

ヒューマン コンピューテーションと クラウドソーシング

馬場雪乃（筑波大学）

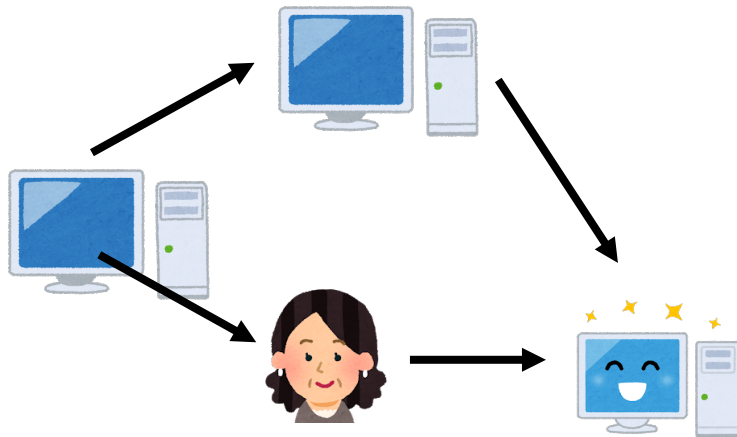
早稲田大学実体情報学博士プログラム コロキウム

2018年7月27日

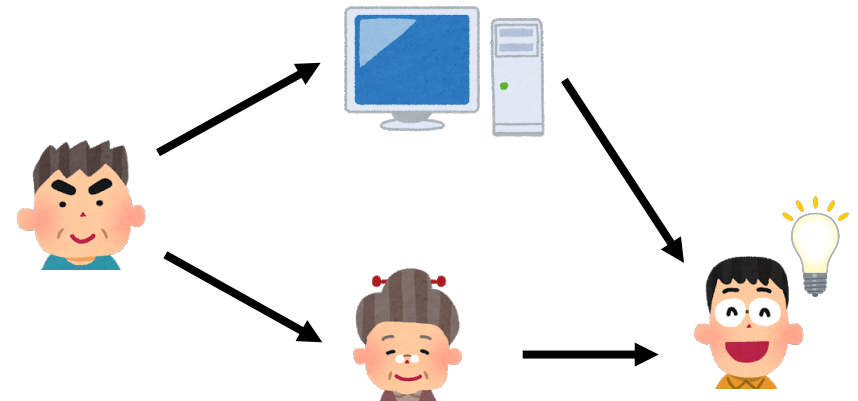
人間と人工知能の組み合わせで難しい問題を解決

- ヒューマンコンピューテーション：
人間と人工知能を組み合わせることで
どちらか一方だけでは解けない問題を解決

人工知能による問題解決を 人間が支援



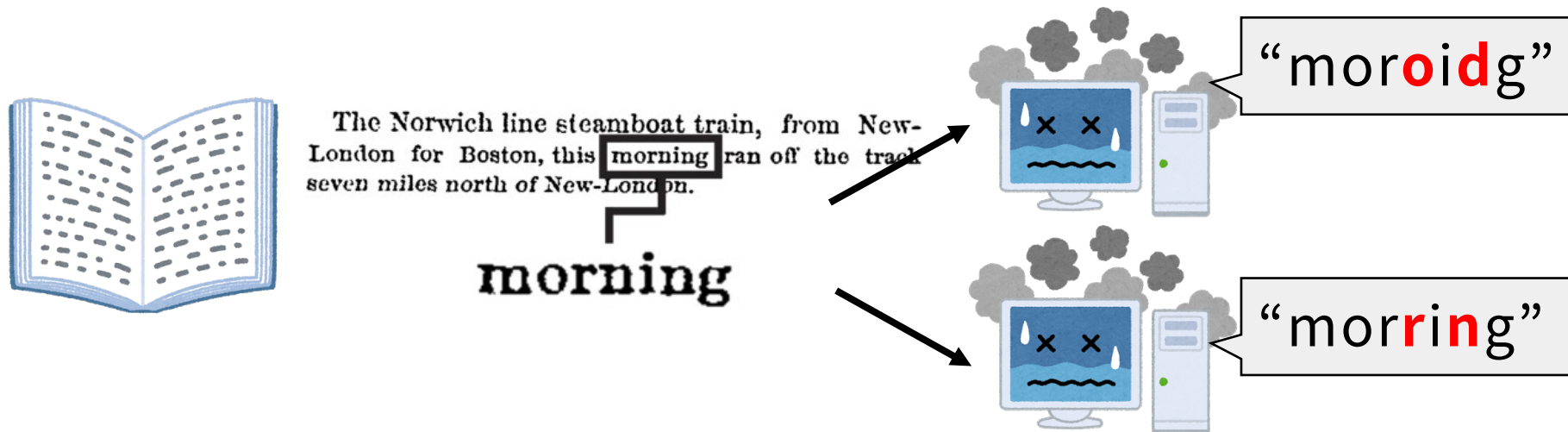
人間による問題解決を 人工知能が支援



ヒューマンコンピューテーションの例：reCAPTCHA

文字認識システムに人間を組み込む

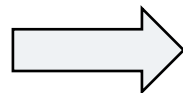
Step 1: 書籍中の文字を2つのOCRシステムに認識させる



Step 2: 結果が不一致のとき人間に問い合わせる



視覚障がい者支援システムに人間を組み込む



① ユーザが質問を投稿
(例：「コーンが入った缶はどれ？」)

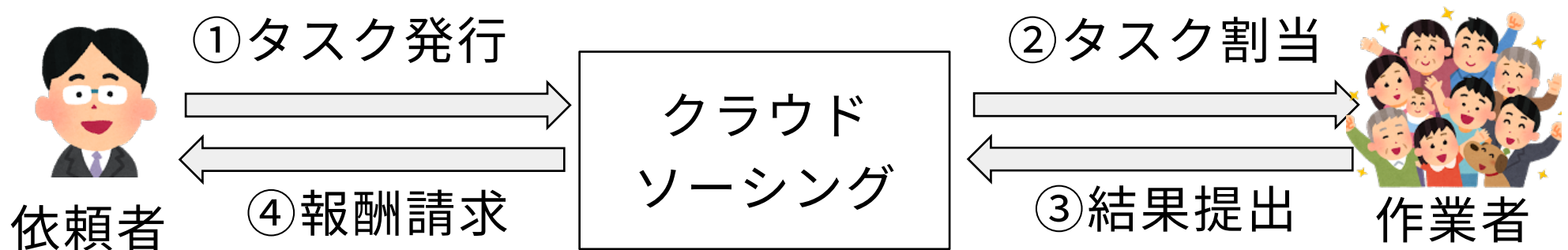
② システム内部
の人間が回答



クラウドソーシング

多数の人間にアクセスするための仕組み

- ヒューマンコンピュテーションでは多くの参加者が必要
- クラウドソーシング
 - インターネットを通じて人間に作業を依頼する仕組み
 - 例：Amazon Mechanical Turk, ランサーズ
 - 一部のサービスではAPIによる作業依頼が可能

