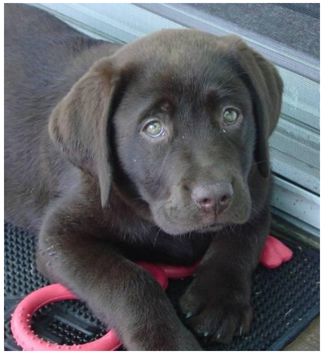
# Our Lab

## CHOCOLATE Lab

Co PIs

David
Fouhey

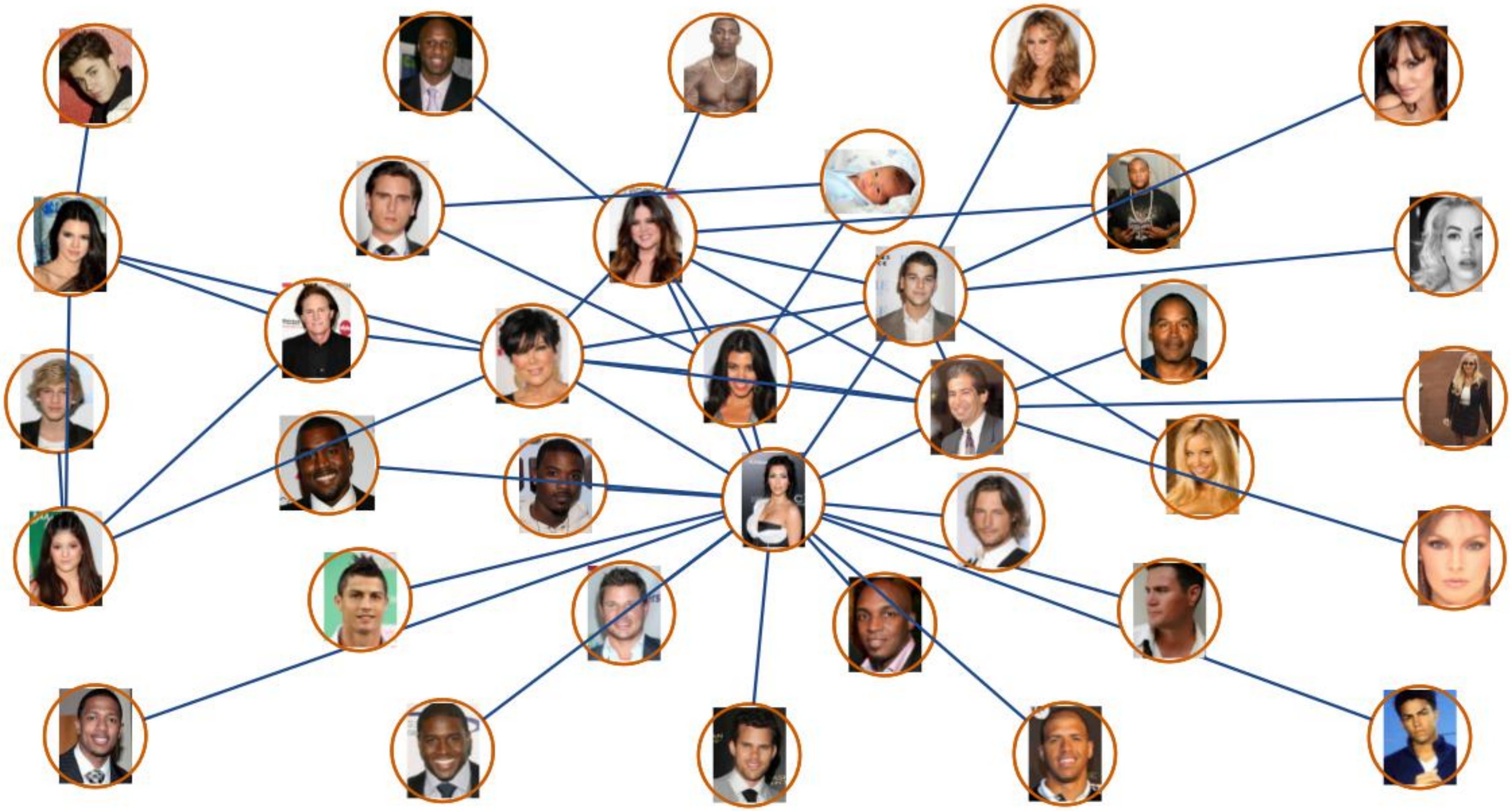
Daniel
Maturana



