Vectoring in Research

CS 197 | Stanford University | Michael Bernstein

Administrivia

Next week: how to give a talk, by Prof. Kayvon Fatahalian

Time to dig in to your projects?

What problem are we solving?

"But how do we start?"

"I'm feeling so lost."

"I thought of an important reason that this won't work."

"It's not working yet. I'm not sure that we're making progress."

Today's big idea: vectoring

What is vectoring?

How do we vector effectively?

What goes wrong if we don't vector?

Bernstein theory of faculty success

To be a Stanford-tier faculty member, you need to master two skills that operate in a tight loop with one another.

Vectoring: identifying the biggest dimension of risk in your project right now

today

Velocity: rapid reduction of risk in the chosen dimension

not today!

What Is Vectoring?

What research is not

1. Figure out what to do.
2. Do it.
3. Publish.

What research is

Research is an iterative process of exploration, not a linear path from idea to result [Gowers 2000]

Problematic points of view

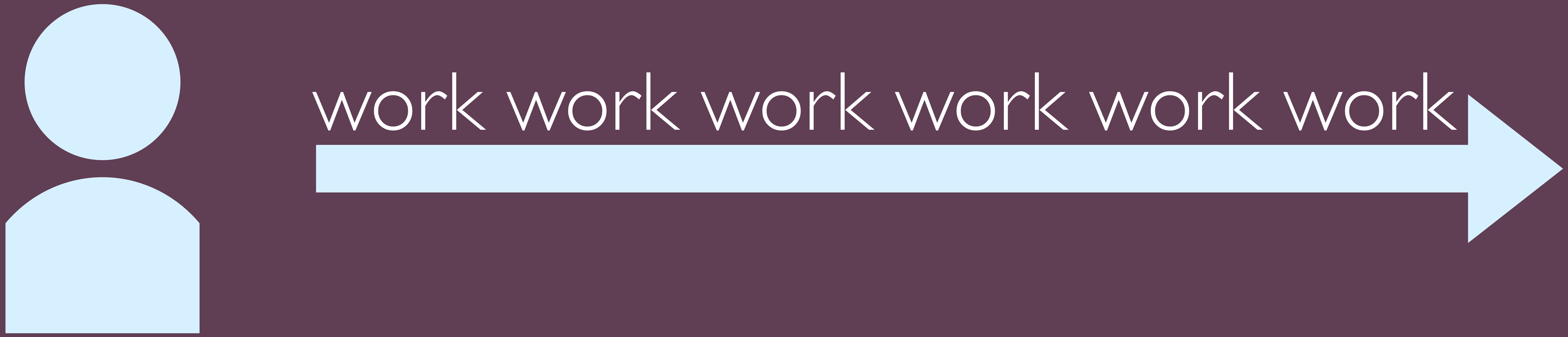
"OK, we have a good idea. Let's build it / model it / prove it / get training data."

"I spent some time thinking about this and hacking on it, and it's not going to work: it has a fatal flaw."

→ **Treating your research goal as a project spec and executing it**

Idea as project spec

Taking a concept and trying to realize it in parallel across all decisions, assumptions, and goals



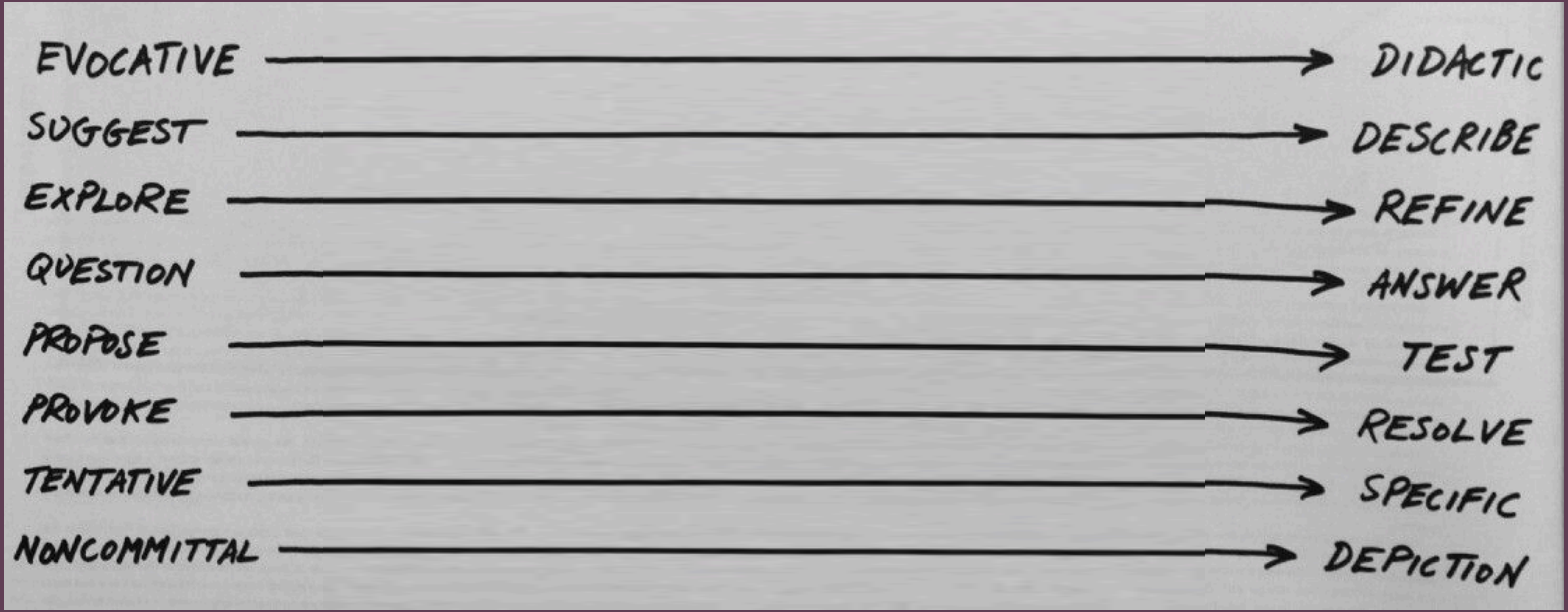
Concept

Result

Idea as project spec

What you should have done

What you did



[Buxton 2007]