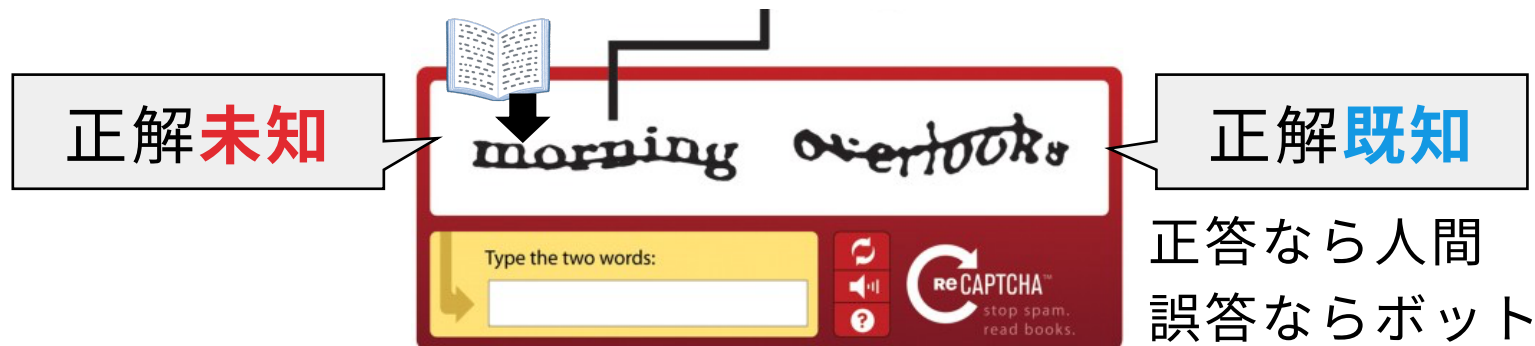


人間からどうやって正しい答えを引き出すか

- 人間が「常に・誰でも」正しい答えを返すとは限らない
 - 不確実性：人は様々な要因でミスをする
 - 多様性：人によって信頼性が異なる
- 人間から正しい答えを引き出すための工夫が必要

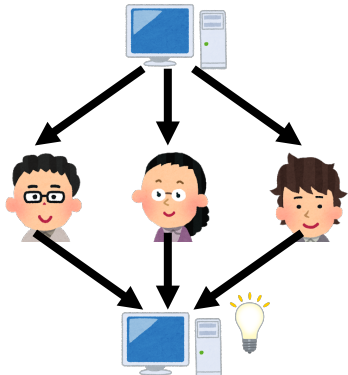

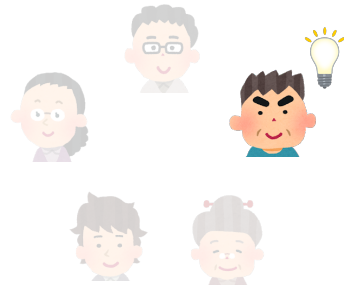
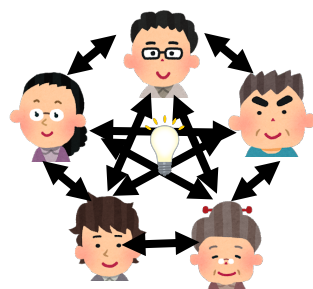
認証システムに組み込み正しい答えを引き出す

- ① **動機づけ**：文字認識作業を認証システムに組み込む
 - 回答者は「自分が人間だ」と示すため真剣に挑む
- ② **正解既知の問題の利用**：2つの文字列を提示する
 - 正解**未知**の文字列：認識したい文字列
 - 正解**既知**の文字列：人間かボットかの判定に利用
- ③ **並列化**：複数の「人間」が同じ答えを返したら採用



集団の力で正しい答えを引き出す

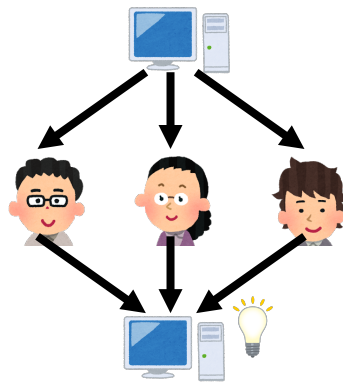
- 集団で問題を解かせる・解ける人を見つけることで人間から正しい答えを引き出す

人工知能による問題解決を人間が支援		人間による問題解決を人工知能が支援	
並列問合せ	直列問合せ	専門家発見	協調支援
			

簡単・単純な問題

困難・複雑な問題

① 並列問合せ



並列問合せ

別々に解かせて答えをまとめる

- 問い合わせ相手が一人だと
その人がミスをする と 正しい答えが得られない
- 複数人に問い合わせ て 回答を統合することで
正しい答えを引き出す
 - 簡単な手法：多数決



