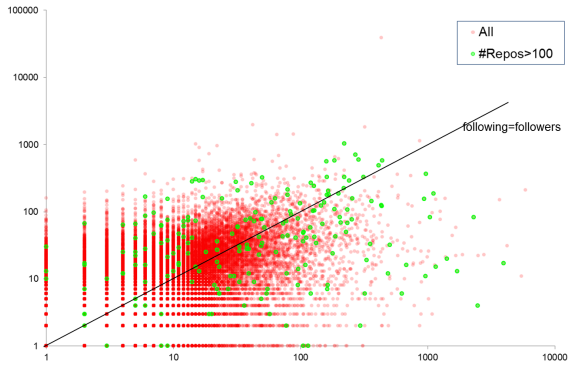

図 2　被フォロー・フォロー関係

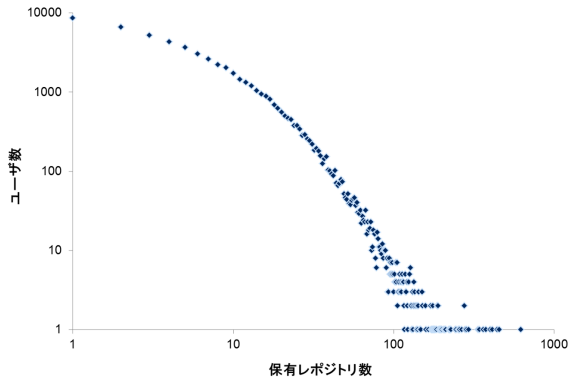

図 3　レポジトリ数の分布

う数や，ユーザ同士の友人関係を取得することができる．

これらのデータは，グリーディ法によって収集する．GitHub は検索 API を公開しているものの，今回はユーザ間のつながりやレポジトリの素性も取得したいため，グリーディ法を採用する．

GitHub の提供する API を通して 2 万ユーザをサンプリングし，それぞれのユーザに対して素性を生成した上で，全体の傾向について調査を行う．

図 2 に被フォロー数とフォロー数の間の関係を示す．これを見ると，被フォロー数とフォロー数は両対数軸上で相関する傾向があり，レポジトリが多い人は，その分フォロー・非フォローも多く，活動の活発さがという成分がある可能性があると言える．

図 3 にレポジトリ数の分布を示す．一部のユーザは 100-1000 という大量のレポジトリを保有している．また，ベキ分布にはあまり近くなく，1-10 付近はなだらかな減少となる．サンプリング方法が，ユーザ間のフォロー関係をたどる方法であったため，利用頻度の少ないユーザがサンプリングされていないことも考えうる．

4.1.2　oDesk

本研究では oDesk から独自にクローリングを行い，4×10^4 ユーザの情報を収集した．

図 4 に典型的な oDesk ユーザのプロフィールを示す．基本的には Facebook などの通常の SNS のプロフィールの要素に加え，oDesk のユーザは，スキル (skill)，カテゴリ (category)，ポートフォリオ (portfolio)，仕事履歴 (work history)，フィードバック (feedback)，時間単価 (rate) といった属性をプロフィールに表示している．スキルとカテゴリは遂行可能なタスクを表現したものであり，カテゴリの方がスキルよりも分類が粗い．また，それぞれ複数をプロフィールに書くことができる．たとえば，`Web Developer` 兼 `Illustrator` などである．仕事履歴とポートフォリオはどちらも過去に行った仕事を表示する欄で，前者は oDesk 上の仕事であり後者はそれ以外の仕事である．ポートフォリオには，たとえば，ウェブ開発者は過去に開発した製品への URL をここに書くことができる．フィードバックは，各タスクの成果物に対する依頼者からの 5 段階評価を平均したものである．また，時間単価はその人の 1 時間あたりの仕事に対して支払われる金額である．

