The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

TurkScanner: マイクロタスクの時給推定

TurkScanner: Predicting the Hourly Wage of Previously Unseen Microtasks

斎藤 奨 *1 Chun-Wei Chiang*2 Saiph Savage*2 中野 鐵兵 *1 小林 哲則 *1
Susumu Saito Teppei Nakano Tetsunori Kobayashi

Jeffrey P. Bigham*3

*1早稲田大学 *2West Virginia University *3Carnegie Mellon University
Waseda University

In crowd markets, workers struggle to earn adequate wages by accurately gauging hourly wage of microtasks that they have not completed before. This paper explores how we might be able to predict the necessary working time (and thus hourly wage) of a previously unseen task based on data collected from prior workers completing other tasks. We collected 9,155 data records using a web browser extension, installed by 84 Amazon Mechanical Turk workers, and explore the challenge of accurately recording working time both automatically and by asking workers. Our predictive model, TurkScanner, was created using ∼150 derived features and was able to predict working time with high accuracy. Future directions include observing its effects on work practices, adapting it to a requester tool for better price setting, and predicting other elements of work (*e.g.*, acceptance likelihood, worker task preference, etc.)

1. はじめに

クラウドソーシングにおけるマイクロタスクの作業時間推定による時給計算手法を設計・評価し，Amazon Mechanical Turk で収集したマイクロタスクの 84.3%において 100%以内誤差の時給予測精度を達成した．

マイクロタスク型クラウドソーシングにおいて，金銭報酬を主な動機とするクラウドワーカーの多くが十分な賃金を獲得できていないことが問題視されている [Irani 13]．主な理由の一つとして，ワーカーへ提示されるマイクロタスク情報の少なさが挙げられる．一般に，ワーカーが仕事を探す際は，公開されたマイクロタスクのリストから自身の好きなものを選ぶことができる．しかし，ここでワーカーに対し開示される情報は，短いタスク説明文やサンプル UI のみの限られた情報である場合が多い．そのため，これらを基に見積もったタスクの所要時間とリクエスタが任意に定めた報酬額とを照らし合わせながら「割の良い」タスクを探して適切に選択することは，多くのワーカーにとって困難である．

こうした問題を受け，近年の研究において最適なタスクを選択する支援を行う取り組みが行われている．Amazon Mechanical Turk のワーカーの作業履歴からタスク滞在時間に基づき時給を計算し可視化する Crowd-Workers [Callison-Burch 14] や，賃金の獲得効率を最適化したタスク作業のスケジューリングを行う TurkBench [Hanrahan 15] などが提案されている．これらは「割の良い」タスク選択の支援に効果的である一方，いずれも過去行われたことのあるタスクに対してのみ有効であり，初めて投稿されたタスクに適用できないという共通点がある．

プラットフォーム上では絶えず新しいマイクロタスクが生成され続けているため，未知なタスクであっても時給が推定可能となることが重要である．経験歴の長いワーカーは，フォーラムサイトでの情報交換やプラグインの使用によって人気のリクエスタのタスクを素早く選択するスキルを持っており [Kaplan & Saito 18]，明らかに優れたタスクはすぐに無くなってしまう．そのため，未知のマイクロタスクにおける時間的な情報をワーカーへ新たに提供し，より良いタスク選択を支援可能となることが極めて重要である．

本論文では，他のワーカーによる他のマイクロタスクの作業情報からマイクロタスクの作業時間を推定し，そこからタスク時給を計算するシステム TurkScanner を提案する．TurkScanner の構築において対処した技術課題は 2 つ存在する．一つ目は，タスク作業時間の正解データの収集である．ワーカーはタスク作業中にしばしば外部サイト閲覧や休憩などの様々な振る舞いをするため，一つ一つのタスクの実働時間を計測することは困難であることが知られている [Bederson 11]．これに対し本研究では，複数の計測方法（自動 ×2 種類，手動 ×2 種類）により収集した作業時間候補の中から，最終的なラベルの決定をワーカーに委ねる方法をとった．これにより，信頼性の保証は完全ではないものの，マイクロタスクごとに作業時間を検出することが可能となる．この手法を基にデータ収集用のウェブブラウザ拡張機能を独自に設計・実装し，それをインストールした 84 人の Amazon Mechanical Turk のワーカーが最大 10 日間で観測した合計 9,155 タスク作業レコードについて，各種タスク情報・リクエスタ情報・ワーカー情報および作業時間ラベルのデータを収集した．二つ目は，収集したデータをもとにしたマイクロタスク作業時間（および時給）の推定である．Gradient Boosting Decision Tree (GBDT) [Friedman 01] により構築したタスク作業時間推定器の出力結果と報酬設定額からタスク時給の推定を行う検証実験の結果，TurkScanner は全テストデータの 69.9%において 75%以内の誤差率，84.3%において 100%以内の誤差率の推定精度を達成した．