写真に鳥が写っていますか？

☐ YES ☐ NO



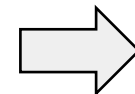
NO



YES










YES



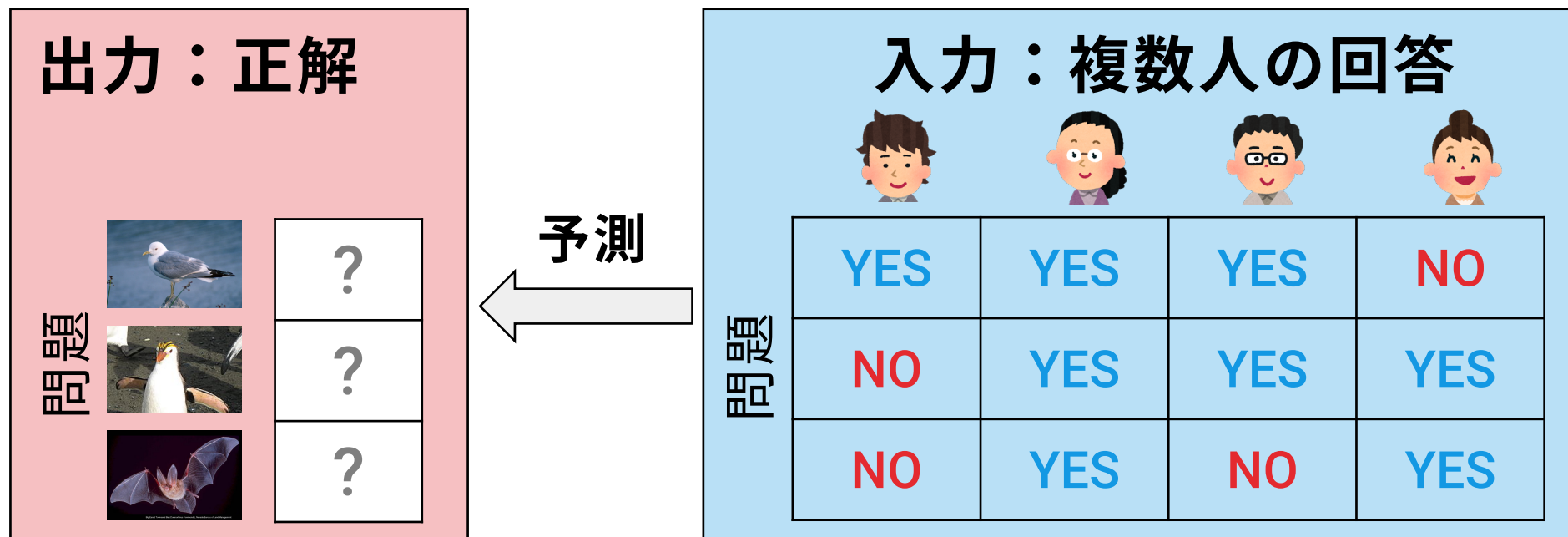
YES を
答えとして
採用

複数人の回答から統計的に正解を予測する

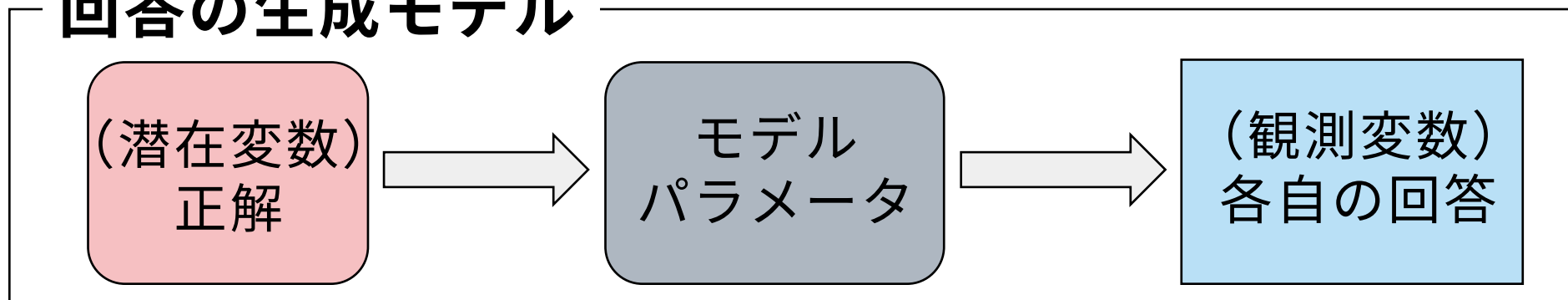
- 多数決よりも洗練された方法：統計的な回答統合
- 複数人の回答から正解を予測する問題として定式化
 - 生徒の回答だけから試験の正解を予測するようなもの

						正解
問題		YES	YES	YES	NO	?
		NO	YES	YES	YES	?
		NO	YES	NO	YES	?



回答の生成過程を統計モデルで表現

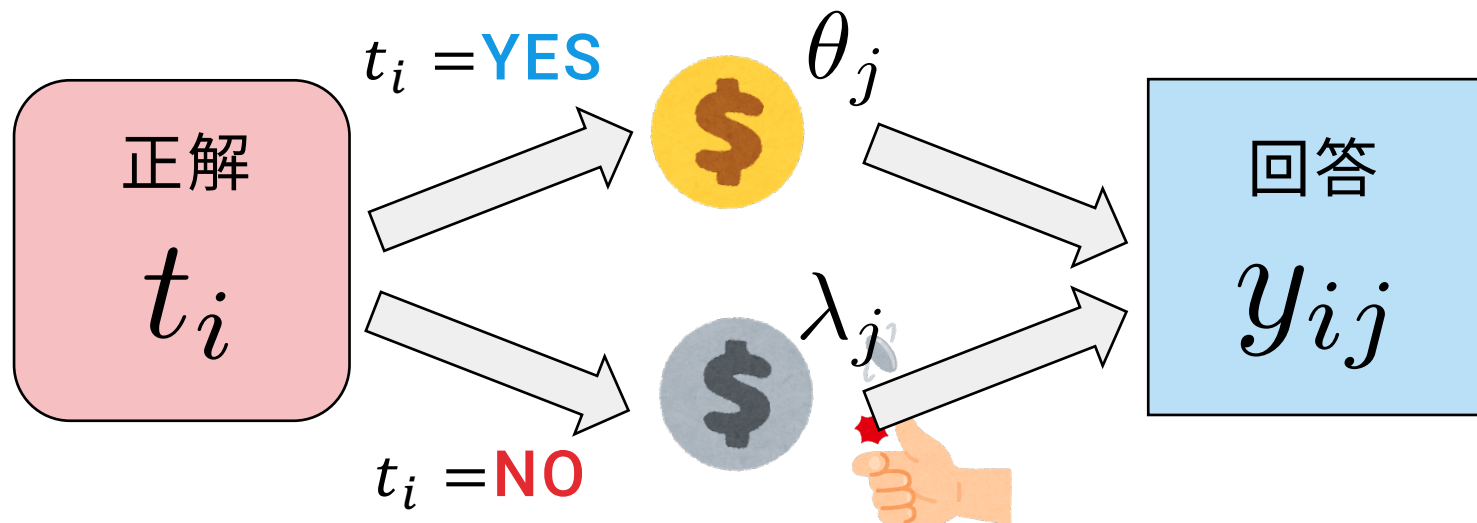


回答の生成モデル



回答者の能力を 2 種類のパラメータで確率的に表現

- 回答者の能力を 2 種類の確率で表現する：
 - θ_j : 正解が**YES**のときに正しく**YES**と答える確率 
 - λ_j : 正解が**NO**のときに正しく**NO**と答える確率 
- この確率を用いると次のような回答生成モデルが得られる



問題の正解と回答者の能力を交互に推定

- 正解を潜在変数としたEMアルゴリズムで正解と能力を交互に推定する

- Step 1: 能力を固定して正解を推定

$$q_i = \Pr[t_i = 1 \mid \{y_{ij}\}] \propto p \prod_j \theta_j^{y_{ij}} (1 - \theta_j)^{(1-y_{ij})}$$

正解

- Step 2: 正解を固定して能力を推定

$$p = \Pr[t_i = 1]$$

$$\theta_j = \frac{\sum_i q_i y_{ij}}{\sum_i q_i}, \lambda_j = \frac{\sum_i (1 - q_i) y_{ij}}{\sum_i (1 - q_i)}$$

正解がYESの問題での正答率（のようなもの）

回答者の能力と問題の難易度を推定し正解予測

- 「難しい問題で正答する回答者の方が能力が高そう」
→問題の難易度も考慮する



能力・難易度に基づく正答確率を表現

- 回答者と問題のパラメータを導入
 - θ_j : 回答者の能力
 - ν_i : 問題の簡単さ
- 回答者・問題のパラメータを用いて正答確率を表現し
正解と各パラメータを推定

回答 正解

$$\Pr [y_{ij} = t_i] = \frac{1}{1 + \exp(-\theta_j \nu_i)}$$

能力 簡単さ

能力が0 or 簡単さが0だと正答確率0.5

能力・簡単さが大きいほど正答確率が1に近づく

回答者に確信度も聞いて正解予測に用いる

- 回答者に確信度も聞く



- (1) 写真に鳥が写っていますか？ ○はい ○いいえ
(2) 回答に自信がありますか？ ○はい ○いいえ

- ただし確信度を正しく答える保証はない
 - 誤答なのに「自信がある」と答える人がいる
(**自信過剰**)
 - 正答なのに「自信がない」と答える人がいる
(**自信過小**)



確信度の生成過程もモデル化

- 確信度の傾向を確率で表現

「自信がある」と
答える確率

	回答がNO	回答がYES
正解がNO	$\pi_j^{(00)}$	$\pi_j^{(01)}$
正解がYES	$\pi_j^{(10)}$	$\pi_j^{(11)}$

- 確信度の生成モデルを導入し
回答生成モデルと組み合わせて正解を予測する

確信度

$$\Pr [c_{ij} \mid t_i = 0, y_{ij} = 1] = \left(\pi_j^{(01)} \right)^{c_{ij}} \left(1 - \pi_j^{(01)} \right)^{(1-c_{ij})}$$