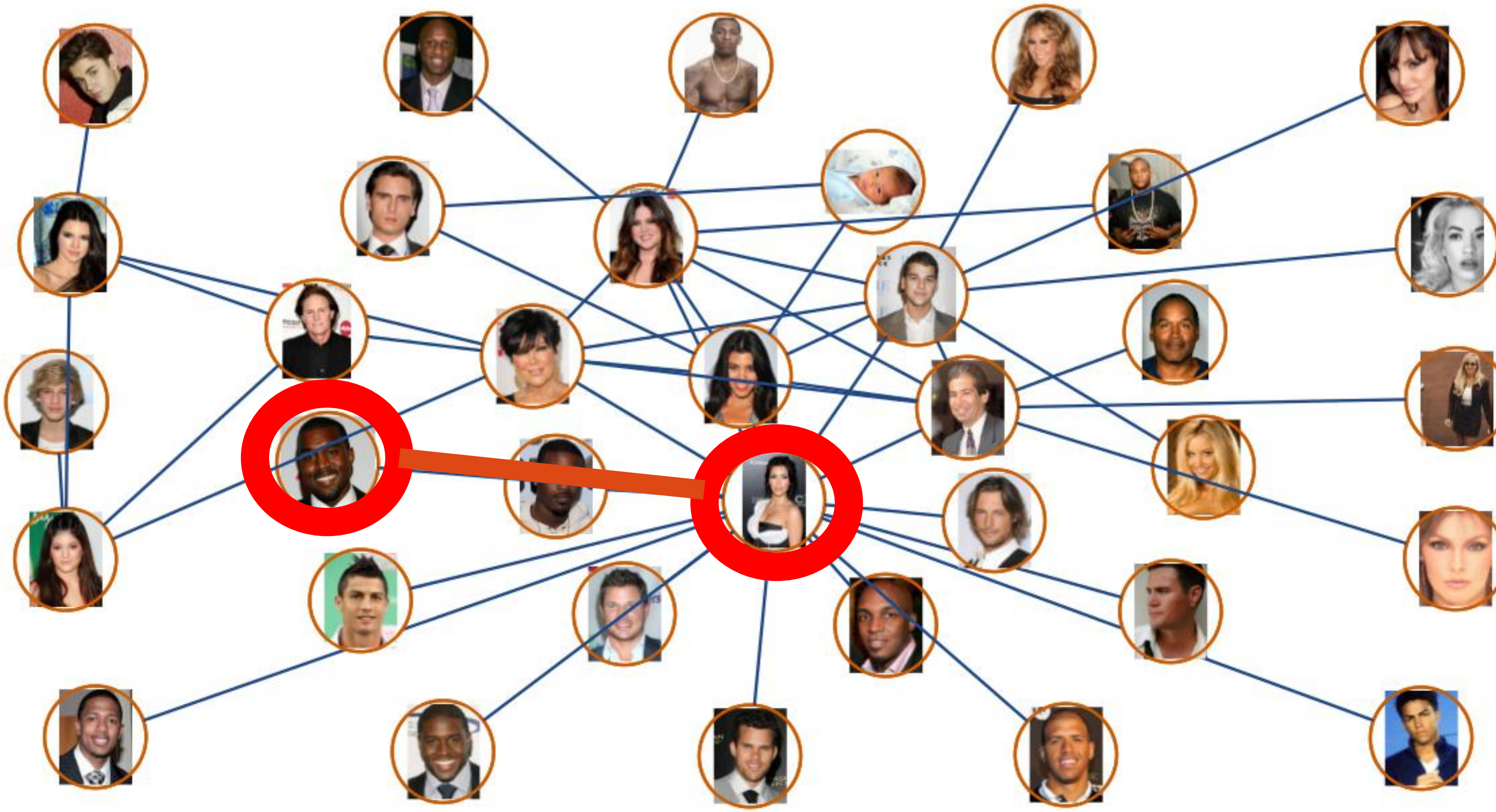
Ruffus
von Woofles

**C**omputational **H**olistic **O**bjective **C**ooperative **O**riented **L**earning **A**rtificial **T**echnology **E**xperts