2. 提案手法

TurkScanner では，まず機械学習によりマイクロタスクの完了に必要な時間（作業時間）を推定し，その後マイクロタスクに設定された報酬額から時給を計算する．本節では，まず最初にデータ収集における作業時間の計測アプローチについて述べ，次にデータ収集用ウェブブラウザ拡張機能の設計について述べる．その後収集データの分析結果を示したあと，最後に作業時間および時給を計算する TurkScanner の設計について述べる．

連絡先: 斎藤 奨，早稲田大学，東京都新宿区大久保 3-4-1，03-5286-3000，susumu@pcl.cs.waseda.ac.jp

2.1 作業時間の計測

マイクロタスクの作業時間を正しく計測することは容易ではない [Bederson 11]．これはタスク作業中におけるワーカーの振る舞いの多様性に起因する．ワーカーは作業中，タスクの一環として外部サイトの閲覧や検索エンジンの利用など，ページを一時的に離れての仕事を依頼される場合がある．一方で，無関係なウェブサイトの閲覧やコンピュータを離れての休憩など作業を中断する場合や，複数の異なるタスクを並行して行うこともある．したがって，こうしたワーカーのあらゆる振る舞いを考慮し，それぞれに費やした時間を作業時間として計上するか否かを都度適切に判断しながら自動的にデータ収集を可能とする仕組みを構築する必要がある．

そこで，以下それぞれ異なる長所・短所を持つ 4 つのヒューリスティックな方法（**自動計測 1・2，手動計測 1・2**）によって同時に作業時間の計測を行い，最終的にワーカー自身に最も妥当なものを選ばせることで，多様な場面において正確な作業時間をラベル付けする方法を設計した．**自動計測 1** は，ワーカーがマイクロタスクを開始してから完了するまでの時間をプログラムにより計測する．この方法ではワーカーが何にも干渉されずにタスクを遂行した場合に最も正確な時間の計測が可能だが，途中で別タブにおける無関係な作業（他のタスクの同時作業や E メールチェックなど）が行われた場合などには正しい計測が行われない．**自動計測 2** では，ワーカーがマイクロタスクを完了するまでにページ上にフォーカスを持っていた時間のみを計測する．これにより別タブにおける他の作業での時間の除外が可能だが，仕事の一環として外部サイトへワーカーを誘導するマイクロタスク（ウェブ検索タスクなど）も一部存在し，そうしたタスクでの正確な時間の計測が難しい．**手動計測 1** では，作業時間計測状態の On/Off 切り替えを行うボタンをワーカーに使用させ，自ら作業時間に当たる時間を計測させる．この方法により，開発者が予期せぬいかなる例外においてもそれが作業時間に含まれるかどうかの適切な判断がワーカー自身によって可能であるが，人間であるワーカーが必ずボタン操作を行う保証をすることは極めて難しい．**手動計測 2** では，上記 3 つのうちいずれもワーカーが正しくないと判断した時，自らに時間を手入力させることで作業時間を決定する．

以上の候補の中から選ばれる最終ラベルに関して，いくつかの仮説を立てた．まず，最も選ばれる候補は手動計測 1 であると予想した．大半のワーカーは様々な理由から外部のページを閲覧したり，途中で席を一時離れたりすることを頻繁に行い，その時は自らの行動に基づいて適切に計測ボタンの On/Off を切り替えることで正確な時間を記録できると考えた．次に，2 番目に選ばれやすい候補として自動計測 1 を挙げた．これは，タスクを開いてからすぐに作業を開始し，かつ途中で休憩を挟まず完了したタスクにおいて，最も高信頼度な作業時間計測方法であるためである．しかし一方で，作業効率の向上を目指す一部のワーカーは，複数のタスクを複数のタブで開いた状態でそれらを一つずつ行うことがあると知られており [Kaplan & Saito 18]，この場合は正しい計測が行われない．よって，手動計測 1 よりも少ないと考える．一方，以上 2 つの計測方法で大半のケースはカバーできることから，自動計測 2 と手動計測 2 は数としては少ないと予想したが，ワーカーがタスクの途中でボタンクリックを忘れたりする場合の予防線としての意味を持つため，一定数はラベルとして選ばれうることが期待された．