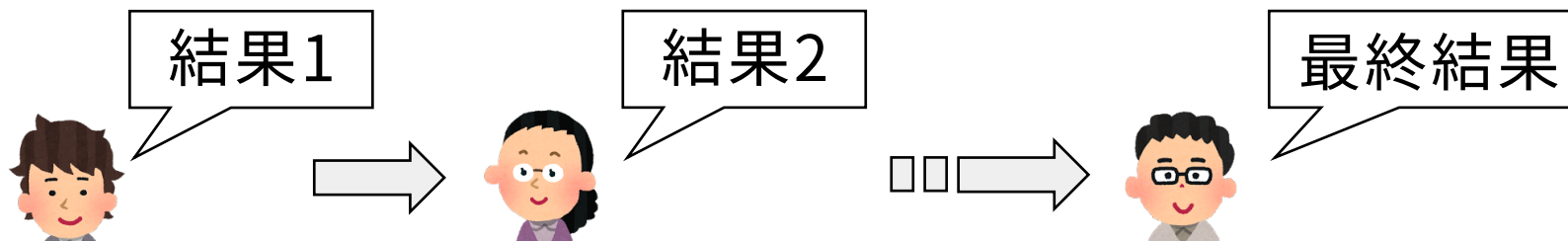
正解 回答

②直列問合せ



他人の答えを使って解かせる

- 他人の答えを使って解かせることで正しい答えを引き出す




- 例：ヒューマンコンピューテーションによる文書校正
Soylentは、Find-Fix-Verifyの3段階で校正を実施


- Find: 問題がある箇所の検出

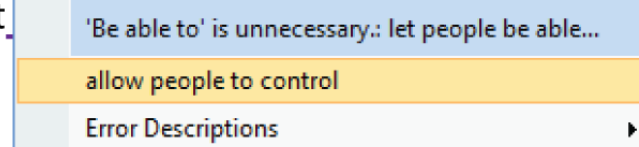
- Fix: 校正

- Verify:
校正誤りの検出

While GUIs le computers more intuitive and easier to learn, they didn't let people be able to control computers efficiently.



While GUIs le computers more intuitive and easier to learn, they didn't



他人の答えを提示し自己訂正を促す





- Step 1: 何も見せずに回答させる

「サンフランシスコの写真を選んでください」

				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

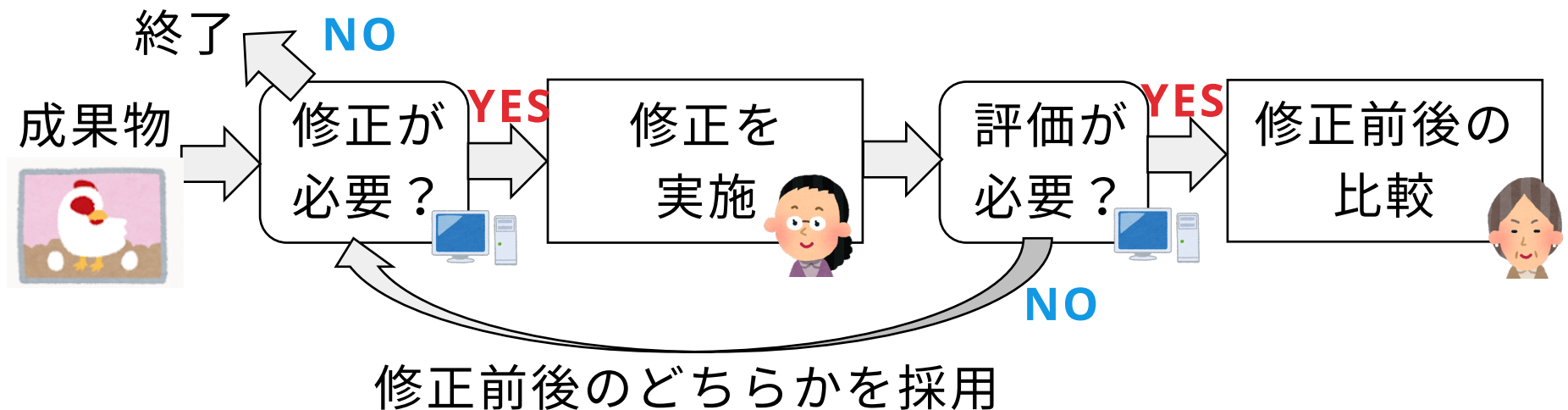
- Step 2: 他の回答者の統合解を提示し、再度回答させる

「あなたの回答が間違っていると思う場合は修正してください」

					
あなたの回答	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
他の人の回答		<input checked="" type="checkbox"/>			<input type="checkbox"/>