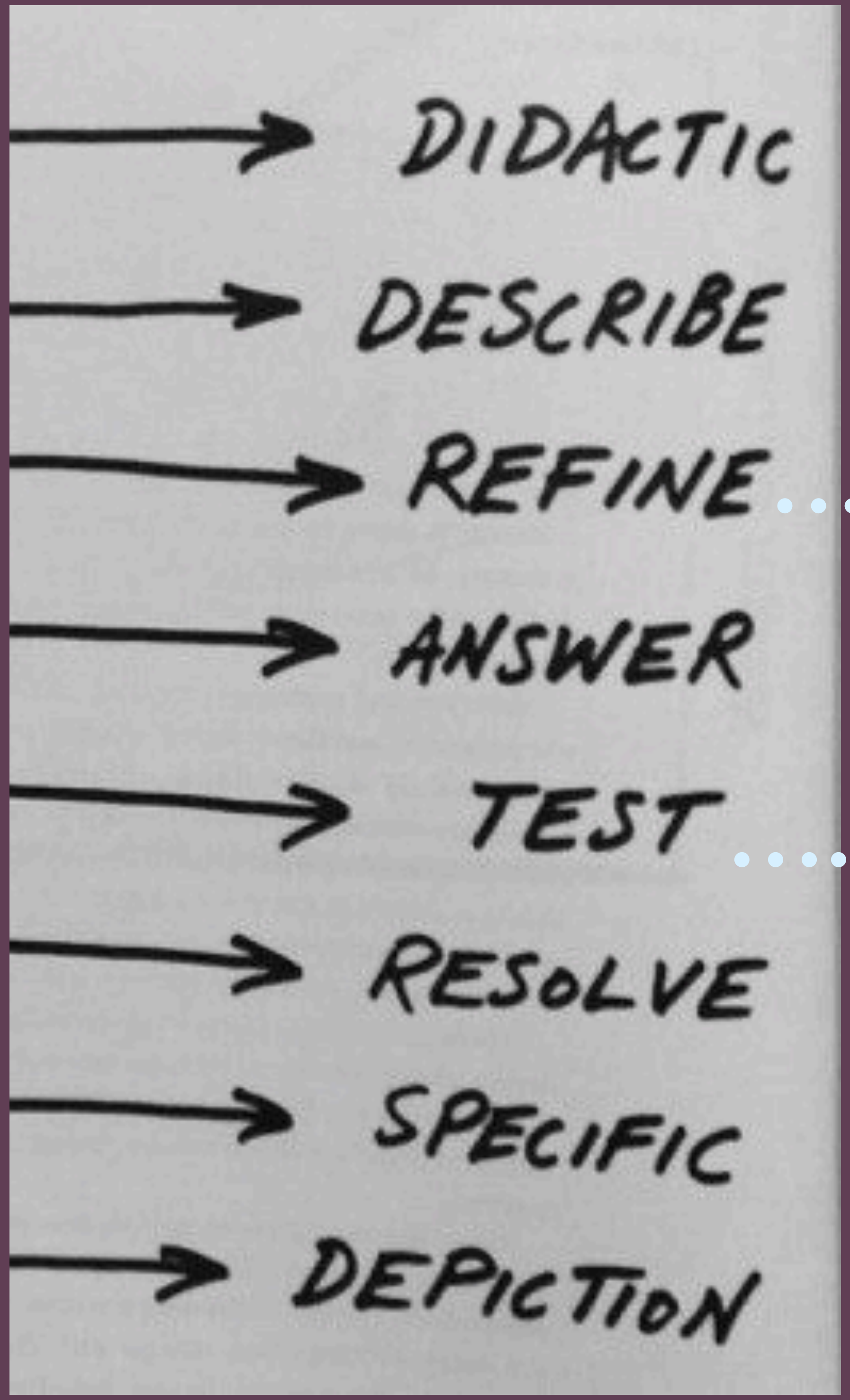
This is **all other points** of a research project

This is the **endpoint** of a research project

Problematic points of view

"OK, we have a good idea. Let's build it / model it / prove it / get training data."

"I spent some time thinking about this and hacking on it, and it's not going to work: it has a fatal flaw."