# Plug: The Kardashian Kernel (SIGBOVIK 2012)

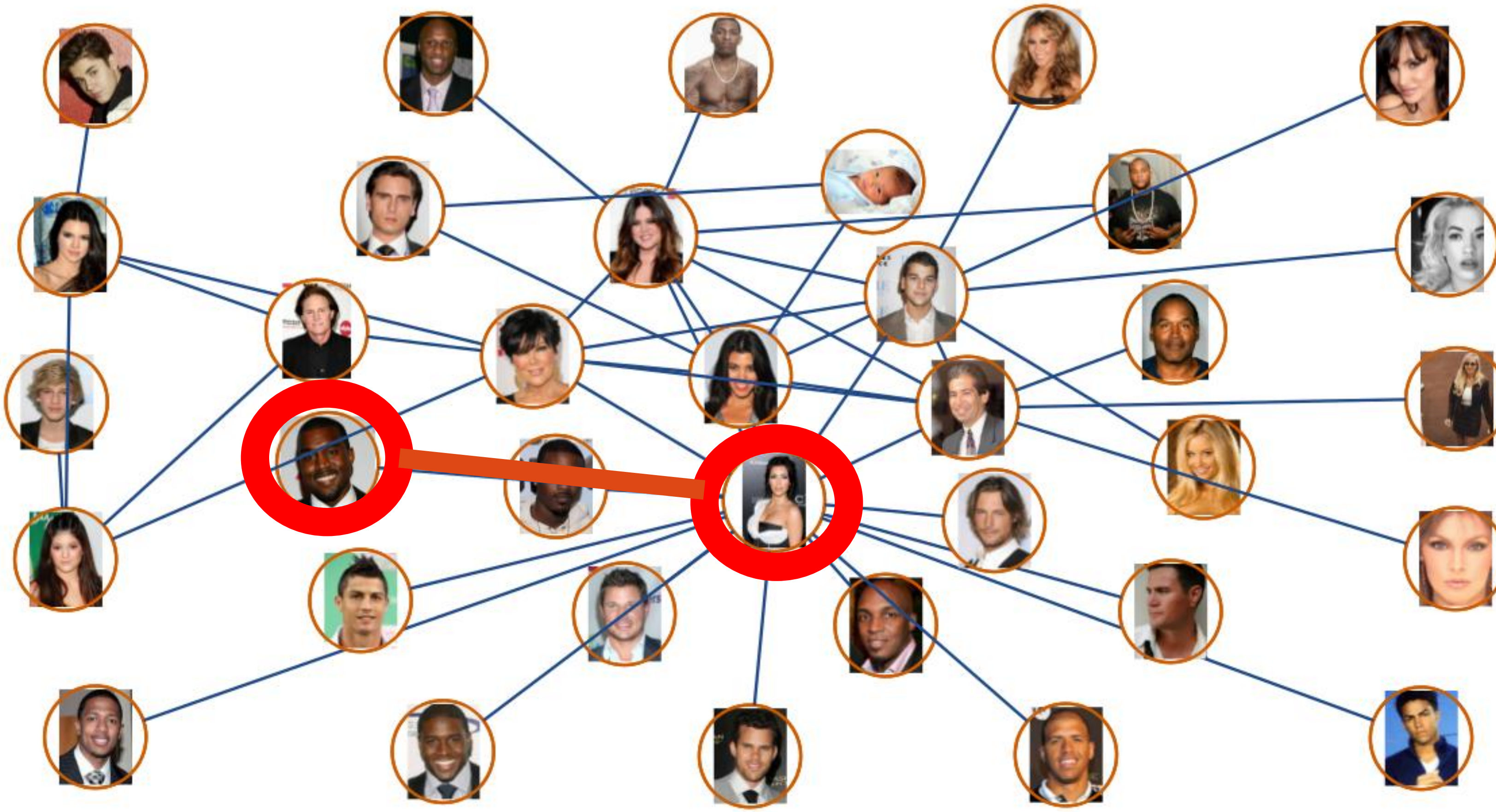


# Plug: The Kardashian Kernel (SIGBOVIK 2012)



Predicted KIMYE March 2012, before anyone else

# Plug: The Kardashian Kernel (SIGBOVIK 2012)



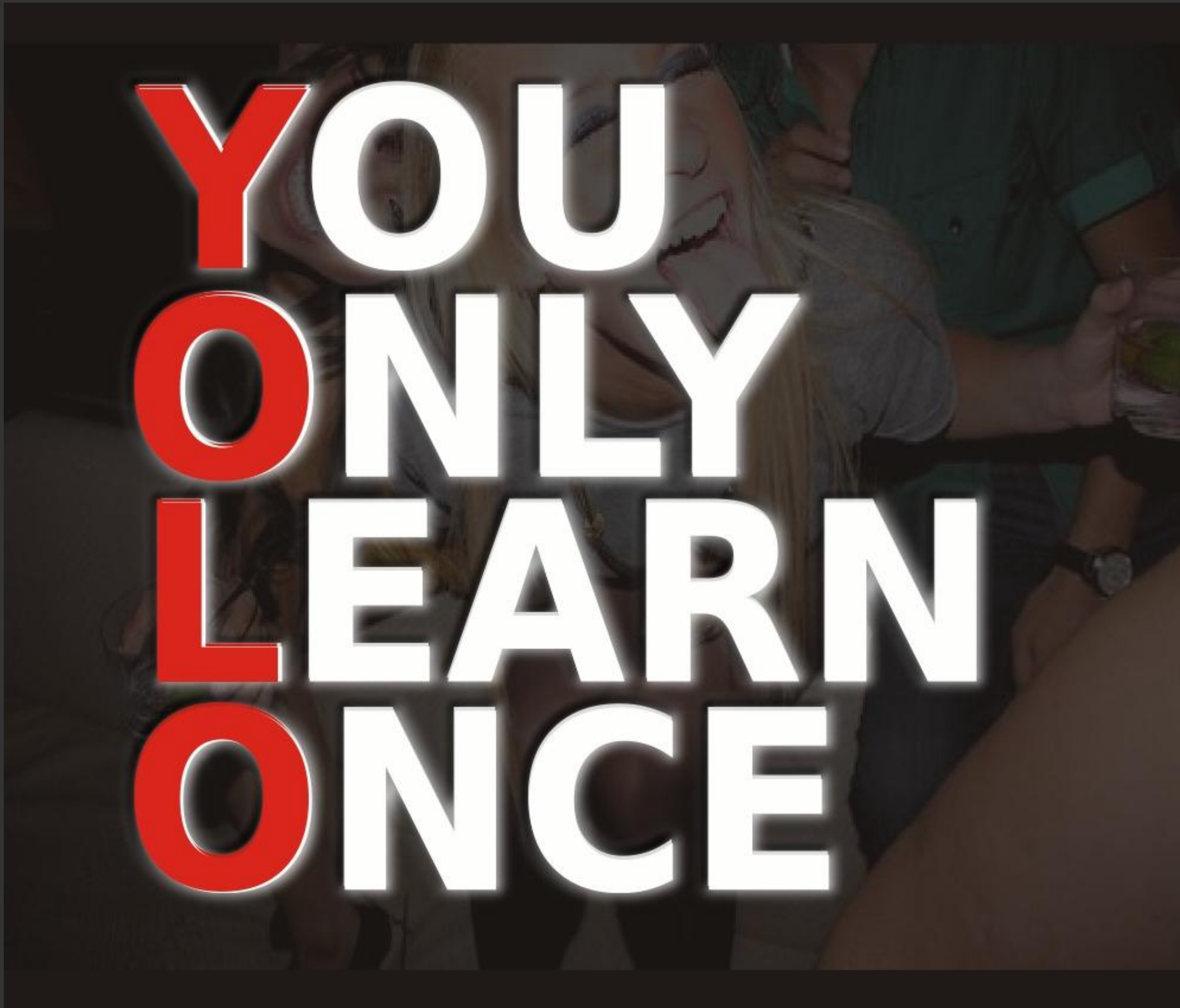
Also confirmed KIM is a reproducing kernel!!

# Minimum Regret Online Learning

Daniel Maturana, David Fouhey
CHOCOLATE Lab – CMU RI

# Minimum Regret Online Learning

- **Online** framework: Only one pass over the data.


YOU
ONLY
LEARN
ONCE

# Minimum Regret Online Learning

- **Online** framework: Only one pass over the data.
- Also online in that it works for #Instagram #Tumblr #Facebook #Twitter

# Minimum Regret Online Learning

- Online framework: Only one pass over the data.
- **Minimum regret:** Do as best as we could've possibly done in hindsight

# Minimum Regret Online Learning

- Online framework: Only one pass over the data.
- **Minimum regret:** Do as best as we could've possibly done in hindsight



# Standard approach: Randomized Weighted Majority (RWM)

**Algorithm 1** Randomized Weighted Majority

initialize regret $R_t \leftarrow 0$
initialize feature weights $w_0 \leftarrow 0$
$t \leftarrow 1$
