CSC 595 - Research Skills

Generating Ideas #1: Research Patterns

Nishant Mehta

*For the structure and some content, credit goes to
Nick Feamster and Alex Gray: https://noise-lab.net/research-course/

Key steps of research

(1) Finding a problem

(2) Gain an understanding of the problem

(3) Devise a plan (and carry it out)

(4) Post-solution

How can I find a problem?

- Be aware of the present
  - Finding trends
- Anticipate the future
  - Predicting trends
- Revisit the past
  - What did others miss?
  - Recall: tunnel vision (limited to paradigm), limited attention (one trend started, what others could have started?)

Why is finding a trend a good idea?

- A community of people gravitating around something
  - People should have identified important problems to work on
  - Continual progress; problems are being solved, new ones arise; some problems evolve. Good starting point for some strategies discussed later
- If enough people joined a party, there should be something interesting there
- If you make contributions and your work is good, your work will be cited (be optimistic! pessimism is your greatest enemy)

How can I spot trends?

- Do a scan of recent proceedings of a top conference
- Look at oral sessions if organized by topic
- Pay attention to papers selected for orals
  OR Look at “best papers”
  - Papers that got more attention likely to see wider adoption
- See what the famous people are doing
  - Careful: Not always a good work, but a good heuristic. People pay attention to big shots
- Look at proceedings from *previous* year (at least 1 year has passed) - see which papers are most highly cited (benefit of hindsight)

How can I anticipate trends?

- Look at funding agencies' calls for proposals
- Money talks - if work in a certain direction is funded, that direction could be a future trend
- Examples: quantum computing, smart health
- Careful: funding proposals might have some lag (crafted by people in grant agencies rather than people directly connected to research "on the ground")
  - So, look for more experimental (more forward looking) proposals
  - Example: "Future of Life" / "Future of Humanity" type proposals for AI
- Beyond NSERC, also look at NSF calls. Why? NSF has much more funding, so more calls
- Also CIHR ↔ NIH

How can I anticipate trends?

- Look at workshops
- A good workshop covers less mature areas of a conference
    - Caution: beware of workshops that focus on mature topics;
      controversial opinion: these are just for papers rejected from a conference)
- Why workshops?
    - Workshop was approved, so there’s interest from at least some more senior researchers
    - A community of people working in emerging topics
    - At workshops, easy to talk with people: ~50 people… vs ~2000 people at conference)
    - More relaxed atmosphere than conferences. Ideas flow more freely

How can I benefit from revisiting the past?

- Look at OLD papers.
- Great researchers often have old works that:
  - are not as well-cited, but have great ideas/problems
  - might be well-cited, but people only took a narrow sliver (hopped on one train; what about the other potential trains that could have emanated from the paper?)
- Another benefit:
  - avoid group-think; immerse yourself in thinking from a previous time to “think different”

How can I benefit from revisiting the past?

- Examples from machine learning:
  - Maillard Sampling - extracted from 9-year old PhD thesis (OK, not *that* old)
  - Generalized Reversed Information Projection (GRIP) - generalization of RIP from 21-year old PhD thesis
  - Thompson sampling from 1933 - did not re-emerge until 2010! And now it's BIG!
  - natural gradient method (information geometry + optimization)

What are strategies for me to find a PRECISE problem?

- Specialize
- Unify
- Interpolate
- Complain (Tell people “You’re doing it wrong”)
- Secret weapon
- Automate
- Decompose
- Revisit old problems

Specialize

- Progress may have stalled on a problem
- Are there important special cases (some added constraints)?
- Can you formulate those special cases + argue why they're important?
- And can you make progress on those (perhaps because you have a secret weapon)?
- Caution:
    - Novelty alone is not enough. Need to argue why new problem is IMPORTANT

Specialize

- Examples:
  - Take any problem framed for central computation; look at distributed version of the problem
  - Consider interface for general population. Does it work well for subgroup (e.g., the deaf)?
  - Take any subclass of learning problems; look at version of problem where data can be corrupted (consider different corruption models)
  - Routing → Fair routing; Learning → Fair learning
  - ANY problem whose input is personal data → Private version of the problem
    - Private convex optimization, Private dimension reduction, Private learning
  - Seek optimal worst-case performance, but also do better in easier cases
    - Each easier case is a new problem!