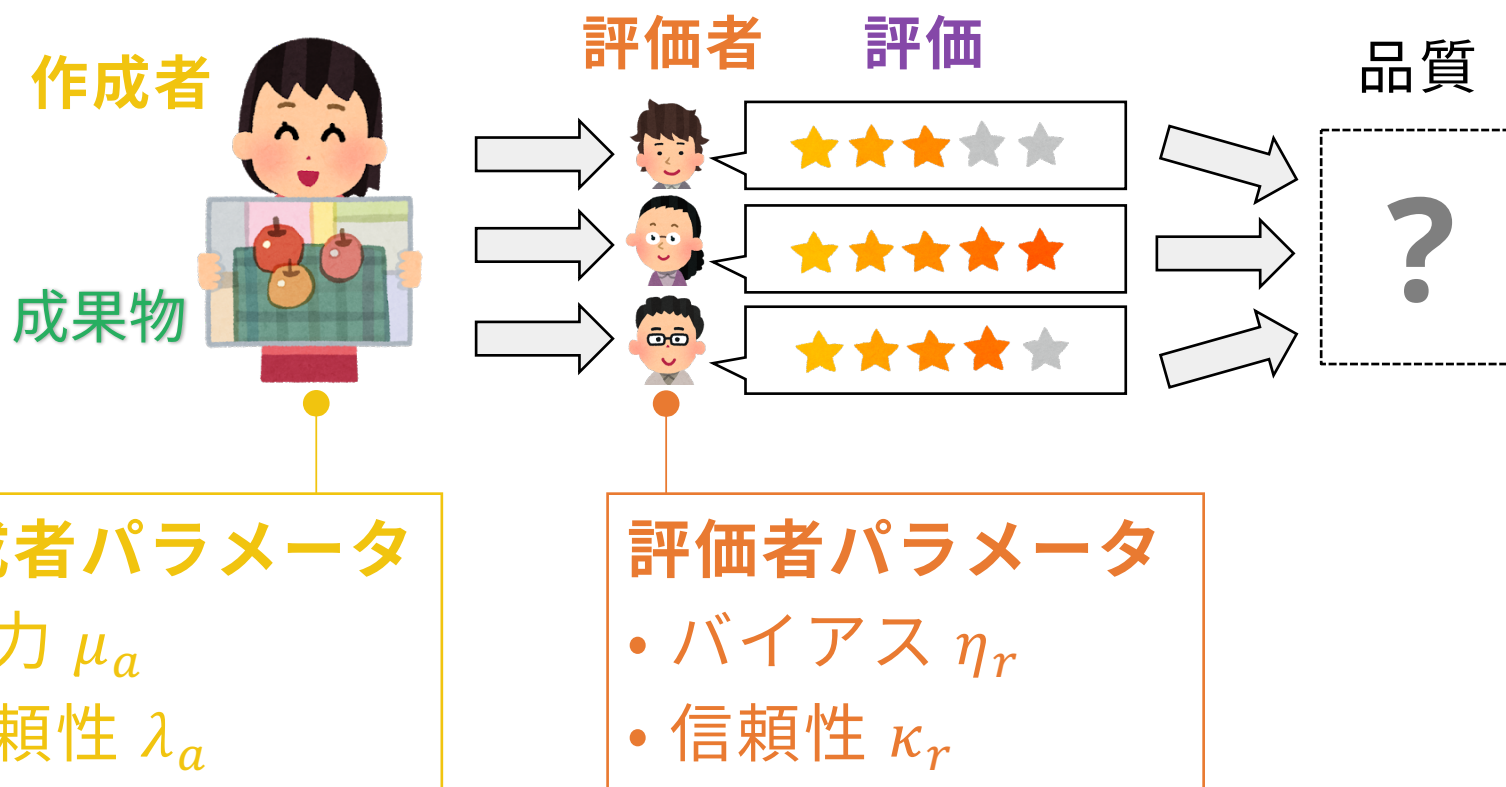
アルゴリズム制御により修正と評価を繰り返す

- TurKontrol: 成果物の修正と評価を繰り返すアルゴリズム
 - 修正・評価の要不要を
品質を状態とする部分観測マルコフ決定過程で決定
 - 十分に修正できたところで終了



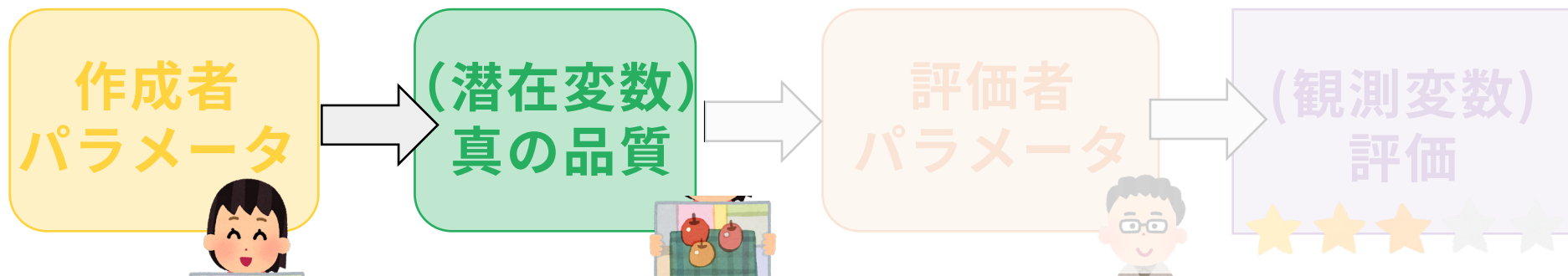
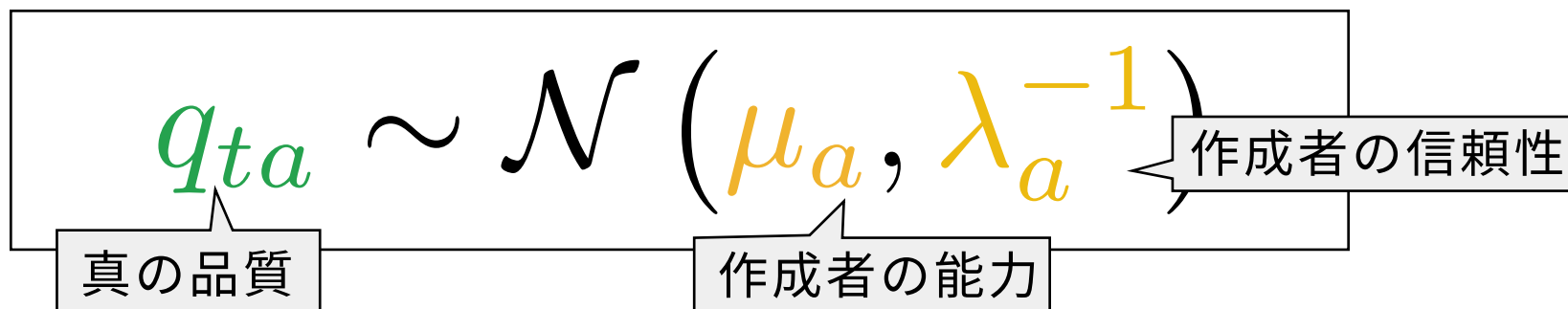
評価者能力を考慮し成果物の品質を予測する