**for** $t = 1$ to $T$ **do**
  observe data $x_t$
  predict outcome $y_t \leftarrow f(x_t, y_t, w_t)$
  receive loss function $\ell_t(y_t)$
  decrease $w_t$ for erroneous features
  update regret $R_t \leftarrow R_{t-1} + \ell_t(y_t)/T$
  $t \leftarrow t + 1$
**end for**

# Standard approach: Randomized Weighted Majority (RWM)

**Algorithm 1** Randomized Weighted Majority

initialize regret $R_t \leftarrow 0$
initialize feature weights $w_0 \leftarrow 0$
$t \leftarrow 1$
**for** $t = 1$ to $T$ **do**
  observe data $x_t$
  predict outcome $y_t \leftarrow f(x_t, y_t, w_t)$
  receive loss function $\ell_t(y_t)$
  decrease $w_t$ for erroneous features
  update regret $R_t \leftarrow R_{t-1} + \ell_t(y_t)/T$
  $t \leftarrow t + 1$
**end for**

**Regret asymptotically tends to 0**

# Our approach: Stochastic Weighted Aggregation

**Algorithm 2** Stochastically Weighted Aggregation

Initialize regr3tt $R_t \leftarrow 0$
$t \leftarrow 1$
**for** $t = 1$ to *death* **do**
  s0mething happened $x_t$
  post tumblr $y_t \leftarrow f(x_t, y_t, \text{\#so \#random})$
  likes/reblogs/retweets $\ell_t(y_t)$
  chill
  regret!1! $R_t \leftarrow R_{t-1} + \ell_t(y_t)/T$
  $t \leftarrow t + 1$
**end for**