2.2 ウェブブラウザ拡張機能を用いたデータ収集

作業時間ラベルの付いたマイクロタスク作業データを収集するために，Amazon Mechanical Turk (AMT) のワーカーにクローリング用のウェブブラウザ拡張機能をインストールさせ，サーバーへ作業データを送信する方法を用いた．

AMT 上でデータ収集に参加するワーカーを募るマイクロタスクとして，ワーカー情報（性別，年齢，世帯収入，ワーカー経験歴，一週間のワーカー労働時間合計など）を質問した後に拡張機能のインストールを行うページへ誘導するアンケート UI を設計した．拡張機能のインストール完了とともにワーカーは最長 10 日間データ収集作業が可能となり，いつでもアンインストールによって終了できる（参加終了までのデータ収集分の報酬は後に支払われる）．

拡張機能インストール後，ワーカーには通常通り AMT 上に存在する任意のマイクロタスク（AMT では “HIT” と呼ばれる）を選び，行うように指示した．一つの HIT 作業データを次の (a)∼(c) の 3 ステップにより収集し，(b) と (c) の両方を正しく完了するたびに 5 セントのボーナスを支払った:

(a) バックグラウンドデータスクレイピング: ワーカーが HIT のページを訪れるたびに，作業時間推定における特徴量として用いられる各種タスク関連情報をバックグラウンドで抽出した．表 1 に抽出した特徴量の一覧を示す．特徴量はタスク情報（`HIT`），ワーカー情報（`WKR`），リクエスタ情報（`REQ`），の 3 つのメインカテゴリで構成される．`HIT` は一回の HIT ページ訪問につき一度ずつ，合計 71 次元のタスクのメタデータや HTML に関連する特徴量がページ内から抽出される．`WKR` に属する特徴量には，ブラウザ情報およびワーカーのダッシュボードページに対する一日に一度の非同期リクエストのレスポンス内容，および拡張機能インストール時のアンケート回答の結果が該当する．`REQ` は，当該 HIT を発行したリクエスタの評判情報にあたる．Turkopticon[*1]，Turkopticon2[*2]，TurkerView[*3]，の 3 つの評判情報サイトが公開する API に非同期リクエストを送り，情報を取得した．

(b) ボタンクリックによる作業時間計測: ワーカーは，各 HIT での作業中に自らの作業時間を記録するように指示される．HIT の UI 上部には拡張機能によって描画された記録ボタンが表示されており，ワーカーはそれをクリックすることによって On/Off を切り替え，そのタスクにおける作業/中断状態を都度記録する．クリック忘れ防止のために 2 つの工夫を施した．まず，HIT 開始時にはページ中央部に黒半透明スクリーンのアラートが表示され，ボタンをクリックしなければタスクを始めることができないように設計した．さらに，ワーカーが作業時間の計測を開始した HIT のページには赤色の外枠を表示し，タイマーの稼働状況を把握しやすくした．なお，複数タブで開かれた複数の HIT におけるそれぞれのボタンは，同時に On にならないように制御されている（HIT の完全なマルチタスキングは可能でないことを仮定しているため）．

(c) HIT 作業後アンケート: HIT 作業レコードに作業時間をラベリングするために，ワーカーは複数の選択肢の中から最も近い作業時間を選択する．HIT を完了すると同時に新しくポップアップウィンドウが開かれ，自動 1（トータル），自動 2（フォーカス時），手動 1（ボタン），手動 2（直接入力），の 4 種類の計測作業時間のうち最も近い一つを選択するように指示される．前者の 3 つの選択肢のうちどれも該当するものが無いとワーカーが判断した場合，4 つ目の選択肢である TIME_CUSTOM の選択肢を選び，テキストボックスの中に作業時間を手入力する（X 分 X 秒の形式）．回答を選択した後，“Submit” ボタンをクリックすることで回答を送信し，ポップ

∗1 https://turkopticon.ucsd.edu/
∗2 https://turkopticon.info/
∗3 https://turkerview.com/