- 評価が正しいとは限らない
- 評価者・作成者の能力を推定した上で成果物の品質を予測



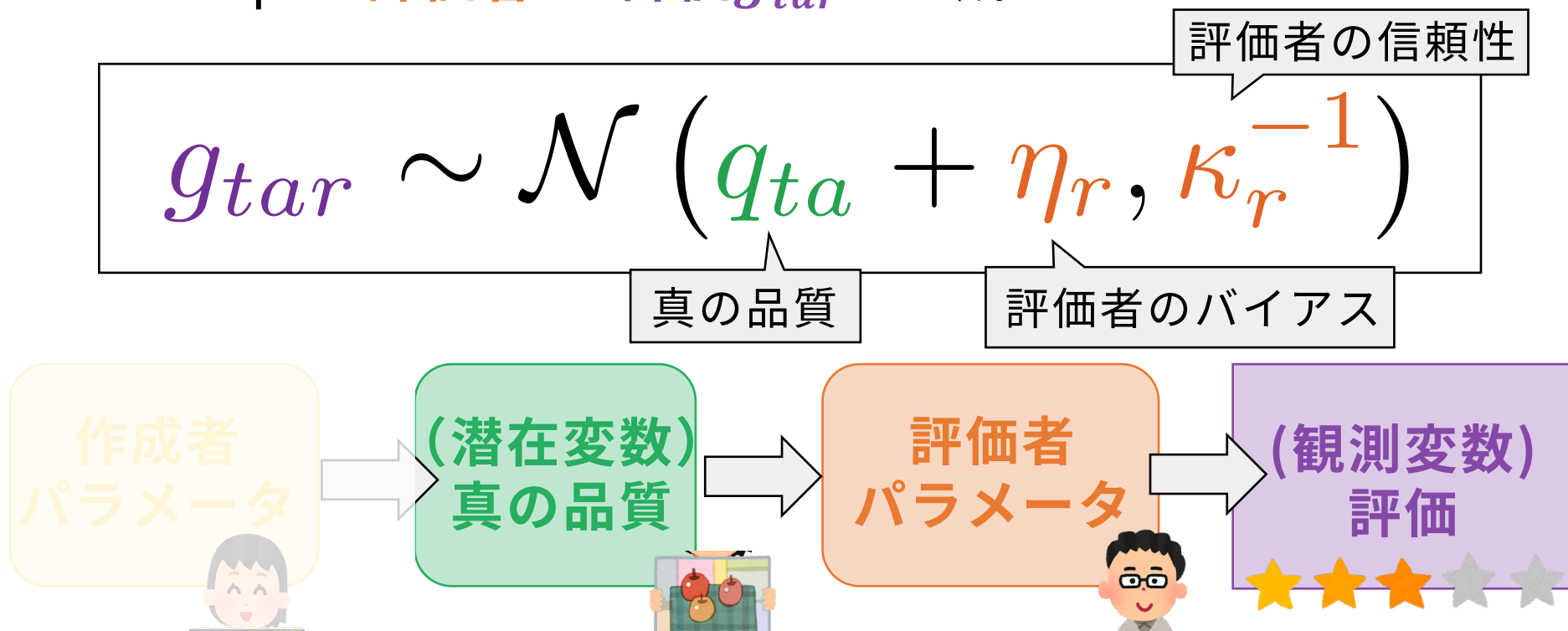
成果物と評価の生成過程をモデル化

- 成果物と評価の生成モデルを導入し
評価結果から成果物の品質を予測する
 - Step 1: 作成者 a が真の品質 q_{ta} をもつ成果物を生成

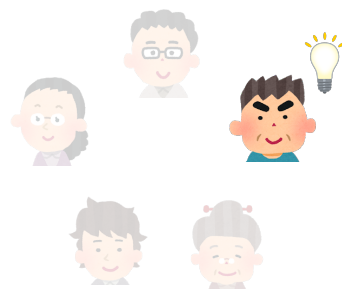


成果物と評価の生成過程をモデル化

- 成果物と評価の生成モデルを導入し
評価結果から成果物の品質を予測する
 - Step 2: 評価者 r が評価 g_{tar} を生成する



③ 専門家発見



解ける人を集団の中から見つける

- 専門性が必要な問題では
解ける人を見つけないと正しい答えが得られない
 - 例：医療に関する問題
- 簡単な方法：正解既知問題での正答率で専門家を見つける
- テストを用意し全員に受けさせるのは煩雑なので
他の手がかりを利用したい

専門性が必要な問題例

Q. 血中濃度を確認する必要性が最も高い医薬品はどれか？

(A) アスピリン

(B) フロセミド

(C) テオフィリン

(D) インドメタシン

専門家に共通する属性を手がかりとする

- 専門性が回答者の属性（学歴、学部等）に依存することはよくある

- 回答者の属性を利用して専門家を発見

- 回答者の属性ベクトルが与えられる

$$\mathbf{a}_j = (a_{j1}, a_{j2}, \dots, a_{jn})$$

医学部出身?

経済学部出身?

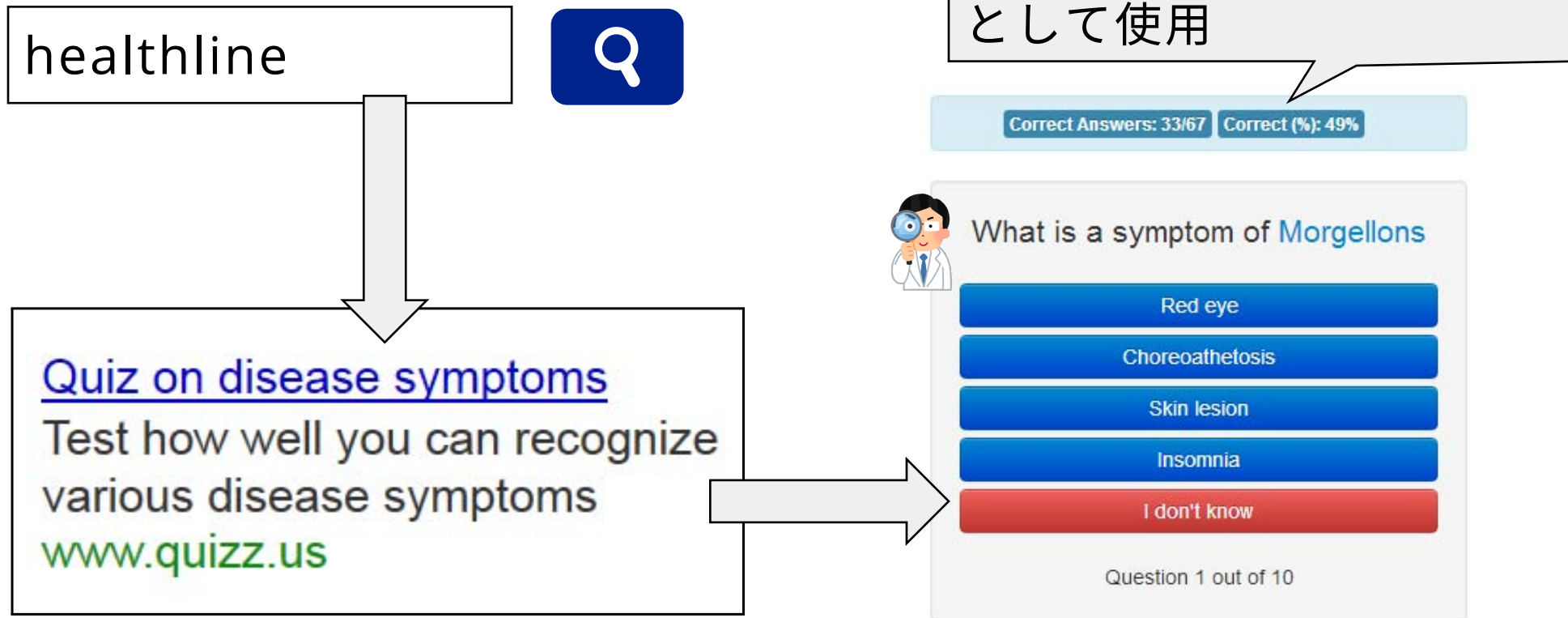
- 回答者能力を属性ベクトルの線形和で表現し各属性に対する重みを推定

$$\text{能力} \rightarrow \theta_j \sim \mathcal{N}(\mathbf{w}^\top \mathbf{a}_j, \sigma^2)$$