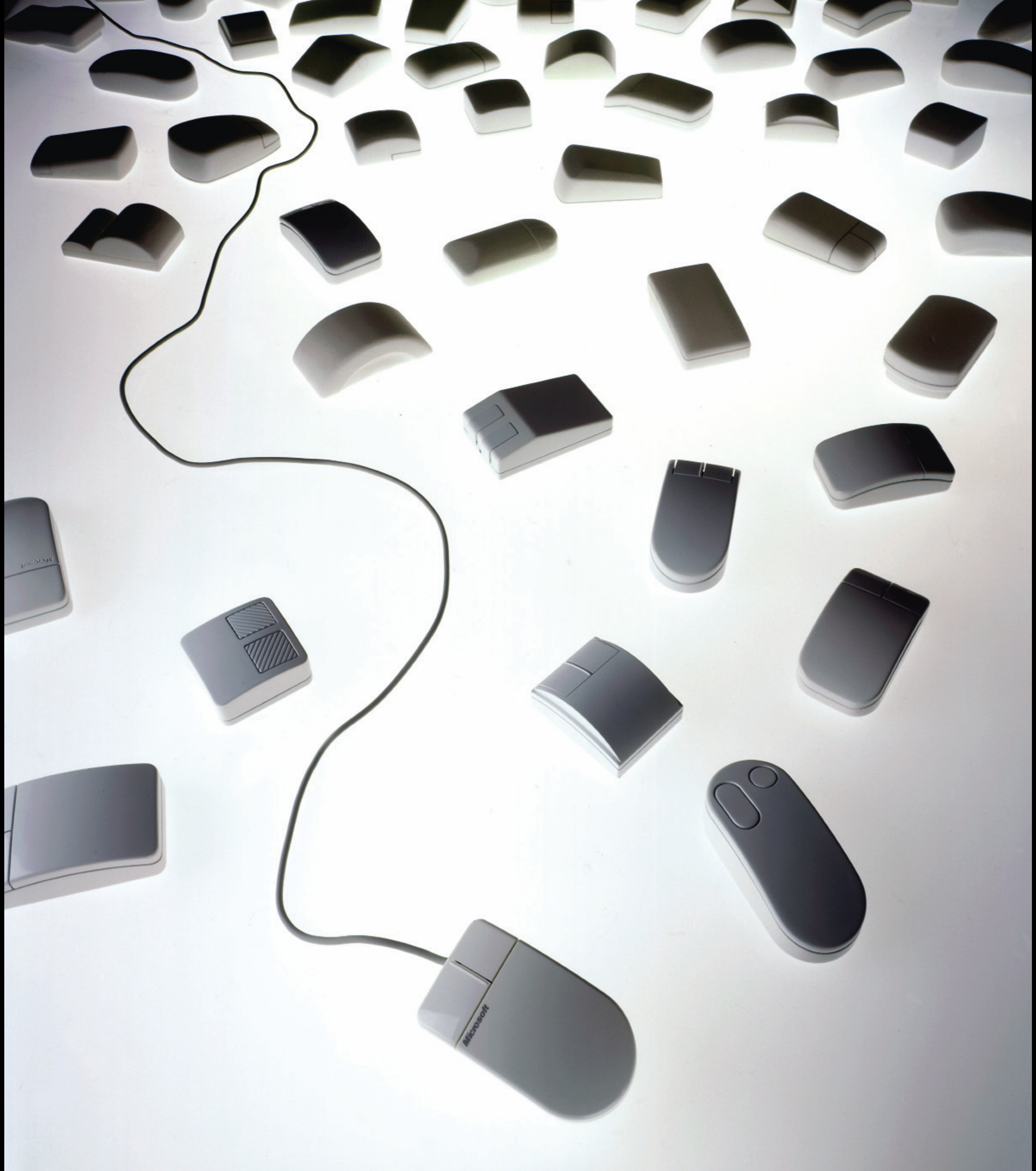
……before knowing what to refine!

……before identifying if that test or flaw is the right one to focus on!

Pick a vector

It may feel like we get stuck unable to solve the problem because we haven't figured out everything else about it. There are too many open questions, and too many possible directions. The more dimensions there are, the harder gradient descent becomes.

Instead of doing trying to do everything at once (project spec), pick one dimension of uncertainty — one vector — and focus on reducing its risk and uncertainty.


Microsoft

Example vectors

Piloting: will this technique work at all? To answer this, we implement a basic version of the technique and mock in the data and other test harness elements.

Engineering: will this technique work with a realistic workload? To answer this, we need to engineer a test harness.

Proving: does the limit exist that I suspect does? To answer this, we start by writing a proof for a simpler case.

Design: what might this interaction look like to an end user? To answer this, we create a low-fi prototype.

Implications

The vectors under consideration will each imply building different parts of your system.

Rather than building them all at once, when you might have to change things later, vectoring instead implies that you start by reducing uncertainty in the most important dimension first — your "inner loop" — and then building out from there.

Vectoring algorithm

1. Generate questions
Untested hunches, risky decisions,
high-level directions

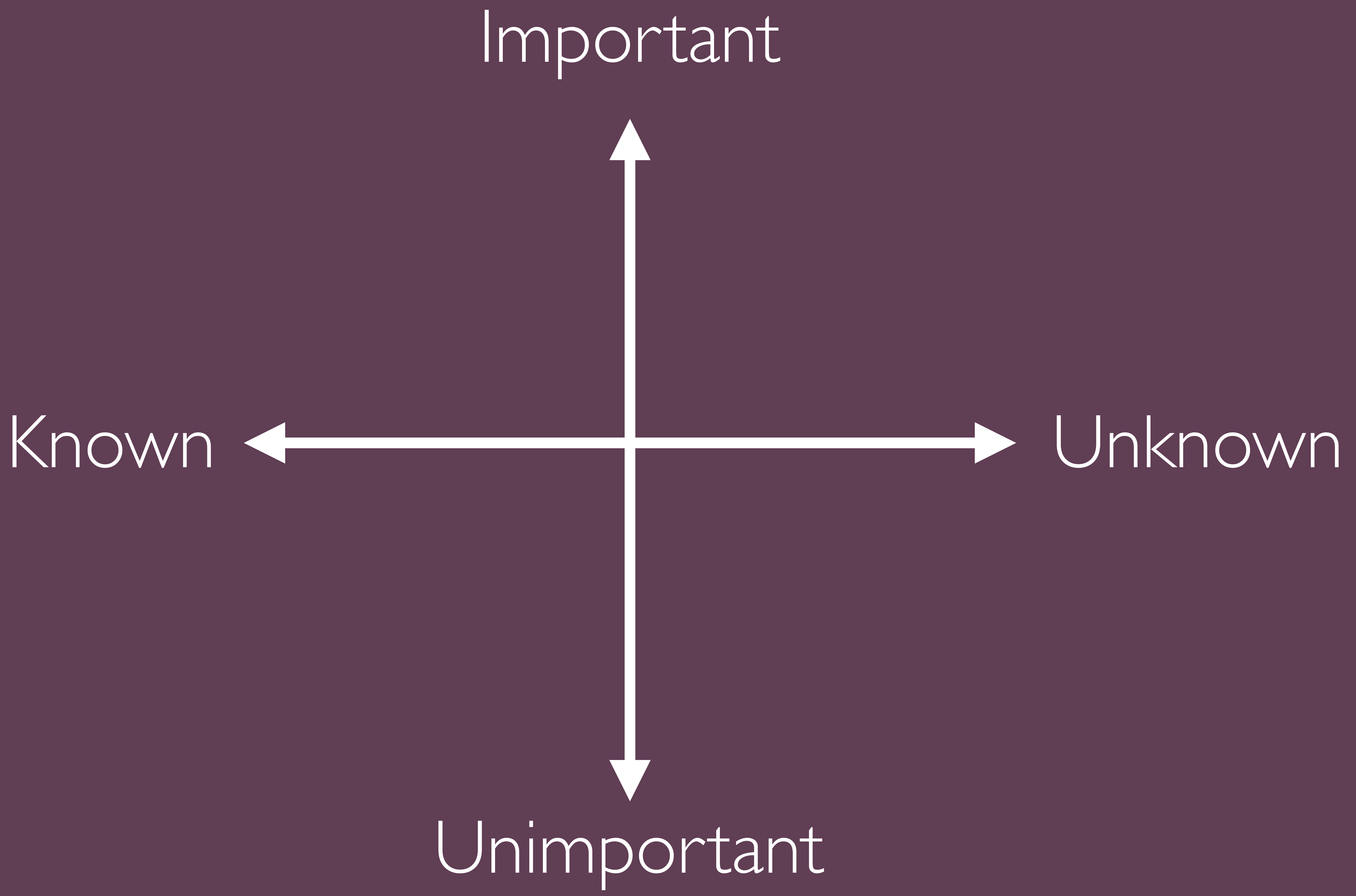
2. Rank your questions
Which is most critical?