表 1: 収集データの特徴量カテゴリ一覧

HIT – タスク関連情報	
META	HIT メタデータ（報酬額，残りタスク数など；計 3 次元）
TMPL	HIT テンプレートの種類（AMT 提供のもの；11 次元の one-hot ベクトル）
URL	ページ内の URL 数（リンク，画像など；計 5 次元）
INP	ページ内の INPUT タグ数（*radio*，*text* などの type 属性ごと；計 18 次元）
TXT	ページ内の単語数（1 次元）
KW	タスク内容関連キーワードの有無（“survey”，“summarize” など；計 32 次元）
WKR – ワーカー関連情報	
PRFL	事前アンケートのワーカー情報（年齢，ワーカー経験，自身の推定時給など；計 16 次元）
EXT	AMT 関連拡張機能所持の有無（*CrowdWorkers*，*MTurk Suite* など；計 8 次元）
HIST	ワーカー作業履歴（回答承認率，承認 HIT 数など；計 4 次元）
REQ – リクエスタ評判情報	
TO	TurkOpticon（リクエスタの報酬設定や連絡などの 5 段階評価平均；計 6 次元）
TO2	TurkOpticon2（タスクの報酬設定やバグ有無などの 5 段階評価平均；計 34 次元）
TV	TurkerView（リクエスタの平均設定時給やレビュー数など；計 9 次元）

アップウィンドウは自動的に閉じられる．

2.3 収集データの分析

データ収集結果. 2018 年 10 月下旬の 10 日間でデータ収集を行った．84 人のワーカーから全 9,155 件の HIT 作業データを収集し，998 人のリクエスタによって投稿された 1,641 種類の HIT Group（同一テンプレートの HIT のバッチ）を含んでいる．ワーカーのデータ収集参加期間は平均 6.5 日間（SD=3.5; Median=8.1）で，一人あたり平均 109HIT（Min=1; Max=1,958; SD=238.1; Median=34）の収集が行われた．

上記データのうち一部を除外する処理を事後的に行った．まず，いくつかの低報酬（*i.e.,* 1〜3 セント/HIT）な HIT Group が一人のワーカーにより大量に行われていることを確認した．これはデータセットのバイアスの原因となりうるため，HIT 提出数の多い上位 3 位までの HIT Group のレコードを 4 位の HIT Group の提出数 (*i.e.,* N=208）に合わせてランダムに選択し，残りを除外した．加えて，スパムワーカーと思われる 1 人の回答と，不自然に長いまたは短い作業時間のタスク作業データ 210 個を無効なデータとして削除した．以上の結果，83 人のワーカーによる 7,608（83.1%）HIT が残った（1,587 種類の HIT Group，977 人のリクエスタ）．このデータの集合を以降の分析に用いる．

作業時間ラベリング. ラベル付けされた作業時間の平均は 277.9 秒 (SD=380.2; Median=148.3) であった．これらの作業時間ラベルと設定された報酬額から計算されたタスクの時給は，平均で$9.15 (SD=29.11; Median=4.23) となり，2,270HIT (29.8%)（674 (42.4%) の HIT Group）が米国の最低賃金 ($7.25/hour) であった．

ワーカーが最終的に作業時間として選択した候補の内訳は，自動 1（トータル）が 3,802HIT (50.0%)，手動 1（ボタン）が 2,525HIT (33.2%)，自動 2（フォーカス時）が 748HIT (9.8%)，手動 2（直接入力）が 533HIT (7.0%) であった．自動 2・手動 2 の総数が少ない点では仮説が支持された一方で，自動 1 が手動 1 を 16.8 ポイント上回った点では仮説に反する結果となった．

そこで，さらなる 2 つの分析を行った．まず 1 つ目に，手動計測 1 がうまく機能しなかった可能性を考慮し，手動計測ボタンの使用状況を分析した．その結果，N=7,037 (92.5%) の HIT でボタンが最低 1 回はクリックされていた上，ワーカーごとのボタン使用率は平均 95.8%で，これを下回ったワーカーはわずか 16 人 (19.3%) だった．ほとんどの場合でボタンは正しく使用されているように見えたが，一方で，そのうち 1 回タイマーが中止されたものはわずか 73HIT，2 回は 8HIT，3 回以上は 8HIT であり，それ以外の作業データ (N=6,948) では HIT の完了まで一度もタイマーが停止されていなかったことが分かった．2 つ目の分析として，作業時間ラベルとして最終的に手動 1 が選ばれた際の，自動 1 の計測時間の秒数との差を分析した．その結果，差がわずか 5 秒以内であるものが 72.5%，また 10 秒以内であるものは 85.1%に上った．これら 2 つの結果から，手動 1 が最終的なラベルとして選択された場合のうちほとんどが，自動 1 の計測時間とほぼ変わらない（*i.e.,* タスクが始まった直後にボタンをクリックし，かつ休憩を挟んでいない）場合に選ばれていることが分かる．したがって，今回の自動 1 と手動 1 は大部分が同じ意味を持つ結果であるといえる．なお，今回のデータ収集では作業時間計測中にワーカーが何をしていたかをトラッキングしていないため，手動 1 の計測結果の妥当性を検証することは出来ない．作業時間ラベルの妥当性は今後詳細に検証していく必要がある．