**Regret asymptotically tends to 0**


STOCHASTIC
WEIGHTED
AGGREGATION

# Our approach: Stochastic Weighted Aggregation

**Algorithm 3** Stochastically Weighted Aggregation

Initialize r3gret $R_t \leftarrow 0$
$t \leftarrow 1$
**for** $t = 1$ to *death* **do**
  s0mething happened $x_t$
  post tumblr $y_t \leftarrow f(x_t, y_t, \text{\#so \#random})$
  likes/reblogs/retweets $\ell_t(y_t)$
  regret!1! $R_t \leftarrow R_{t-1} + \ell_t(y_t)/T$
  chill
  ~~update regret~~ $\cancel{R_t \leftarrow R_{t-1} + \ell_t(y_t)}$ $R_t \leftarrow 0$ yolo lol
  $t \leftarrow t + 1$
**end for**

**Regret is instantaneously 0!!!!**

# Generalization To Convex Learning

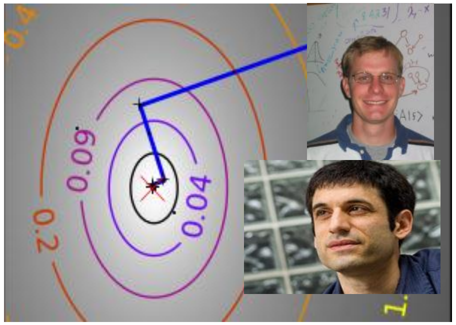
- Total Regret = 0
- For t = 1, …
  - Take Action
  - Compute Loss
  - Take Subgradient
  - Convex Reproject
  - Total Regret += Regret

# Generalization To Convex Learning

- Total Regret = 0
- For t = 1, …
  - Take Action
  - Compute Loss
  - Take **Subgradient**
  - Convex **Reproject**
  - Total Regret += Regret

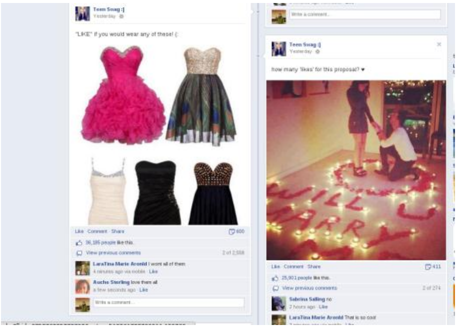
# How to Apply? #enlightened

## “Real World”



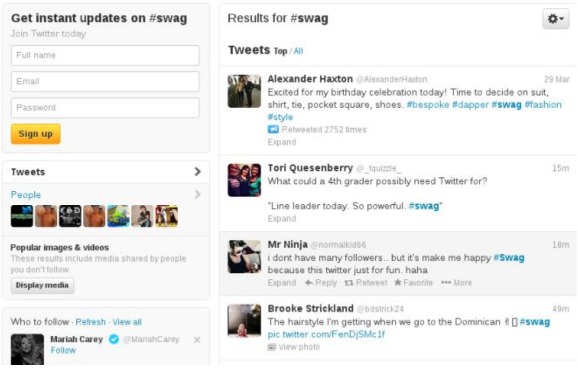

Subgradient = Subgradient
Convex Proj. = Convex Proj.