まず，図 5 に，クラウドソーシングの一つである oDesk の統計データを示す．図 5(a)–(f) を順番に説明する．図 5(a), (b) は，それぞれプロフィールに記載されたスキル数，カテゴリ数の分布である．スキルの分布は二極化しており，まったくスキルを表示していないユーザから，上限いっぱいまで表示しているユーザにわかれている．また，カテゴリは 2 を最頻値とする釣鐘型の分布になっている．図 5(c) は，ポートフォリオのアイテム数の分布である．グラフから明らかなように，べき分布をしている．また，グラフから 100 件近くポートフォリオを持っている人もいることがわかる．図 5(d) は，フィードバックに関するグラフである．フィードバックに関しては約 8 割が 4.5 以上の評価を持っていることがわかる．図 5(e) は，登録日に関するグラフである．2013 月夏から急上昇し，2014 年 7 月をピークに減少に転じている．図 5(f) は，最終活動日に関するグラフである．右の点ほど，最近活動したユーザを表している．最終活動日が 1ヵ月以上前のユーザが 8 割であり，それほど多くのユーザが活発に活動しているわけではない．これは，毎回のタスクが大きなプロジェクトであることや，登録のみを行い，その後案件を行っていないユーザが存在していることが理由として挙げられる．

4.2　能力予測の評価

能力予測に関する実験では，Precision@10 による評価および nDCG による評価の二つを行う．

4.2.1　機械学習の評価

まず，機械学習を用いて素性を複合的に用いることにより能力予測の精度が高くなることについて実証するために，予測モデルを情報検索の分野で用いられる Precision@10 により評価する．対照手法として，公開レポジトリ数のみ (`public repos`)，フォロワー数のみ (`followers`)，s-index のみ (`s-index`) の 3 つを考え，SVR を使ってそれぞれの素性を組み合わせたものを `original` とする．

実験結果を図 6 に示す．この図 6 から，SVR で素性を組み合わせる手法が最も精度が高いことがわかる．

4.2.2　情報統合の評価

oDesk のみの素性，GitHub のみの素性，両方を統合した素性のそれぞれを用いた場合のモデルを比較し，情報統合が制度に貢献していることを明らかにする．

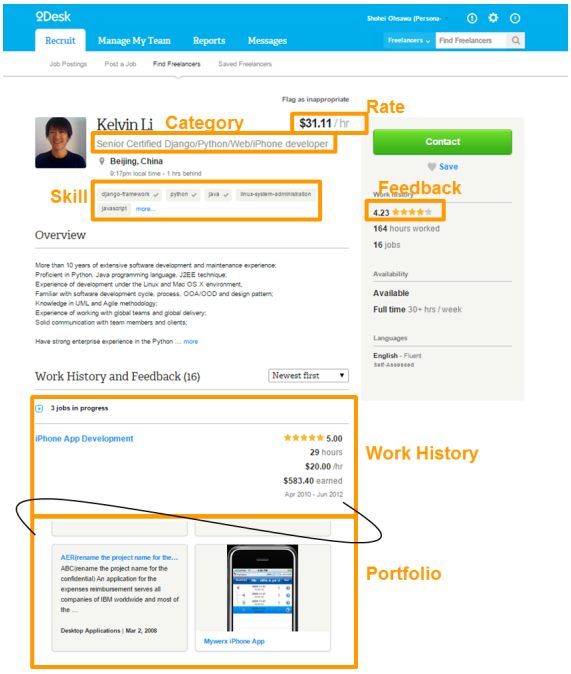

図 4　oDesk のユーザプロフィール

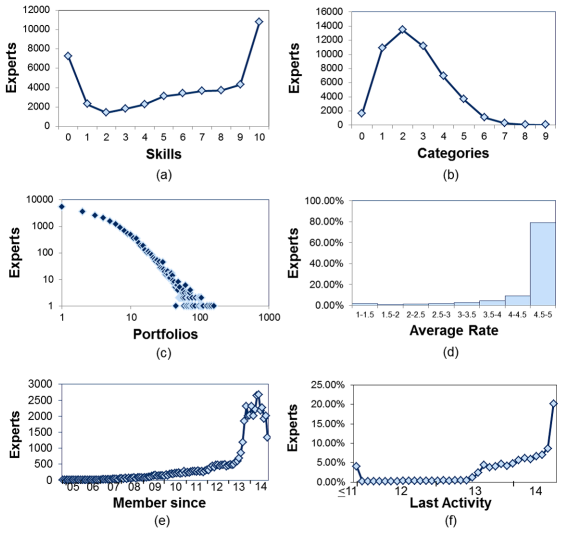

図 5　oDesk の基本的統計

評価指標として，情報検索の分野で広く用いられている，検索システムのランキングアルゴリズムの評価指標である**正規化減損累積利得** (normalized discounted cumulative gain; nDCG) [9] を用いる．

nDCG は $[0,1]$ の間の値を取り，1 は理想的なランキングが実現できている場合を意味する．検索システムは上位数件のみを表示することが多いので，上位 k 件のみの部分列を評価することがある．これを nDCG@k と書く．長さ k のランク順に整列された検索結果 $(u_1,\cdots,u_k)$ に対し，各ユーザ u_i の理想的な順位を r_i で表記し，$\mathbf{r}_k=(r_1,\cdots,r_k)$ を出力ランク列とする．この時，nDCG@k は，出力ランク列と理想的なランク列それぞれに対して，DCG と呼ばれる指標を計算し，前者の後者における割合を求める．すなわち，次式で定義される．