3. Pick one and answer it rapidly
Answer only the most critical question
(This is where velocity comes into play)

Assumption mapping

Assumption mapping is a strategy for articulating questions and ranking them.

Try assumption mapping your project [5min]



Let’s Try It

Trolling

While everyone thinks that trolling online is due to a small number of antisocial sociopaths, we had a hunch that "normal" people were responsible for much trolling behavior when triggered.

What's our first step?

We have: dataset of 16M CNN comments (w/ troll flags), Mechanical Turk for studies

Anyone Can Become a Troll: Causes of Trolling Behavior in Online Discussions

Justin Cheng[1], **Michael Bernstein**[1], **Cristian Danescu-Niculescu-Mizil**[2], **Jure Leskovec**[1]
[1]Stanford University, [2]Cornell University
{jcccf, msb, jure}@cs.stanford.edu, cristian@cs.cornell.edu

ABSTRACT

In online communities, antisocial behavior such as trolling disrupts constructive discussion. While prior work suggests that trolling behavior is confined to a vocal and antisocial minority, we demonstrate that ordinary people can engage in such behavior as well. We propose two primary trigger mechanisms: the individual's mood, and the surrounding context of a discussion (e.g., exposure to prior trolling behavior). Through an experiment simulating an online discussion, we find that both negative mood and seeing troll posts by others significantly increases the probability of a user trolling, and together double this probability. To support and extend these results, we study how these same mechanisms play out in the wild via a data-driven, longitudinal analysis of a large online news discussion community. This analysis reveals temporal mood effects, and explores long range patterns of repeated exposure to trolling. A predictive model of trolling behavior shows that mood and discussion context together can explain trolling behavior better than an individual's history of trolling. These results combine to suggest that ordinary people can, under the right circumstances, behave like trolls.

ACM Classification Keywords

H.2.8 Database Management: Database Applications—*Data Mining*; J.4 Computer Applications: Social and Behavioral Sciences

Author Keywords

Trolling; antisocial behavior; online communities

INTRODUCTION

As online discussions become increasingly part of our daily interactions [24], antisocial behavior such as trolling [37, 43], harassment, and bullying [82] is a growing concern. Not only does antisocial behavior result in significant emotional distress [1, 58, 70], but it can also lead to offline harassment and threats of violence [90]. Further, such behavior comprises a substantial fraction of user activity on many web sites [18, 24, 30] – 40% of internet users were victims of online harassment [27]; on CNN.com, over one in five comments are removed by moderators for violating community guidelines. What causes this prevalence of antisocial behavior online?

In this paper, we focus on the causes of *trolling behavior* in discussion communities, defined in the literature as behavior that falls outside acceptable bounds defined by those communities [9, 22, 37]. Prior work argues that trolls are born and not made: those engaging in trolling behavior have unique personality traits [11] and motivations [4, 38, 80]. However, other research suggests that people can be influenced by their environment to act aggressively [20, 41]. As such, is trolling caused by particularly antisocial individuals or by ordinary people? Is trolling behavior innate, or is it situational? Likewise, what are the conditions that affect a person's likelihood of engaging in such behavior? And if people can be influenced to troll, can trolling spread from person to person in a community? By understanding what causes trolling and how it spreads in communities, we can design more robust social systems that can guard against such undesirable behavior.

This paper reports a field experiment and observational analysis of trolling behavior in a popular news discussion community. The former allows us to tease apart the causal mechanisms that affect a user's likelihood of engaging in such behavior. The latter lets us replicate and explore finer grained aspects of these mechanisms as they occur in the wild. Specifically, we focus on two possible causes of trolling behavior: a user's mood, and the surrounding discussion context (e.g., seeing others' troll posts before posting).

Online experiment. We studied the effects of participants' prior mood and the context of a discussion on their likelihood to leave troll-like comments. Negative mood increased the probability of a user subsequently trolling in an online news comment section, as did the presence of prior troll posts written by other users. These factors combined to double participants' baseline rates of engaging in trolling behavior.