属性の重み

専門家がよく使う検索クエリを手がかりとする

- 検索連動型広告から問題へ誘導し
専門家がよく使う検索クエリを発見

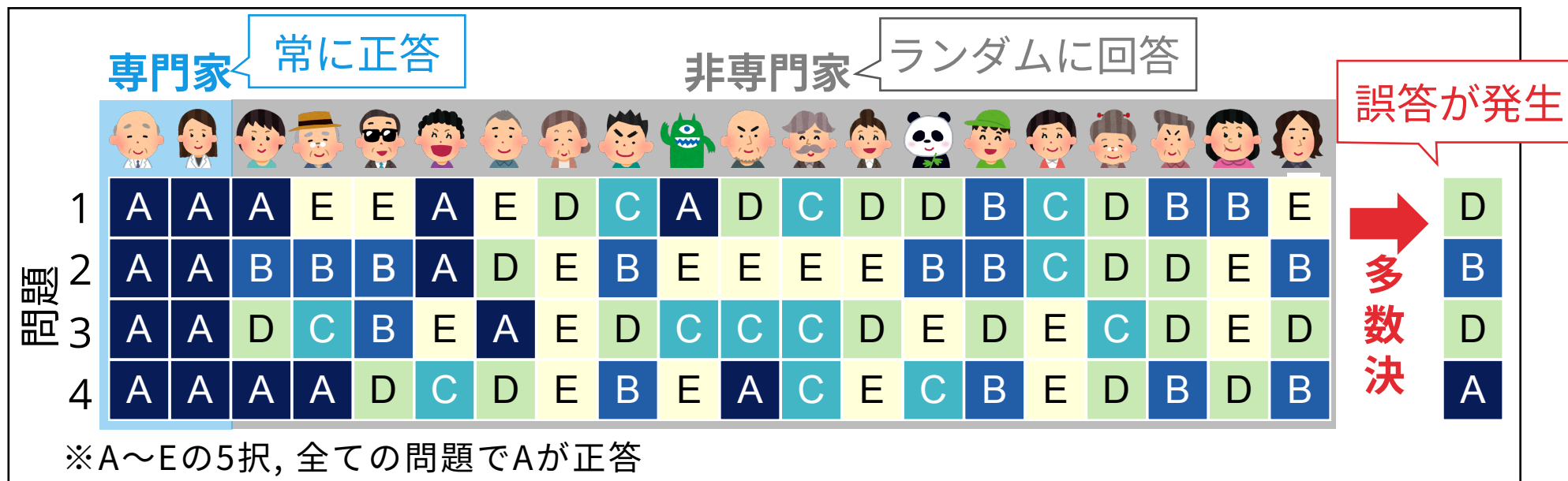


正解未知の問題を使った専門家発見

正解未知の問題に対する回答から専門家を見つける

- 正解を常に用意できるとは限らない
- 多数決解で代用したいが
専門家が少数派となる問題では多数決は失敗する

多数決が失敗する極端な例

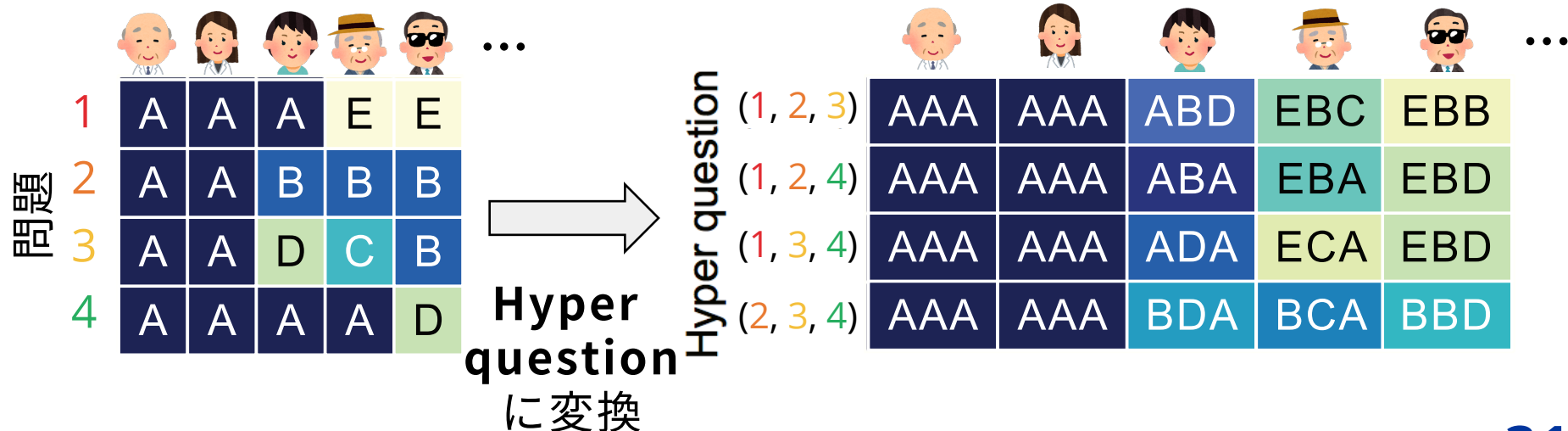


専門家同士の回答が一致することを利用した多数決

- 専門家同士の回答は複数問で一致しやすいが
非専門家同士の回答は一致しにくい
- 複数問をまとめて一つの問題とみなした、
Hyper question に対する回答で多数決を取る

専門家の回答は一致し
多数派となる

非専門家の回答
はばらつく



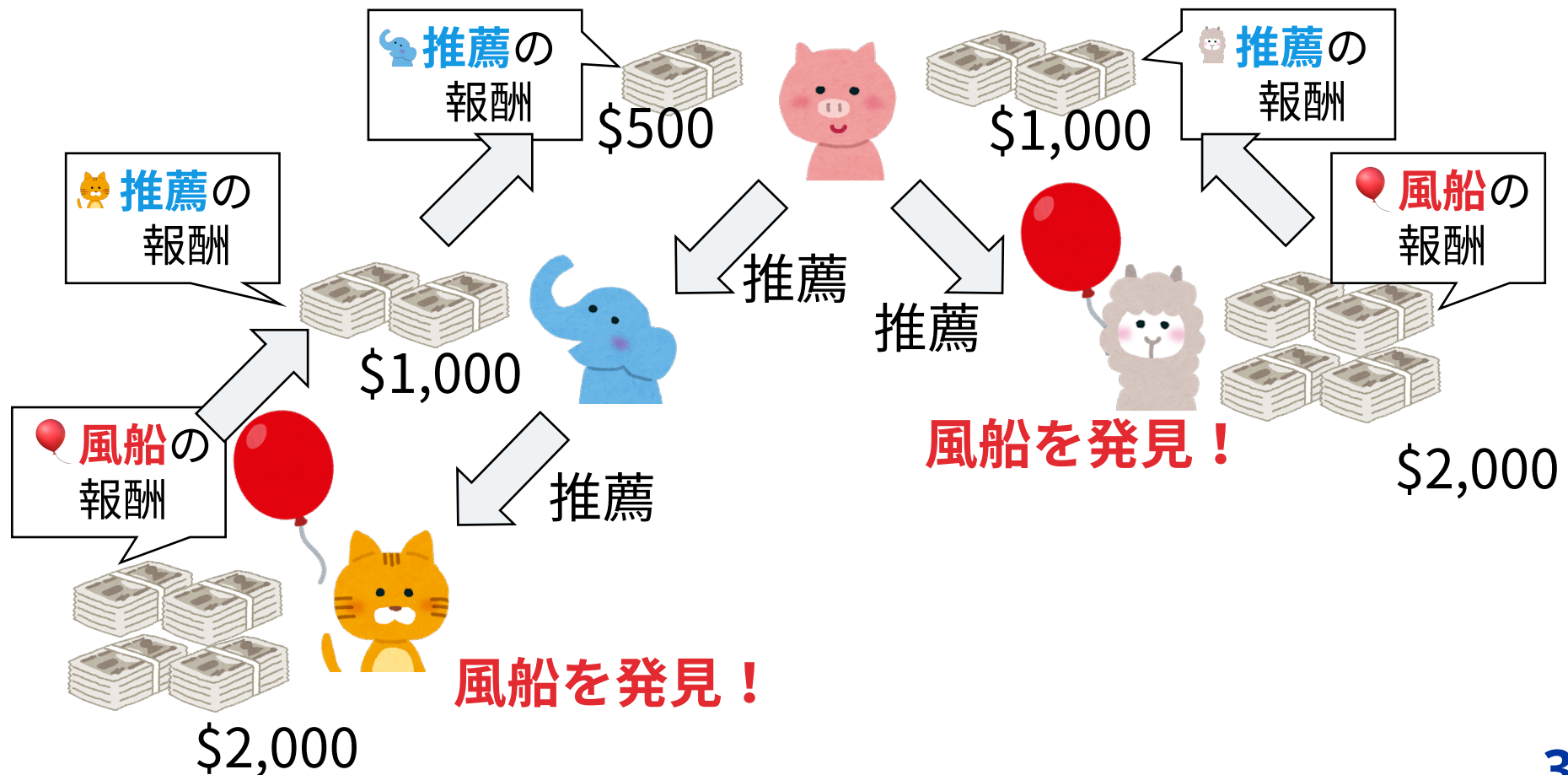
他人からの推薦に基づいて専門家を見つける

- DARPA Network Challenge
 - 全米10箇所に赤い風船🎈を設置
 - 最初に全ての風船を見つけたチームに賞金
 - 優勝チーム：
「風船を見つけそうな人」
を口コミで発見

