I Fought The Odds (And The Odds Won)

Mixed Martial Arts Prediction:
Style Inference with Graph Algorithms

Andrew Hunter

Mixed Martial Arts - Overview

Gabriel Gonzaga knocking out Mirko Filipović at UFC 70

- Rapidly growing new sport
- 1-on-1 fighting allowing strikes, wrestling, chokeholds, and joint-locks
- Eclipsed boxing in 2006 PPV revenue
- Several competing promotions: UFC, EliteXC, WEC, DREAM, many others
- Like many sports, extensive gambling

Mixed Martial Arts – Unpredictable

Matt Serra KOing Georges St-Pierre, a 11-1 favorite coming in

- Fights aren't random
- Skills and athletic ability matter
- Fights aren't sure things
- Upsets happen all the time

Conclusion: We can make a lot of money if we can accurately predict fight winners.

My Goals

Lyoto Machida about to submit Rameau Thierry Sokoudjou

Vegas sets odds by many factors:

- Physical characteristics
- Past record
- Mutual fights
- **Styles**
- Expert knowledge

These are hard to automate:

- How do I tell a computer "Lyoto looks a bit stronger, but his chin's a little suspect?"

I want to try and make predictions based off one piece of data: who has fought whom, and what happened. I won't beat Vegas, but I should be able to make decent predictions.

Power Levels

There's no way that can be right, *can* it?

- Ideally Bob would have rating 1000, and Alice would have rating 500.
- Then, Bob probably wins
- Used in chess, go, soccer (!)
- ELO rating/algorithm well known, simple to use, produces great results

Styles Make Fights

Commonly accepted wisdom:

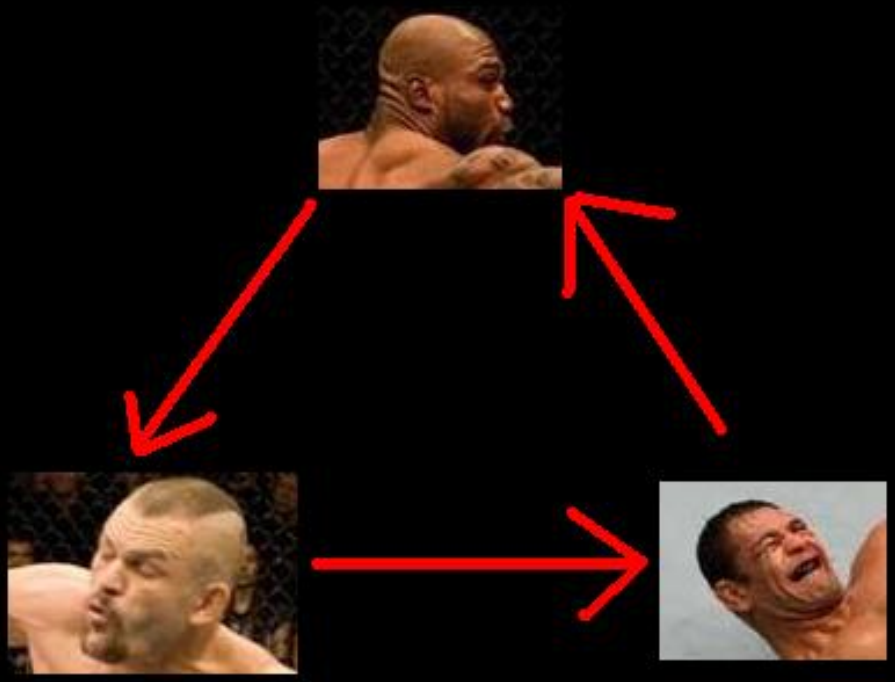
- It's not just how good you are, it's what you're using
- Boxing isn't "better" or "worse" than wrestling, they're just good against different people
- Not everything is equal—Muay Thai is better than Karate—but there are many "good enough" styles

How does this work?


HARLEY-DAVIDSON


MICKEY'S
TAPOUT
SPEED


UFC.COM



Style based model

- Each fighter has one of *n* (some small number) styles
- Within a style, fighters have ratings (good wrestler beats bad wrestler)
- We can assign a probability, *ceteris paribus*, that a fighter with style A beats a fighter with style B
- Combining cross-style probability with in-style ranking, we can predict any pair

Method:

1. Input lists of fighters and fights
2. ???
3. Partition fighters by style
4. Compute style rankings
5. Make predictions
6. Profit!!!

How can we find styles?

First guess would be reported styles, but that has several problems:

1. Styles are self reported—I don't *have* styles for many fighters
2. Styles are self reported—they're often just wrong
3. Named styles are far too general
4. Named styles are influenced by marketing, not facts

How can we find styles?

OBSERVATION:

- The problem with ELO is non-transitivity/cycles
- Liddell -> Ortiz -> Couture -> Liddell
- Within styles, better should beat worse—no cycles!
- Partition fighters to minimize cycles, we should find styles.

Oh god, the pain…

- This is computationally difficult, to say the least
- (Prof Chen and I think) *counting* cycles in a graph is #P-complete
- That's very hard, and in fact difficult to even approximate
- Minimizing cycles is probably much harder

Can we nonetheless use approximation techniques to find a decent partition?

Optimization

I have a good heuristic, now how do I find good partitions?

- Hill climbing is usually nice, right?
- Not here
- Too many neighbors—difficult to actually identify minima!
- Could try something more complicated (simulated annealing?)
- It would be nice if I could modify climbing, which is simple, to work here.

ADD-Search

Like hill-climbing with random restart, *but*:

- Instead of trying all neighbors to a candidate, try *n* of them randomly
- If any of them improve, move there and continue
- If not, get bored, guess that it's a local minima, and return
- Nicely avoids the 50000-adjacency problem
- This has produced some nicely optimized partitions.

Conclusions

- In the future, try to avoid working on computationally intractable problems
- The available data sucks—simultaneously too large and too small
- Prediction on a corpus where participants have > 100 matches is easier than when they have ~ 10