Large-scale data analysis. We augment these results with an analysis of over 16 million posts on *CNN.com*, a large online news site where users can discuss published news articles. One out of four posts flagged for abuse are authored by users with no prior record of such posts, suggesting that many undesirable posts can be attributed to ordinary users. Supporting our experimental findings, we show that a user's propensity to troll rises and falls in parallel with known population-level mood shifts throughout the day [32], and exhibits cross-discussion persistence and temporal decay patterns, suggesting that negative mood from bad events linger [41, 45]. Our data analysis also recovers the effect of exposure to prior troll posts in the discussion, and further reveals how the strength of this effect depends on the volume and ordering of these

Trolling

Possible vectors:

- Do people really troll when pissed off?
- Can we train a classifier to predict when someone would troll, and compare weights of personal history vs. other posts and title?
- Does the same person troll more on certain (angry) topics than on other (boring) ones?

Anyone Can Become a Troll: Causes of Trolling Behavior in Online Discussions

Justin Cheng[1], **Michael Bernstein**[1], **Cristian Danescu-Niculescu-Mizil**[2], **Jure Leskovec**[1]
[1]Stanford University, [2]Cornell University
{jcccf, msb, jure}@cs.stanford.edu, cristian@cs.cornell.edu

ABSTRACT

In online communities, antisocial behavior such as trolling disrupts constructive discussion. While prior work suggests that trolling behavior is confined to a vocal and antisocial minority, we demonstrate that ordinary people can engage in such behavior as well. We propose two primary trigger mechanisms: the individual's mood, and the surrounding context of a discussion (e.g., exposure to prior trolling behavior). Through an experiment simulating an online discussion, we find that both negative mood and seeing troll posts by others significantly increases the probability of a user trolling, and together double this probability. To support and extend these results, we study how these same mechanisms play out in the wild via a data-driven, longitudinal analysis of a large online news discussion community. This analysis reveals temporal mood effects, and explores long range patterns of repeated exposure to trolling. A predictive model of trolling behavior shows that mood and discussion context together can explain trolling behavior better than an individual's history of trolling. These results combine to suggest that ordinary people can, under the right circumstances, behave like trolls.

ACM Classification Keywords

H.2.8 Database Management: Database Applications—*Data Mining*; J.4 Computer Applications: Social and Behavioral Sciences

Author Keywords

Trolling; antisocial behavior; online communities

INTRODUCTION

As online discussions become increasingly part of our daily interactions [24], antisocial behavior such as trolling [37, 43], harassment, and bullying [82] is a growing concern. Not only does antisocial behavior result in significant emotional distress [1, 58, 70], but it can also lead to offline harassment and threats of violence [90]. Further, such behavior comprises a substantial fraction of user activity on many web sites [18, 24, 30] – 40% of internet users were victims of online harassment [27]; on CNN.com, over one in five comments are removed by moderators for violating community guidelines. What causes this prevalence of antisocial behavior online?

In this paper, we focus on the causes of *trolling behavior* in discussion communities, defined in the literature as behavior that falls outside acceptable bounds defined by those communities [9, 22, 37]. Prior work argues that trolls are born and not made: those engaging in trolling behavior have unique personality traits [11] and motivations [4, 38, 80]. However, other research suggests that people can be influenced by their environment to act aggressively [20, 41]. As such, is trolling caused by particularly antisocial individuals or by ordinary people? Is trolling behavior innate, or is it situational? Likewise, what are the conditions that affect a person's likelihood of engaging in such behavior? And if people can be influenced to troll, can trolling spread from person to person in a community? By understanding what causes trolling and how it spreads in communities, we can design more robust social systems that can guard against such undesirable behavior.

This paper reports a field experiment and observational analysis of trolling behavior in a popular news discussion community. The former allows us to tease apart the causal mechanisms that affect a user's likelihood of engaging in such behavior. The latter lets us replicate and explore finer grained aspects of these mechanisms as they occur in the wild. Specifically, we focus on two possible causes of trolling behavior: a user's mood, and the surrounding discussion context (e.g., seeing others' troll posts before posting).

Online experiment. We studied the effects of participants' prior mood and the context of a discussion on their likelihood to leave troll-like comments. Negative mood increased the probability of a user subsequently trolling in an online news comment section, as did the presence of prior troll posts written by other users. These factors combined to double participants' baseline rates of engaging in trolling behavior.

Large-scale data analysis. We augment these results with an analysis of over 16 million posts on *CNN.com*, a large online news site where users can discuss published news articles. One out of four posts flagged for abuse are authored by users with no prior record of such posts, suggesting that many undesirable posts can be attributed to ordinary users. Supporting our experimental findings, we show that a user's propensity to troll rises and falls in parallel with known population-level mood shifts throughout the day [32], and exhibits cross-discussion persistence and temporal decay patterns, suggesting that negative mood from bad events linger [41, 45]. Our data analysis also recovers the effect of exposure to prior troll posts in the discussion, and further reveals how the strength of this effect depends on the volume and ordering of these

Teaming

We wanted to create an algorithm that would weave collaboration networks to help spread ideas over time by moving people from team to team.

What's our first step?