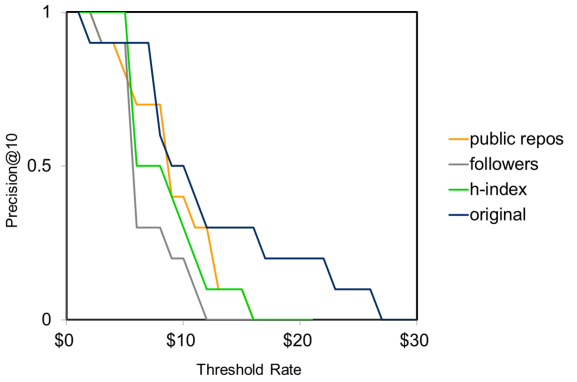

図 6　Precision@10 による各手法の評価

表 1　素性毎の精度の比較

Features	nDCG					
	@1	@5	@10	@30	@50	@100
OD	.515	.684	.701	.727	.755	.803
GH	.721	.739	.748	.767	.780	.807
GHP	.796	.785	.763	.782	.792	.813
OD+GHP	**.833**	**.805**	**.814**	**.824**	**.827**	**.856**

$$\mathrm{nDCG}(\mathbf{r}_k)\equiv\frac{\mathrm{DCG}(\mathbf{r}_k)}{\mathrm{DCG}(\hat{\mathbf{r}}_k)}.$$

$$\mathrm{DCG}(\mathbf{r}_k)\equiv\sum_{i=1}^{k}\frac{2^{\rho(r_i)}-1}{\log_2(i+1)}.$$

ただし，$\hat{\mathbf{r}}_k\equiv(1,\cdots,k)$ は理想的なランキングであり，$\rho(\cdot)$ は関連度を表す．本研究ではランクが小さいほど関連度が高いと考え $\rho(r)\equiv r^{-1}$ とする．

本手法では提案手法に加え，以下の対照手法の素性と比較を行う．すべて利用している学習器は SVM である．

- **oDesk のみの素性 (OD)**　クラウドソーシングから得られる指標のみを用いる．情報を統合しない最もシンプルな手法．
- **GitHub プロフィールのみの素性 (GH)**　OSS コミュニティのプロフィール情報のみを素性として利用する．s-index は利用しない．

これに対して，提案手法では次の二つを検証する．

- **GH に加え成果物の情報を利用 (GHP)**　GH に加え，成果物（レポジトリ）の情報も利用する．s-index を使った素性生成を行う．
- **OD と GHP の直和 (OD+GHP)**　二つの情報源から得られる素性の直和を取る．

実験結果を表 1 に示す．表 1 より，情報を組み合わせた場合に精度が高まるといえる．

5. 考　　察

5.1 プロフィールが与える印象への影響に関する実験

ここまでで，OSS コミュニティから得られる情報に s-index を加え素性を生成をしたものが最も予測精度が高くなることが明らかになり，OSS コミュニティの情報に基づきクラウドソー