## Facebook



Subgradient = Like
Convex Proj. = Comment

## Twitter



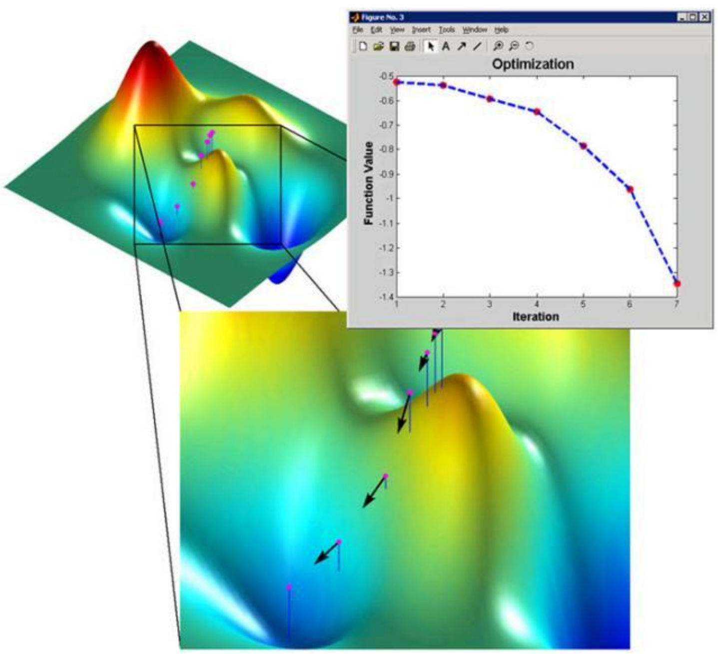

Subgradient = Retweet
Convex Proj. = Reply

## Reddit

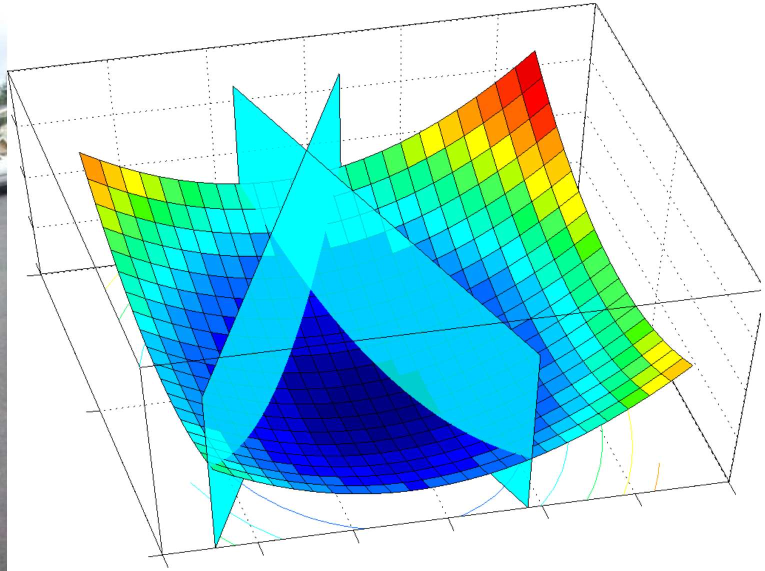

Subgradient = Upvote
Convex Proj. =
Comment

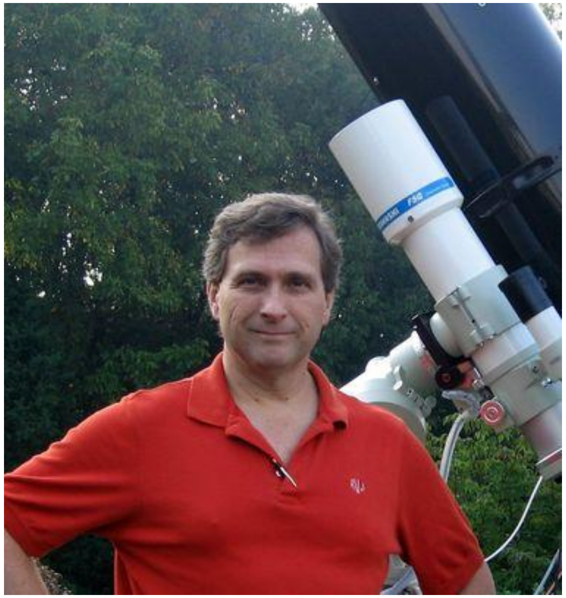
# Another approach #swag #QP

- Convex Programming – don't need quadratic programming techniques to solve SVM




# Head-to-Head Comparison
# SWAG QP Solver vs. QP Solver



4 Loko
Phusion et al. '05

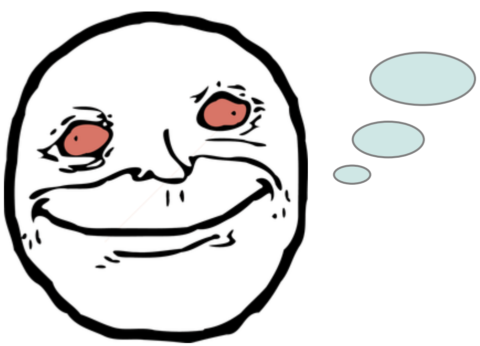
LOQO
Vanderbei et al. '99

# A Bayesian Comparison

| | Solves QPs | Refresh-ing | SWAG | # States Banned | % Alcohol | # LOKOs |
|---|---|---|---|---|---|---|
| 4 LOKO | NO | YES | YES | 4 | 12 | 4 |
| LOQO | YES | NO | NO | 2 | 0 | 1 |

# A Bayesian Comparison

| | Solves QPs | Refresh-ing | SWAG | # States Banned | % Alcohol | # LOKOs |
|---|---|---|---|---|---|---|
| 4 LOKO | NO | YES | YES | 4 | 12 | 4 |
| LOQO | YES | NO | NO | 2 | 0 | 1 |

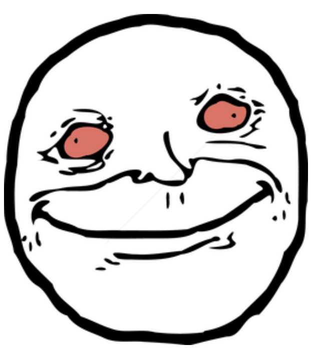


Use ~Max-Ent Prior~
All categories
equally
important.