Unify

- People have been working problems A, B, C, …, Z
  - These are all special cases of a unified (more general) problem
  - Example: Minimize squared loss, Minimize cross-entropy loss, MInimize exponential loss. Unification: minimize *any* proper loss
- People have solutions 1, 2, …, 100 to related problems
  - Can these solutions fit into an entire framework? Unifying *solutions* is a good *problem*
- Unification is often appreciated. Forming connections can provide more understanding
- Sometimes unifying a problem provides clearer understanding and makes the problem *easier* to think about
- Also good for the field. No need for the 101st solution. Let people shift to new problems!

Interpolate

- What is in between?
  - Squared loss …………… Huber Loss ……………… Absolute loss
  - probabilistic graphical model ……… Markov Logic Network ……… First-order logic
  - Full-information feedback ……… Graphical Feedback ……… Bandit feedback
  - Truthfulness ……… Approximate Truthfulness ……… Non-truthfulness
  - Adversarial data ……… Easy Data (different types of “luckiness”) ……… Nice, i.i.d. data
- Can lead to very exciting questions.
- Arguably, entire research areas can emerge from this (best of both worlds, beyond minimax if you can)

Complain: Tell people “you’re doing it wrong!”

- Is it the type of problem with a quantitative objective? Do people have the right objective? Is it truly a multi-objective problem?
- Is it an area with a lot of (old) theory? Does the theory not describe what actually happens?
- Is the problem actually impossible? Or two/three popular objectives cannot all be optimized simultaneously?
  - Impossibility results
  - Trade-offs
- Can you identify a major issue with a certain often-used kind of user study or ethnographic study?

Complain: Tell people “you’re doing it wrong!”

- Caution:
  - Have to present work very carefully. It’s similar to paradigm shift. People get defensive about what they’ve done. Need to frame this as an invitation.
  - Don’t just complain. Offer solution or path toward solution. Just complaining rarely leads to a paper!
    - Psychologically, people like to accept positive results (whether or not that’s how it should be)
- Examples:
  - Understanding generalization in deep learning requires rethinking generalization
  - Impossibility results in clustering
- **Class exercise: what examples can you think of in your field? Think of BIG examples**

Time Log - Reflect

Did you have a plan for certain work periods? Did you do what you intended to do?

What did you learn about your habits?

When were you best able to work without interruptions?

What were examples of interruptions?

Secret weapon

- Do I have special knowledge that most people don’t have?
- Or can I get this knowledge (training from expert, ambition to learn on my own)?
- Then I have a secret weapon: a way to tackle problems that most people do not
- Using this “hammer”, look for nails (problems that match up with your techniques)
- Having a unique “hammer” can lead to people approaching me (“I’m the expert”)

Secret weapon: examples

- Background in neuroscience, now doing CS
- Background in linguistics —> opportunity to supercharge research in natural language processing using hardcore linguistics aspects (prosody, etc.)
- Strong statistics background while working in an HCI
- Can read French… or Japanese (access to MANY old papers)

Secret weapon: examples

- Expertise (or exposure to an expert) in e-values; learn them, get better statistical results than everyone else
- Understanding an obscure algorithmic framework
- Lower bounds / hardness results are intuitive for you (aren’t for most people)
- Hard subfield of math: algebraic geometry, complex analysis, topology, module theory (?)

Class Exercise

Do you have a secret weapon? What is it?
Why do most people in your area not have it?

OR Is there a specific secret weapon you hope to get?

Automate

- Consider what can be automated:
  - Problem: need to find humans to label data (or do other manual tasks)
  - Solution: Crowdsourcing - learning from wisdom of the crowds. Mechanical Turk

  - Problem: Many sets of roommates struggling to find fair way to split rent
  - Solution: Frame as a fair division problem. Spliddit for figuring out how to split rent

  - Problem: for each statistical model, someone manually derives specialized algorithm for learning that model
  - Solution: AutoBayes - automatically derive algorithms for learning the model

Automate

- Consider what can be automated:
  - Problem: need to manually create prototypes (lots of work!)
  - Solution: 3D Printing - rapid prototyping

  - Problem: Manual selection of “kernel function” in kernel methods for machine learning
  - Solution: “Learning the Kernel” - data-driven way to learn the kernel function

  - Problem: Online learning/optimization methods have annoying tuning parameters that practitioners or theorists need to set
  - Solution: Parameter-free online learning: no more hyperparameters (set in data-driven way)

Decompose (Note: this gets into a path toward a solution)

- Split problem into subproblems/subgoals (work backwards)
- Is there a subproblem that is interesting in their own right?
  If yes, solving a subproblem could be a separate paper!
  - Think about how a solution to a subproblem might feed into solutions
    to other problems in the future (value beyond the original problem)

Linchpins

- Linchpin: keeps the wheel attached the axle (so wheel doesn't roll off the car)
  - Metaphorically, a linchpin is something on which *many* things depend
- Do many problems depend on a common subproblem?
  Work on that subproblem if you can!
  Even better if you're the first one to realize that the subproblem is a linchpin
- Example from theoretical computer science: Unique Games Conjecture
  - If conjecture is true and $P \neq NP$, many problems of interest cannot be exactly solved in polynomial time AND we can't even get an approximate solution in polynomial time
- Example from optics: Blue Laser
  - Allowed for Blu-Ray (high density optical storage), much better underwater communication, many medical applications, etc.