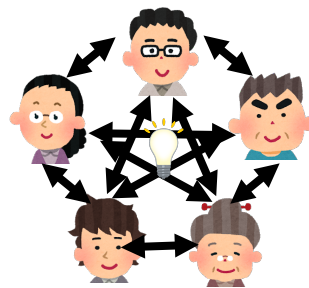


有益な推薦が得られるように報酬を設計

- 被推薦者が風船を発見した場合、
推薦者にも報酬が分配されるように設計



④ 協調支援



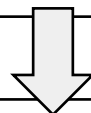
集団による問題解決を支援する

- 困難・複雑な問題の解決には人間同士の協調が不可欠
 - 問題を分割しみんなで解く
 - 良い解をみんなで見つける
- 言語による議論はスケールしないため
システムティックな協調支援が求められる

問題分割を集団で行い大規模な問題を解決する

- Turkomatic: 問題分割・解決を集団で行う仕組み
 - Step 1: 問題を分割する
 - － 十分単純な部分問題になるまで分割を繰り返す
 - Step 2: 各部分問題を人間が解く
 - Step 3: 部分問題の解を集約し最終的な解を出力

問題例：「『嘘も方便』は正しいか？」がテーマの小論文執筆



部分問題1: 「嘘はよくない」という段落を執筆

部分問題2: 「必要な嘘がある」という段落を執筆

...

確率的勾配降下法に倣って集団で解を探索する

- 解が数値で表される問題における協調最適化
 - ランダムに回答者を選ぶ
 - 回答者は一定範囲内の好みの場所に解を動かす
 - 範囲を狭めながら解の更新を繰り返す
- 様々なケースでの収束性が理論的・実験的に示されている

問題例：予算の決定



集団からアイデアを募集するプラットフォーム

- Innocentive:
特定の問題の解（アイデア）を募集するプラットフォーム
 - 問題例：
「フッ素パウダーを空気中に拡散させずに歯磨きチューブに入れる方法は？」
 - アイデアが採用されると賞金が支払われる



遺伝的アルゴリズムに倣ってアイデアを探索する

- 集団でアイデアの選抜・交叉を繰り返す

例：子供向けの椅子のデザイン

Step 1: 第1世代の個体を各自が生成



Step 2: 投票で優良個体を選抜し次世代の個体を交叉により生成

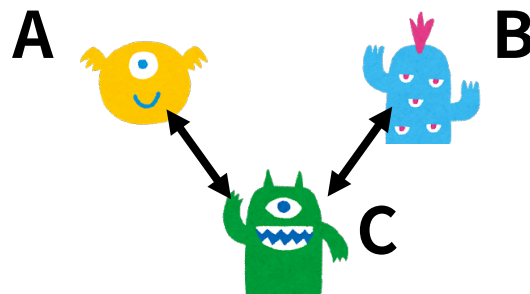


Step 3: 選抜・交叉を繰り返す

アイデア類似度の可視化

アイデアを見せ合うことで互いの発想を促す

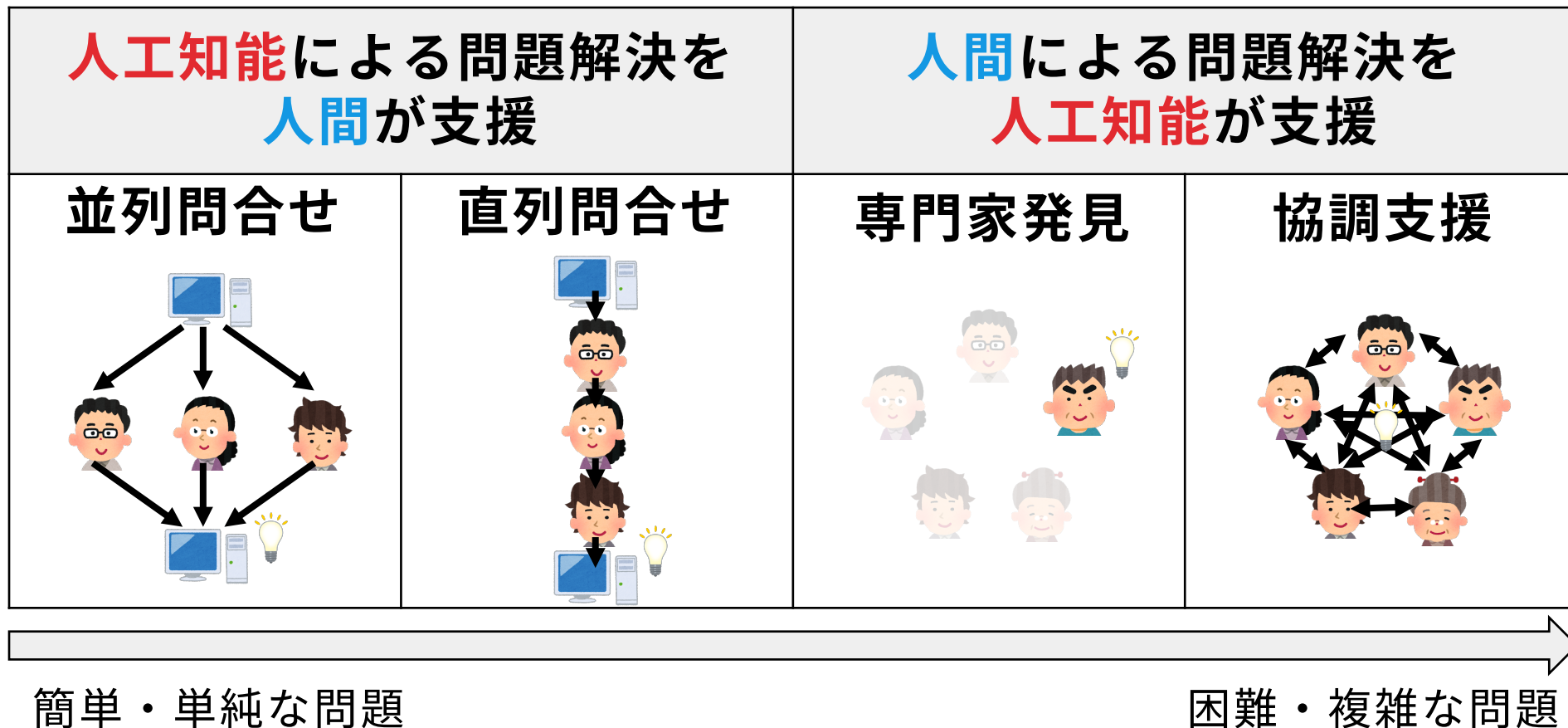
- アイデア類似度の可視化：
似たアイデアを近くに配置することで全体像把握を支援
- アイデア類似度も集団が判断
 - 相対的な類似度を尋ねる：
「AとBのどちらの方がCに似ている？」



まとめ



人間の集団と人工知能による問題解決の手法を紹介



高度で複雑な実社会問題への適用

- ヒューマンコンピューターションのグランドチャレンジ
 - ヒューマンコンピューターションによる研究開発？
 - － 映画「ロレンツォのオイル」：
素人夫婦が文献調査と専門家との議論により
息子の病気の治療法を発見
 - ヒューマンコンピューターションによる裁判員裁判？
 - ヒューマンコンピューターションによる政策決定？