Le Me, A
Bayesian

# A Bayesian Comparison

| | Solves QPs | Refresh-ing | SWAG | # States Banned | % Alcohol | # LOKOs |
|---|---|---|---|---|---|---|
| 4 LOKO | NO | YES | YES | 4 | 12 | 4 |
| LOQO | YES | NO | NO | 2 | 0 | 1 |



Use ~Max-Ent Prior~
All categories equally important.

Le Me, A Bayesian


Winner!

~~thx lol~~


#SWAGSPACE

**Before:** Minimum Regret Online Learning

**Now:** Cat Basis Purrsuit

# Cat Basis Purrsuit

Daniel Caturana, David Furry
CHOCOLATE Lab, CMU RI

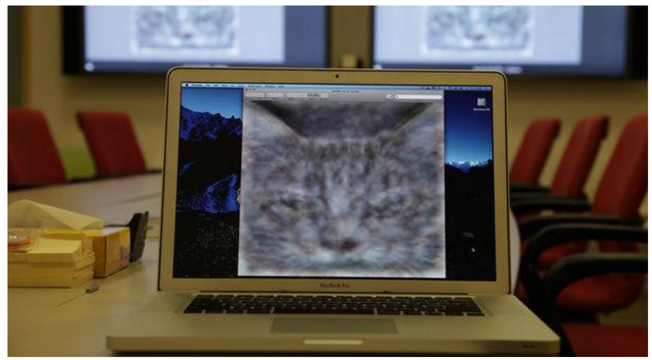
# Motivation

- Everybody loves cats

- Cats cats cats.
- Want to maximize recognition and adoration with minimal work.
- Meow

# Previous work

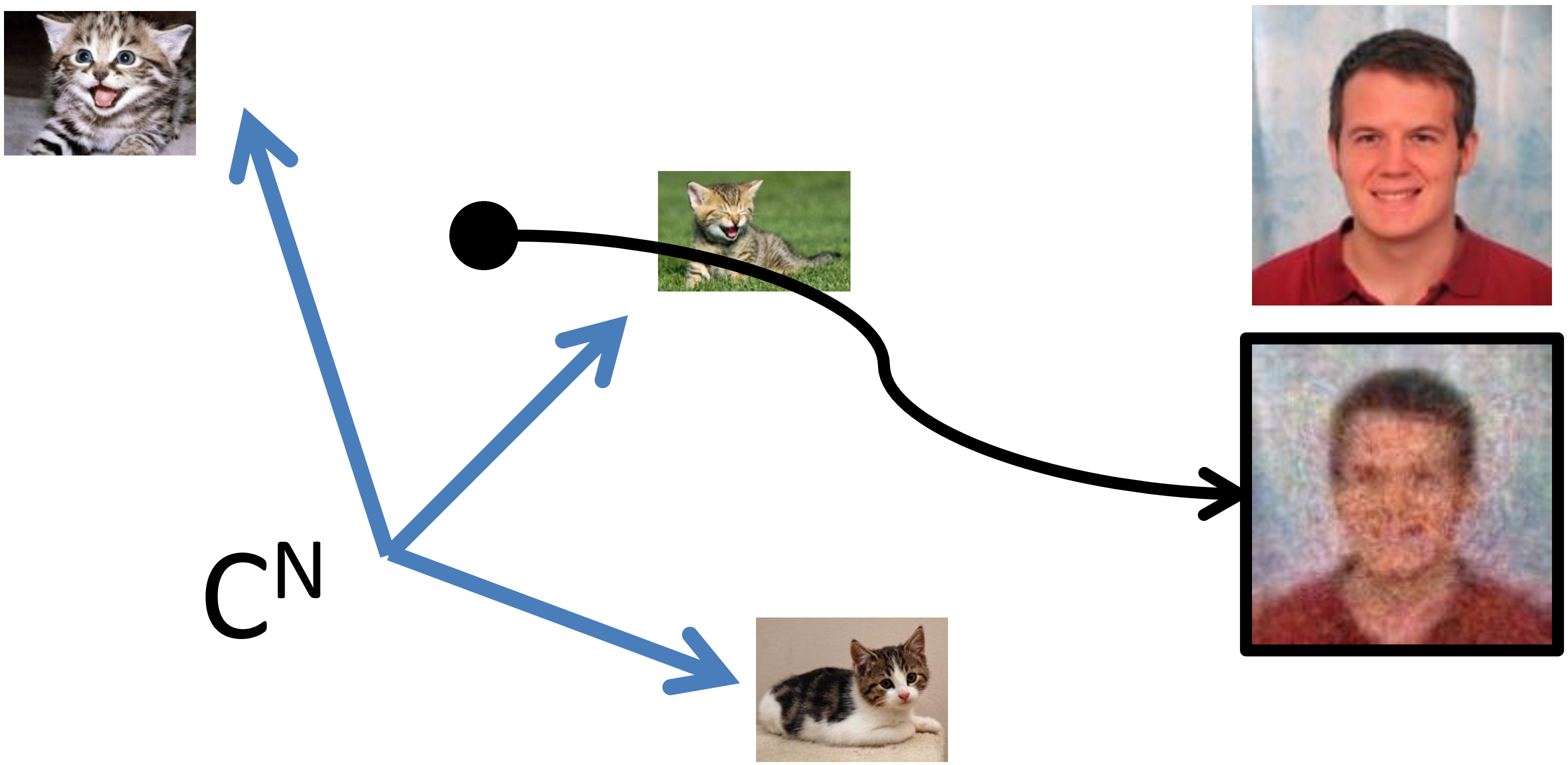
- Google found cats on Youtube [Le et al. 2011]