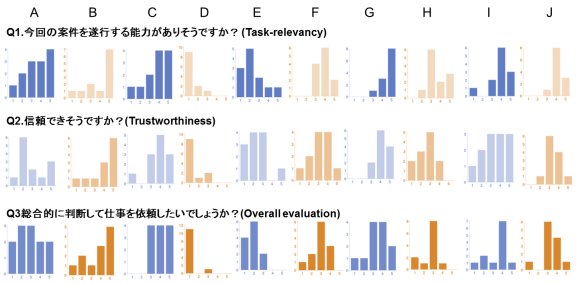

図 7　回答の全容

図 8　ユーザ B に関する回答（調査の結果 7 割が「最終的に仕事を依頼したい」と回答）

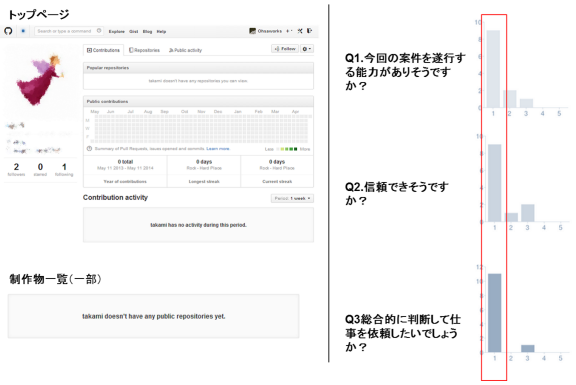

図 9　ユーザ G に関する回答（最も総合点が低かった技術者）

表 2　属性推定における信頼性・専門性・相互評価の RMS 誤差．太字は最も誤差が小さかったことを示す．

	専門性	信頼性	総合評価
Baseline	1.073	0.856	0.855
ST	0.998	0.517	0.875
SC	0.630	0.671	0.734
SC + ST	**0.268**	**0.315**	**0.602**

シングの能力予測が可能であることを示唆する結果となった．

一方で，発注者の立場が OSS コミュニティのプロフィールを目視で評価した場合に，いくつかの要素が印象を左右することも事実である．そこで，具体的にどのような要素が印象に左右するかについて，教師データに基づく予測モデルを構築し，考察を行う．

教師データは，被験者 13 人に対してアンケートを通して取得する．具体的には，10 人の GitHub からランダムサンプリングした技術者に対して，GitHub のユーザページをポートフォリオとして参考にしてもらう．

5.1.1　画面と評価値の関係

図 7 に，回答の全容を示す．人物によって十分なばらつきがあり，評価に関してもあまり割れることがないことがわかる．したがって，評価の高いユーザには，確固とした要因があると考えられる．これについて掘り下げていく．

まず，図 8 と図 9 はそれぞれ最も総合的能力が高かった人物と低かった人物の GitHub の画面を比較したものである．これを見ると，活動度が高い，すなわち多く活動しているユーザの方の評価が高いことがわかる．

5.1.2　予測モデル構築

実験結果を次の通り示す．これを見ると，専門性を予測する上ではプロフィールに基づく指標はあまり定数予測器からほぼ改善が見られないのに対し，ソーシャル指標のみを用いたモデルは精度の改善に成功してことがわかる．また，両方を用いるとさらに精度が改善する．これは総合評価にも同じ傾向が見られる．

次に，信頼性について見ると，プロフィールのみを用いたほうの精度が高い．これは，OSS コミュニティ上で友人数が多いだけではその人は信頼に足らないと被験者が判断しているからであると考えられる．信頼性に関してはソーシャル指標よりもプロフィール・実績指標の方が上回る形となったが，しかし両方を組み合わせて使う方が全体と比較して精度が高いことから，見る人は総合的に判断していることがわかる．

これにより，グラフ構造を考慮に入れた指標を用いることで，精度が上がっていると言える．

5.1.3　相関分析

実験結果である予測モデルの構築では，ソーシャル指標を加えると精度が上がることがわかったが，具体的にどのような素性が寄与していたのか相関分析を行なうことで明らかにする．

専門性・信頼性・総合評価をそれぞれポートフォリオの内容に基づきモデリングすると表 3 のようになる．

これらを見ると，評価指標ごとにそれほど差はない．すなわち，潜在する成分が別にあると考えられる．レポジトリ数の対数がすべてにおいて最も相関しており，第一印象に強く作用すると考えられる．CSS のレポジトリ数は（Web エンジニアとしての）専門性・信頼性・総合評価 3 つと高く相関している．実名でポートフォリオを作ったり，ブログへのリンクを貼ると発注されやすくなったりする可能性が上がる．

5.2　主成分分析による能力の分解

ソフトウェア開発者の能力やそれがプロジェクトに及ぼす影響について評価した文献がある．Crowston [6] らは，オープンソースソフトウェアの一つである SourceForge の分析を行い，プロジェクトの成功がソフトウェア開発者のプログラミング言語の習得度に依存することを示している．

能力に関して評価を行う研究として，コンピテンシーに関するものがいくつかある [13], [14]．Mccelland らは，知識や短期的な能力以外に長期的なパフォーマンスを発揮する汎用的