2.4 TurkScanner: マイクロタスク時給の推定

TurkScanner では，i) タスク関連情報から作業時間を推定し，ii) 推定作業時間とタスクに設定された報酬額からタスク時給を計算する．タスク作業時間の推定には Gradient Boosting Decision Tree (GBDT) [Friedman 01] を用いた．なお，今回は作業時間の出力値を対数値とすることで，作業時間の秒数が小さいほど誤差のペナルティの重みを与えた．

手法の評価として，ワーカーオープンな 4-fold クロスバリデーションを行った．ここでのワーカーオープンとは，学習データとテストデータをワーカー単位で分ける方針のことである．以降の分析では，それぞれのクロスバリデーションのペアにおいて，学習とテストを 50 回ずつ繰り返し，得られた結果の平均をとったものを分析対象とした．

3. 実験結果と考察

タスク作業時間予測. 図 1 は，実測の作業時間と GBDT を用いた回帰により推定した作業時間の関係を示したヒートマップである．この結果は，多くの割合の推定された作業時間がヒートマップの対角成分もしくはその近辺に属していることを示している．全体の推定作業時間のうち対角成分に属するのは 17.0%，隣り合ったセルを含めたとき 47.4%，2 つまで隣り合ったセルを含めた時 70.8%であった．

また全体の傾向として，作業時間が短い（60 秒未満）ほど推定作業時間は実際よりも長く，また作業時間が長い（600 秒以上）ほど推定作業時間は実際よりも短くなる現象が見られた．これは，データセット内の作業時間が対数正規分布に従っていること，および作業時間を対数で推定していることが原因として挙げられる．今回用いた GBDT は学習データ全体における予測誤差の総和を最小化するように最適化されているため，データ数の多い中程度の長さのマイクロタスクにおける作業時間推定が優先的に行われていると考えられる．

タスク時給予測. TurkScanner の最終出力であるマイクロタスク時給の予測結果について述べる．全てのテストデータ内のタスクにおける平均の予測時給は$5.21（SD=4.53; Me-