Hive: Collective Design Through Network Rotation

NILOUFAR SALEHI, UC Berkeley, USA
MICHAEL S. BERNSTEIN, Stanford University, USA

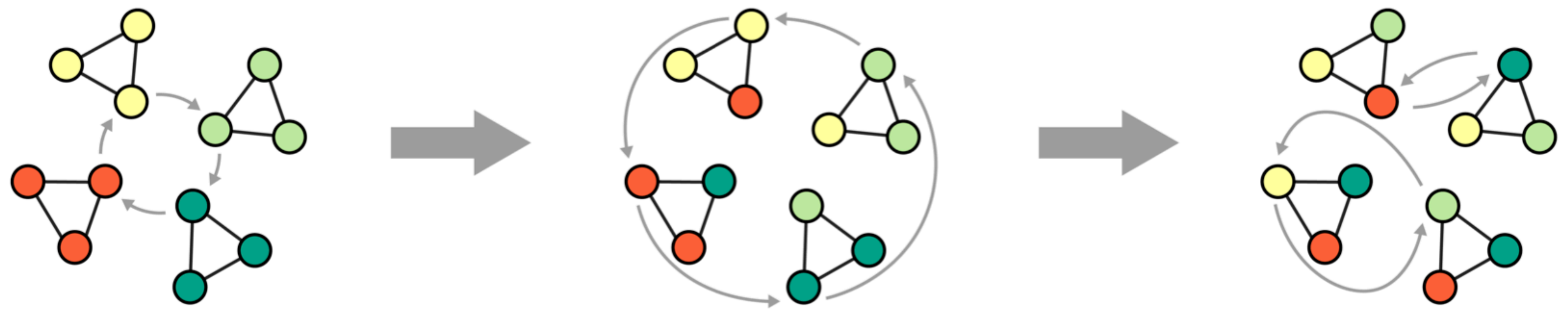


Fig. 1. Hive facilitates engagement with diverse viewpoints by rotating team membership in a collective over time. We introduce algorithmically-mediated *network rotation* to manage who should move, and when, to bring positive external influence to a team.

Collectives gather online around challenges they face, but frequently fail to envision shared outcomes to act on together. Prior work has developed systems for improving collective ideation and design by exposing people to each others' ideas and encouraging them to intermix those ideas. However, organizational behavior research has demonstrated that intermixing ideas does not result in meaningful engagement with those ideas. In this paper, we introduce a new class of collective design system that intermixes *people* instead of *ideas*: instead of receiving mere exposure to others' ideas, participants engage deeply with other members of the collective who represent those ideas, increasing engagement and influence. We thus present Hive: a system that organizes a collective into small teams, then intermixes people by rotating team membership over time. At a technical level, Hive must balance two competing forces: (1) networks are better at connecting diverse perspectives when network efficiency is high, but (2) moving people diminishes tie strength within teams. Hive balances these two needs through *network rotation*: an optimization algorithm that computes who should move where, and when. A controlled study compared network rotation to alternative rotation systems which maximize only tie strength or network efficiency, finding that network rotation produced higher-rated proposals. Hive has been deployed by Mozilla for a real-world open design drive to improve Firefox accessibility.

CCS Concepts: • **Human-centered computing** → **Collaborative and social computing systems and tools**;

Additional Key Words and Phrases: Design; online collaboration; participatory design; teams.

ACM Reference Format:
Niloufar Salehi and Michael S. Bernstein. 2018. Hive: Collective Design Through Network Rotation. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 151 (November 2018), 26 pages. https://doi.org/10.1145/3274420

Authors' addresses: Niloufar Salehi, UC Berkeley, School of Information, South Hall, Berkeley, CA, 94720, USA, nsalehi@berkeley.edu; Michael S. Bernstein, Stanford University, 353 Serra Mall, Stanford, CA, 94305, USA, msb@cs.stanford.edu.

Teaming

Possible vectors:

Do new members with new perspectives actually exert influence in practice?

If we prioritize or de-prioritize membership rotation in a simple (greedy) algorithm, does it lead to different outcomes in the collaboration network?

Hive: Collective Design Through Network Rotation

NILOUFAR SALEHI, UC Berkeley, USA
MICHAEL S. BERNSTEIN, Stanford University, USA

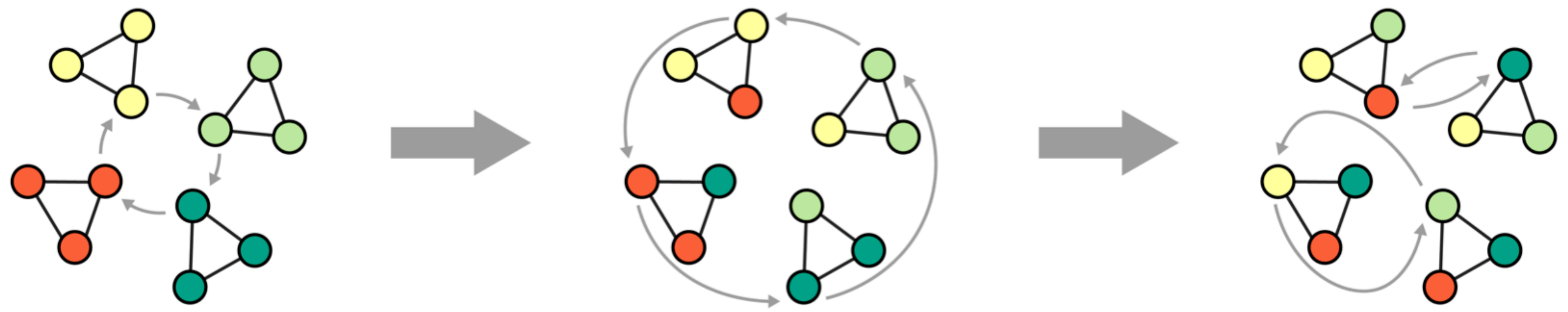


Fig. 1. Hive facilitates engagement with diverse viewpoints by rotating team membership in a collective over time. We introduce algorithmically-mediated *network rotation* to manage who should move, and when, to bring positive external influence to a team.

Collectives gather online around challenges they face, but frequently fail to envision shared outcomes to act on together. Prior work has developed systems for improving collective ideation and design by exposing people to each others' ideas and encouraging them to intermix those ideas. However, organizational behavior research has demonstrated that intermixing ideas does not result in meaningful engagement with those ideas. In this paper, we introduce a new class of collective design system that intermixes *people* instead of *ideas*: instead of receiving mere exposure to others' ideas, participants engage deeply with other members of the collective who represent those ideas, increasing engagement and influence. We thus present Hive: a system that organizes a collective into small teams, then intermixes people by rotating team membership over time. At a technical level, Hive must balance two competing forces: (1) networks are better at connecting diverse perspectives when network efficiency is high, but (2) moving people diminishes tie strength within teams. Hive balances these two needs through *network rotation*: an optimization algorithm that computes who should move where, and when. A controlled study compared network rotation to alternative rotation systems which maximize only tie strength or network efficiency, finding that network rotation produced higher-rated proposals. Hive has been deployed by Mozilla for a real-world open design drive to improve Firefox accessibility.