Revisit old problems

- Worthwhile to revisit old problems. Why? Recent progress might make them more approachable

- Especially VERY recent progress, combined with quite old problems

    - Fewer people may make the connection; can you be the first one?
      This requires some expertise (related to secret weapon)

Key steps of research

(1) Finding a problem

(2) Gain an understanding of the problem

(3) Devise a plan (and carry it out)

(4) Post-solution

Gain an understanding of the problem

- Good ideas are typically based on past experience and previously acquired knowledge
- So, gather the “materials”
  - Have I seen related problems?
  - If I have seen *many* related problems, which ones have similar unknowns?

Gain an understanding of the problem

- Collect data
    - Might mean running preliminary experiments or simulations
    - OR looking at related literature to bootstrap some expertise (need to start *somewhere*!)
- Develop intuition (get a feel for the problem)
    - Come up with toy model
    - Simplify problem, play with simplified version

Key steps of research

(1) Finding a problem

(2) Gain an understanding of the problem

(3) Devise a plan (and carry it out)

(4) Post-solution

Devise a Plan: Refine/Formalize the problem

- Performance measure (objective) matters
  - User satisfaction (how to measure?)
  - Error (how to measure?)
  - Quality of animation rendering (how to measure?)
- If no concrete objective yet, try to come up with one
- If already have objective, use newly gained understanding of problem to reconsider objective

Devise a Plan: Refine/Formalize the problem

- Can I come up with warm-up problems?
  - Useful for gaining understanding
  - Part of a plan. Slowly relax “warm-up assumptions” by reintroducing one difficulty at a time

Decomposition

- Can I split the problem into separate pieces?
- And can I understand and attack those pieces separately?
- And can do I have an understanding of how the pieces plug back to together to solve original problem?
- Opportunity for collaboration:
  - Different people work on different pieces, and share understanding periodically
  - Finally, combine pieces together, check very carefully that solution works

Use analogical reasoning

- Related problems might not immediately pop out
- Consider analogies: think abstractly to see if one problem can be mapped to another
  - How might solution to the other problem map to a solution to my problem?

Start simply

- Consider a sequence of warm-up problems which eventually approach the full problem
- Why warm-up problems?
  - What's more fun?

Learning chess with an expert that crushes you

Learning chess with someone slightly above your level

Start simply

- Can you solve each warm-up problem?
  - Get more understanding (and strength for solving full problem) along the way
  - Similar to “curriculum learning”
- How to get warm-up problems?
  - Include additional assumptions
  - Simplify structure, discard some components

Devising a Plan

- Analysis: work backwards (Pappus of Alexandria)
    - Suppose the goal can be achieved (there is a way to cross a river)
    - From what antecedent A can we get our goal?
      (is there a fallen tree connecting the river banks?)
    - From what antecedent B can we get A?
      (is there a tree and do I know how to make it fall?)
    - From what antecedent C can we get B?
      (can I form a cutting instrument (an axe)?)
- Can lead to new tools (axe, bridge)

Carrying out the plan

- Synthesis
  - Follow the steps from the “working backwards” plan in forward order
  - Constructive
  - Putting everything together

Look back at your solution

- For math, data/statistical analysis, etc., important to check each step VERY carefully
  - human nature is to stop when we achieve desired result
  - mistakes that make results seem significant aren't corrected!
    (not good for science, will hurt your reputation in the long run)
  - So, don't stop as soon as you got answer you like. Treat the result as if it came from a competing researcher, and see if you can poke holes in the argument.

Look back at your solution

- Consolidate the knowledge developed along the way to solving the problem
- Can you get the result in a different way? (two ways of understanding are better than one)
  Can you get it in a simpler way?
- Can the *path* you took help with other problems?
- Can the solution be applied/adapted for other problems?
- Was a tool developed (a new hammer) that can be applied to other problems?

Generalize

- Can you generalize what you have done to widen its reach?
- If so, can you form a theory?