表 3　各指標に寄与する素性．上位 9 件のみを掲載し，それ以外は省略している．

	専門性	信頼性	総合評価
Blog	.46	.48	.45
RealName	.67	.57	.62
social:log-public_repos	.74	.79	.69
social:log-public_gists	.74	.79	.69
social:log-followers	.62	.45	.43
social:log-following	.68	.64	.58
lang:CSS	.50	.53	.42
lang:JavaScript	.42	.55	.48
lang:Perl	.37	.36	
lang:PHP			.28

な能力があるとし，コンピテンシーと呼んでいる [13]．中でも Spencer らの氷山モデル [14] について言及する．氷山モデルは，行動と能力に関して区別をしたものであり，観測可能な行動の下に，知識や能力があり，さらにその下に性格や資質があるというものである．こうしたモデルの中で，能力は短期的な能力（スキル）と，長期的な能力（コンピテンシー）があるとされ，明確に区別される．本研究が評価するのはあくまでスキルであり，コンピテンシーの評価は今後の課題である．

本研究では能力予測について扱ったが，一口に能力といっても多岐にわたる．実際，oDesk では発注者のフィードバックは，「スキルの高さ (Skill)」「成果物の質の高さ (Quality)」「対応可能な時間帯の長さ (Availability)」「締切を守るか (Deadline)」「コミュニケーション力の高さ (Communication)」「協調性の高さ (Cooperation)」の 6 種類の指標を総合したものである．これらの指標は潜在的には指標かが難しい本人の能力が顕在化したものであると考え，これらをを主成分分析することにより，潜在する要素について解釈を行う．

能力の潜在要素発見を oDesk に対して行ったものを表 4 に示す．具体的には，主成分分析を用いて，どの程度貢献しているかについて分析を行った．これを観察すると，第一主成分として総合的な能力が，第二主成分として能力の偏りが抽出されている．第二主成分では，正負の成分として，(Skills, Quality), (Availability, Deadlines, Communication, Cooperation) がそれぞれ挙げられるが，前者は専門性に関する潜在要素，後者は信頼性に関する潜在要素であると解釈できる．

これを確かめるため，表 5 のように相関分析を行った．これを見ると，Skill と Quality は 0.95 と，すべての組み合わせの中で相関が最も高い一方で，他の成分とは比較的相関していないことがわかる．

5.3　有効範囲と限界

本研究では oDesk と GitHub を使って実験を行ったが，本手法は一般的なクラウドソーシングならびに OSS コミュニティに適用することができる．各サービスに特化した指標を使っていないためである．したがって，Amazon Mechanical Turk におけるワーカーの評価にも応用することが可能であると考えられる．

表 4　能力値の主成分分析結果．PCi は i 番目の主成分を表す．

	PC1	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**
σ	2.1	.6	.4	.3	.3	.2
Skills	+.4	+.6	+.1	+.2	0	+.7
Quality	+.4	+.6	+.1	+.1	−.1	−.7
Availability	+.4	−.5	+.3	+.7	+.3	0
Deadlines	+.5	−.3	+.5	−.7	−.1	0
Communication	+.4	−.2	−.6	+.1	−.6	0
Cooperation	+.4	0	−.5	−.3	+.7	0

表 5　能力値の相関分析結果

	S.	**Q.**	**A.**	**D.**	**Com.**	**Coo.**
Skills						
Quality	.95					
Availability	.73	.75				
Deadlines	.75	.78	.90			
Communication	.82	.83	.80	.83		
Cooperation	.84	.84	.79	.80	.88	

一方で，本研究にはいくつかの限界がある．それは次の通りである．

- **欠損値を評価できない**　本研究は基本的にウェブ上に存在する情報に基づき人物の素性を抽出している．しかし，仮に対象サービス上に情報がない場合に，それが本人の特性によるものなのか，単に対象サービス利用していないだけなのかの評価は難しい．これは二人の人物がいたときに，それぞれを公平に評価することを難しいことを示唆している．そのため，本研究では行っていないが，OSS コミュニティに存在する情報量にあるレベルを設定し，レベルが一定以下であれば判別不能とすることも必要である．
- **特定性の低い人物は統合しにくい**　匿名化 [12], [15] の手法により特定性が低減したデータに関しては，統合を行うことが難しい．
- **グラフ上で 2 ホップ以上の関係を素性化していない**　本研究で提案した素性化の手法は基本的に 1 ホップ以内にある関係のみを素性化している．そのため，2 ホップ以上の関係を素性化していない．これには，隣接行列の多項式を素性として利用するなどの方法で素性化することが必要である．
- **プロフィールを非公開にしているユーザは取得できない**　プロフィールを非公開にしているなどの理由で検索をできないようにサービスに対して設定しているユーザは，検索 API を通して取得することができない．