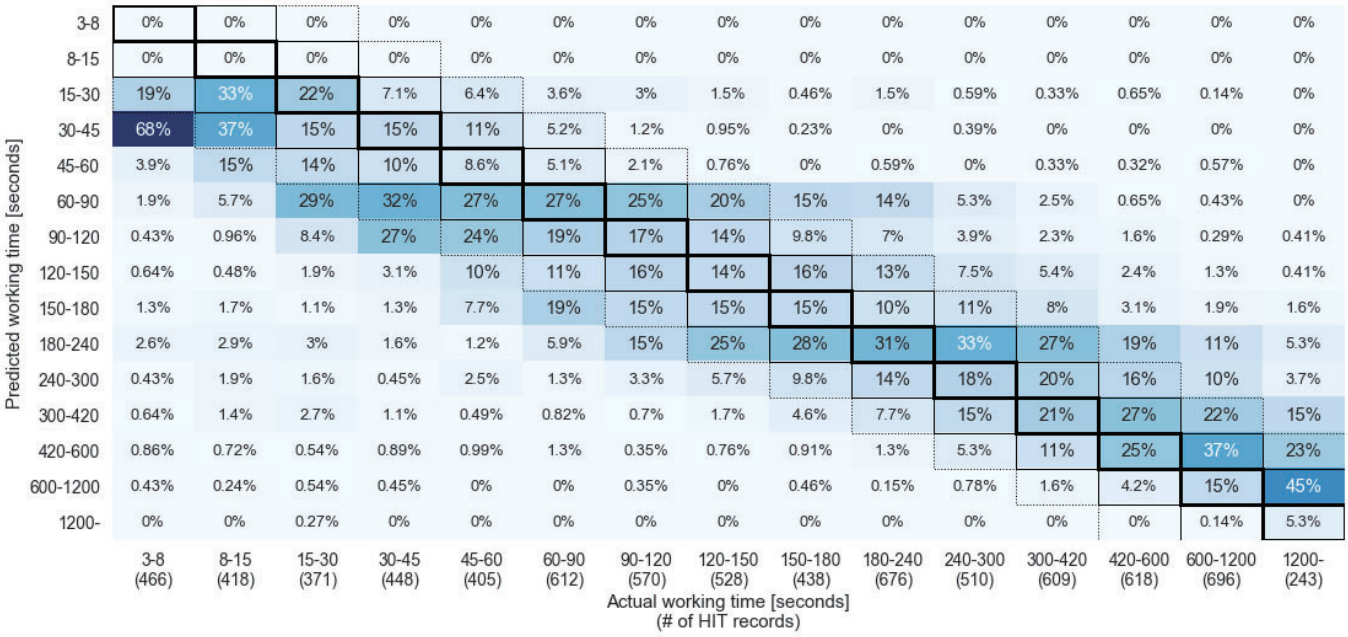

図 1: 作業時間推定結果のヒートマップ.

dian=4.20）であった．N=5,297（全データの 69.6%）において 75%以内誤差，N=6,412（全データの 84.3%）において 100%以内誤差をそれぞれ達成した．

また，タスクの実測作業時間に基づく時給別に結果分析を行ったところ，およそ\$15/h 以下のタスクにおける予測は一定の精度で行えている一方で，それ以上の時給額のタスクでは実際よりも低めに予測されていることがわかった．ここでうまく予測が行われなかったタスクについて目視の精査を行ったところ，これらのタスクは *(i)* 外部の URL へ誘導するアンケートタスク，または *(ii)* JavaScript によって動的に描画されるコンテンツを含むタスク，であったことが確認された．これらのタスクは静的な HTML 情報を多く持たないという共通の性質を持っている．今回実装した拡張機能は動的に描画されるコンテンツ情報を取得する機能を持たなかったため，こうしたタスクではページ内にコンテンツがほぼ何も含まれないものとして扱われている．一方で，タスク時給の決定にはタスクの難易度や作業量などのタスク内に顕れる情報が大きな影響を持つと考えられる．そのため，タスク内容の特徴量成分が失われているデータに関して，特に時給予測が機能しなかった可能性がある．このことを定量的に検証するため，今後，各特徴量毎の寄与度などのさらなる詳細な分析を行っていく必要がある．

4. 結論

過去に他のワーカーが行った他のタスクの作業情報に基づきマイクロタスクの作業時間を推定する手法を提案した．本研究では 2 つの課題に取り組んだ．一つ目は，マイクロタスク関連情報および作業時間ラベルを取得するデータ収集方法として，4 つの異なる方法により計測する作業時間のなかからワーカーに正しいものを選択させてラベリングを行うデータ収集用ウェブブラウザ拡張機能を設計した．二つ目は，GBDT による作業時間の回帰推定およびマイクロタスクの時給計算を行う TurkScanner を設計・評価し，ワーカーがマイクロタスクを開始する前に作業内容が報酬に見合うかどうかを事前に判断できる可能性を示した．今後は TurkScanner を AMT のワーカーに対して導入し，運用実験により実用性の検証を行っていく．

参考文献

[Irani 13] Irani, L. C., & Silberman, M.: Turkopticon: Interrupting worker invisibility in amazon mechanical turk. *In Proc. of the SIGCHI conference on human factors in computing systems*. ACM, 611620 (2013).

[Kaplan & Saito 18] Kaplan, T., Saito, S., Hara, K., & Bigham, J. P.: Striving to earn more: a survey of work strategies and tool use among crowd workers. *In Sixth AAAI Conference on Human Computation and Crowdsourcing*. AAAI (2018).

[Bederson 11] Bederson, B. B., & Quinn, A. J.: Web workers unite! addressing challenges of online laborers. *In CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 97-106 (2011).

[Friedman 01] Friedman, J. H.: Greedy function approximation: a gradient boosting machine. *Annals of statistics*. 1189-1232 (2001).

[Callison-Burch 14] Callison-Burch, C. Crowd-workers: Aggregating information across turkers to help them find higher paying work. *In Second AAAI Conference on Human Computation and Crowdsourcing*. AAAI (2014).

[Hanrahan 15] Hanrahan, B. V., Willamowski, J. K., Swaminathan, S., & Martin, D. B.: TurkBench: Rendering the market for Turkers. *In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1613-1616 (2015).