CCS Concepts: • **Human-centered computing** → **Collaborative and social computing systems and tools**;

Additional Key Words and Phrases: Design; online collaboration; participatory design; teams.

ACM Reference Format:
Niloufar Salehi and Michael S. Bernstein. 2018. Hive: Collective Design Through Network Rotation. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 151 (November 2018), 26 pages. https://doi.org/10.1145/3274420

Authors' addresses: Niloufar Salehi, UC Berkeley, School of Information, South Hall, Berkeley, CA, 94720, USA, nsalehi@berkeley.edu; Michael S. Bernstein, Stanford University, 353 Serra Mall, Stanford, CA, 94305, USA, msb@cs.stanford.edu.

Learning

We thought that, in domains where ML still cannot succeed, we could draw on crowdsourcing to identify human-labeled predictive features. In other words, that people are great at identifying potentially informative features, but might be poor at weighing those features correctly to arrive at a prediction.

What's our first step?

Flock: Hybrid Crowd-Machine Learning Classifiers

Justin Cheng and Michael S. Bernstein
Stanford University
{jcccf, msb}@cs.stanford.edu

ABSTRACT
We present hybrid crowd-machine learning classifiers: classification models that start with a written description of a learning goal, use the crowd to suggest predictive features and label data, and then weigh these features using machine learning to produce models that are accurate and use human-understandable features. These hybrid classifiers enable fast prototyping of machine learning models that can improve on both algorithm performance and human judgment, and accomplish tasks where automated feature extraction is not yet feasible. *Flock*, an interactive machine learning platform, instantiates this approach. To generate informative features, Flock asks the crowd to compare paired examples, an approach inspired by analogical encoding. The crowd's efforts can be focused on specific subsets of the input space where machine-extracted features are not predictive, or instead used to partition the input space and improve algorithm performance in subregions of the space. An evaluation on six prediction tasks, ranging from detecting deception to differentiating impressionist artists, demonstrated that aggregating crowd features improves upon both asking the crowd for a direct prediction and off-the-shelf machine learning features by over 10%. Further, hybrid systems that use both crowd-nominated and machine-extracted features can outperform those that use either in isolation.

Author Keywords
Crowdsourcing, interactive machine learning

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION
Identifying predictive features is key to creating effective machine learning classifiers. Whether the task is link prediction or sentiment analysis, and no matter the underlying model, the "black art" of feature engineering plays a critical role in success [10]. Feature engineering is largely domain-specific, and users of machine learning systems spend untold hours experimenting. Often, the most predictive features only emerge after many iterations [36]. And though feature engineers may have deep domain expertise, they are only able to incorporate features that are extractable via code.

CSCW 2015, March 14–18, 2015, Vancouver, BC, Canada.

ACM 978-1-4503-2922-4/15/03 ...$15.00.
http://dx.doi.org/10.1145/2675133.2675214

Figure 1. Flock is a hybrid crowd-machine learning platform that capitalizes on analogical encoding to guide crowds to nominate effective features, then uses machine learning techniques to aggregate their labels.

However, *embedding crowds inside of machine learning architectures* opens the door to hybrid learners that can explore feature spaces that are largely unreachable by automatic extraction, then train models that use human-understandable features (Figure 1). Doing so enables fast prototyping of classifiers that can exceed both machine and expert performance. In this paper, we demonstrate classifiers that identify people who are lying, perform quality assessment of Wikipedia articles, and differentiate impressionist artists who use similar styles. Previous work that bridges crowdsourcing and machine learning has focused on optimizing the crowd's efforts (e.g., [8, 21, 39]): we suggest that inverting the relationship and embedding crowd insight inside live classifiers enables machine learning to be deployed for new kinds of tasks.

We present *Flock*, an end-user machine learning platform that uses paid crowdsourcing to speed up the prototyping loop and augment the performance of machine learning systems. Flock contributes a model for creating hybrid classifiers that intelligently embed both crowd and machine features. The system allows users to rapidly author hybrid crowd-machine learners by structuring a feature nomination process using the crowd, aggregating the suggested features, then collecting labels on these new features. It loops and gathers more crowd features to improve performance on subsets of the space where the model is misclassifying many examples. For instance, given a decision tree that uses machine-readable features, Flock can dynamically grow subtrees from nodes that have high classification error, or even replace whole branches. In addition to

Learning

Possible vectors:

- Can people identify predictive features for a single domain, e.g., lie detection?
- Can people estimate which features are going to be informative?
- Would a hybrid classifier (human features and labels as input to an ML model) actually perform well?

Flock: Hybrid Crowd-Machine Learning Classifiers

Justin Cheng and Michael S. Bernstein
Stanford University
{jcccf, msb}@cs.stanford.edu

ABSTRACT

We present hybrid crowd-machine learning classifiers: classification models that start with a written description of a learning goal, use the crowd to suggest predictive features and label data, and then weigh these features using machine learning to produce models that are accurate and use human-understandable features. These hybrid classifiers enable fast prototyping of machine learning models that can improve on both algorithm performance and human judgment, and accomplish tasks where automated feature extraction is not yet feasible. *Flock*, an interactive machine learning platform, instantiates this approach. To generate informative features, Flock asks the crowd to compare paired examples, an approach inspired by analogical encoding. The crowd's efforts can be focused on specific subsets of the input space where machine-extracted features are not predictive, or instead used to partition the input space and improve algorithm performance in subregions of the space. An evaluation on six prediction tasks, ranging from detecting deception to differentiating impressionist artists, demonstrated that aggregating crowd features improves upon both asking the crowd for a direct prediction and off-the-shelf machine learning features by over 10%. Further, hybrid systems that use both crowd-nominated and machine-extracted features can outperform those that use either in isolation.