これらの限界に関しては今後の課題となる．

6. 結　　論

本研究では，クラウドソーシング上のエンジニアの評価値予測問題を扱った．クラウドソーシング上だけでは得られないエンジニアの実績情報を予測するために，OSS コミュニティとエ

ンジニアの情報を統合する方法論について明らかにした．

実験では，oDesk と GitHub に対して提案手法を適用し，nDCG において提案手法が対照手法を上回ることが明らかになった．これは，クラウドソーシングと OSS コミュニティを統合することにより，属性推定を高精度化できることを示唆している．

得られた知見として，OSS コミュニティ上で評価されている人物はクラウドソーシング上でも高く評価されることがわかった．一方で，今回の手法はクラウドソーシングにおいて評価の高いエンジニアを予測するのには向いているが，一方で評価の悪いエンジニアの予測は難しいことが明らかになった．OSS コミュニティ上でスコアの悪いエンジニアは，単にその OSS コミュニティを使っていないだけである可能性もあるからである．このように，OSS コミュニティ上において成績が悪いことが，本人の能力の低さによるものなのか，データの不足によるものなのかを区別できないことは，OSS コミュニティを情報源としていることによる限界であると考えられる．

クラウドソーシングにおいて複数の応募者の中から一人を選別するという本研究の問題意識に立ち返ると，判別不能なエンジニアに対する評価は，優秀なエンジニアを取りこぼす損害を楽観的に見積もるか悲観的に見積もるかによって二つの立場に分かれる．前者の立場では，判別不能である候補者の採用は行わない．後者の立場では，判別不能である候補者に対して，テストなどの方法を使ってさらなる精緻な評価を行う．前者は性善説であり，後者は性悪説であるという見方もできる．

判別不能なエンジニアを減らすためにも，ネガティブな評価を OSS コミュニティ上から抽出することが課題である．今回，ポジティブな評価を抽出することは可能であることが示せたが，ネガティブな評価に関しては難しいことがわかったためである．これには発言のフィードバックを感情分析することなどの方法が今後の課題として考えられる．

文　献

[1] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50. ACM, 2006.

[2] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si. Expertise retrieval. *Foundations and Trends in Information Retrieval*, 6(2-3):127–256, 2012.

[3] A. Bozzon, M. Brambilla, S. Ceri, M. Silvestri, and G. Vesci. Choosing the right crowd: expert finding in social networks. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 637–648. ACM, 2013.

[4] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom. Expertise identification using email communications. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 528–531. ACM, 2003.

[5] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the trec 2005 enterprise track. In *Trec*, volume 5, pages 199–205, 2005.

[6] K. Crowston and B. Scozzi. Open source software projects as virtual organisations: competency rallying for software development. *IEE Proceedings-Software*, 149(1):3–17, 2003.

[7] S. Ghosh, N. Sharma, F. Benevenuto, N. Ganguly, and K. Gummadi. Cognos: crowdsourcing search for topic experts in microblogs. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 575–590. ACM, 2012.

[8] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences of the United States of America*, 102(46):16569–16572, 2005.

[9] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[10] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[11] T. Lappas, K. Liu, and E. Terzi. A survey of algorithms and systems for expert location in social networks. In *Social Network Data Analytics*, pages 215–241. Springer, 2011.

[12] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[13] D. C. McClelland. Testing for competence rather than for" intelligence.". *American psychologist*, 28(1):1, 1973.

[14] L. M. Spencer and P. S. M. Spencer. *Competence at Work models for superior performance*. John Wiley & Sons, 2008.

[15] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[16] J. Tang, M. Hong, J. Zhang, B. Liang, and J. Li. A new approach to personal network search based on information extraction.

[17] J. Zhang, J. Tang, and J. Li. Expert finding in a social network. In *Advances in Databases: Concepts, Systems and Applications*, pages 1066–1069. Springer, 2007.

[18] 中田喜文 and 宮崎悟. 日本の技術者–技術者を取り巻く環境にどの様な変化が起こり, その中で彼らはどの様に変わったのか (特集日本的雇用システムは変わったか?–受け手と担い手の観点から). **日本労働研究雑誌**, 53(1):30–41, 2011.