Author Keywords

Crowdsourcing, interactive machine learning

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Identifying predictive features is key to creating effective machine learning classifiers. Whether the task is link prediction or sentiment analysis, and no matter the underlying model, the "black art" of feature engineering plays a critical role in success [10]. Feature engineering is largely domain-specific, and users of machine learning systems spend untold hours experimenting. Often, the most predictive features only emerge after many iterations [36]. And though feature engineers may have deep domain expertise, they are only able to incorporate features that are extractable via code.

CSCW 2015, March 14–18, 2015, Vancouver, BC, Canada.

ACM 978-1-4503-2922-4/15/03 ...$15.00.
http://dx.doi.org/10.1145/2675133.2675214

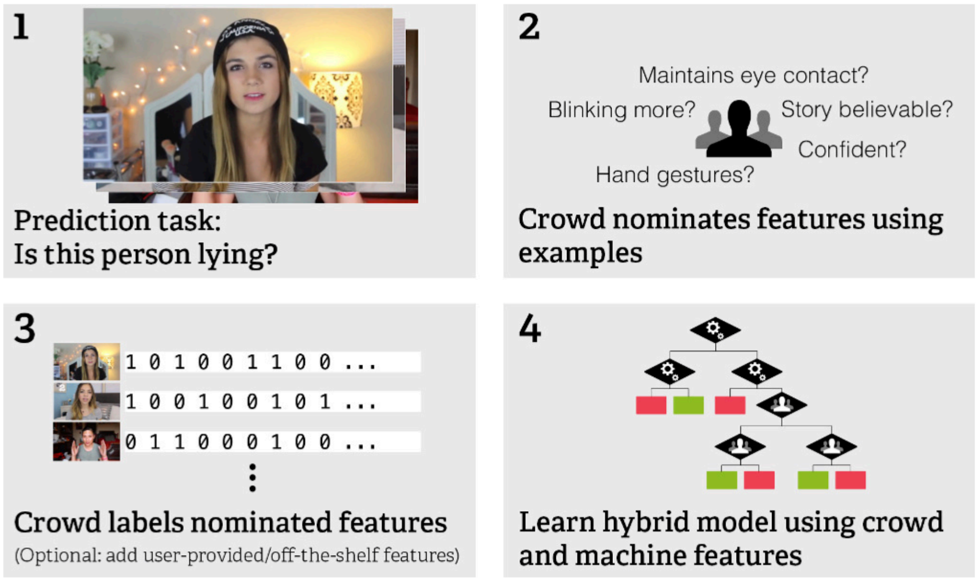


Figure 1. Flock is a hybrid crowd-machine learning platform that capitalizes on analogical encoding to guide crowds to nominate effective features, then uses machine learning techniques to aggregate their labels.

However, *embedding crowds inside of machine learning architectures* opens the door to hybrid learners that can explore feature spaces that are largely unreachable by automatic extraction, then train models that use human-understandable features (Figure 1). Doing so enables fast prototyping of classifiers that can exceed both machine and expert performance. In this paper, we demonstrate classifiers that identify people who are lying, perform quality assessment of Wikipedia articles, and differentiate impressionist artists who use similar styles. Previous work that bridges crowdsourcing and machine learning has focused on optimizing the crowd's efforts (e.g., [8, 21, 39]): we suggest that inverting the relationship and embedding crowd insight inside live classifiers enables machine learning to be deployed for new kinds of tasks.

We present *Flock*, an end-user machine learning platform that uses paid crowdsourcing to speed up the prototyping loop and augment the performance of machine learning systems. Flock contributes a model for creating hybrid classifiers that intelligently embed both crowd and machine features. The system allows users to rapidly author hybrid crowd-machine learners by structuring a feature nomination process using the crowd, aggregating the suggested features, then collecting labels on these new features. It loops and gathers more crowd features to improve performance on subsets of the space where the model is misclassifying many examples. For instance, given a decision tree that uses machine-readable features, Flock can dynamically grow subtrees from nodes that have high classification error, or even replace whole branches. In addition to

Why is vectoring so important?

"If Ernest Hemingway, James Mitchener, Neil Simon, Frank Lloyd Wright, and Pablo Picasso could not get it right the first time, what makes you think that you will?"

— Paul Heckel

Iteration >> planning

Ideas rarely land exactly where you expect they will. It's best to test the most critical assumptions quickly, so that you can understand whether your hunch will play out, and what problems are worth spending time solving vs. kludging.

Human creative work is best in a loop of reflection and iteration. Vectoring is a way to make sure you're getting the most iteration cycles.

Re-vectoring

Often, after vectoring and reducing uncertainty in one dimension, it raises new questions and uncertainties.

In the next round of vectoring, you re-prioritize:

> If you get unexpected results and are confused (most of the time!), maybe it means you take a new angle to reduce uncertainty on a vector related to the prior one.
>
> If you answer your question to your own satisfaction (not completely, just to your satisfaction), you move on to the next most important vector

Magnitude of your vector

The result of vectoring should be something achievable in about a week's sprint. If it's not, you've picked too broad a question to answer.

> If your vectoring for "Can normal people be responsible for a lot of the trolling online?" is "Can normal people be responsible for a lot of the trolling on CNN.com?", you're still way too broad.
>
> That's evidence that you've just rescaled your project, ↔ not picked a vector.

Takeaways, in brief

1) The temptation is to try and solve the problem that’s set in front of you. Don’t.

2) Vectoring is a process of identifying the dimension of highest impact+uncertainty, and prioritizing that dimension while scaffolding the others

3) Successful vectoring enables you to rapidly hone in on the core insight of your research project

Assignment 4

At this point, your project transitions to a state where your team is working to try and achieve the goal you set out in Assignment 3.

Each week for the next several weeks, your team will perform vectoring, submit a brief summary and slide, and report in section:

This week's vector

This week's plan

This week's result

Next week's vector

Next week's plan

Vectoring in Research