- Lots of other work on cat detection[Fleuret and Geman 2006, Parkhi et al. 2012].
- Simulation of cat brain [Ananthanarayanan et al. 2009]

# Our Problem

- Personalized cat subspace identification: write a person as a linear sum of cats

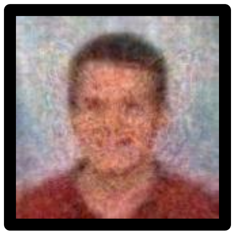


# Why?

- People love cats. Obvious money maker.

# Problem?

- Too many cats are lying around doing nothing.

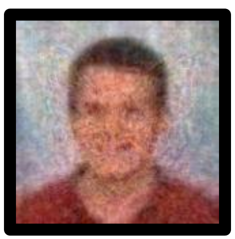

=4.3

+1.2

+0.8

+...+0.01


- Want a spurrse basis (i.e., a sparse basis)


=8.3

+0.0

+1.3

+...+0.00


# Solving for a Spurrse Basis

- Could use orthogonal matching pursuit
- Instead use metaheuristic

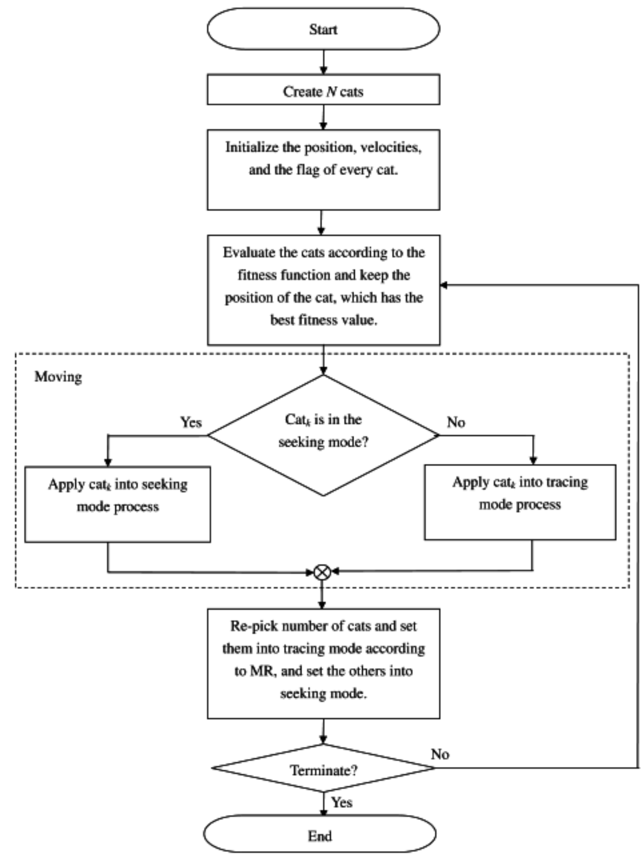
# Solution – Cat Swarm Optimization

## Cat Swarm Optimization

Shu-Chuan Chu[1], Pei-wei Tsai[2], and Jeng-Shyang Pan[2]

[1] Department of Information Management,
Cheng Shiu University
[2] Department of Electronic Engineering,
National Kaohsiung University of Applied Sciences

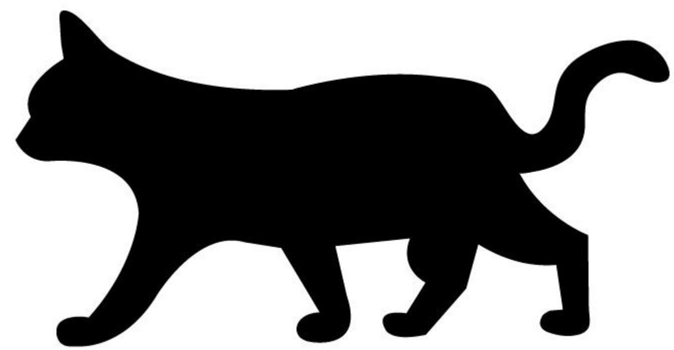
**Abstract.** In this paper, we present a new algorithm of swarm intelligence, namely, Cat Swarm Optimization (CSO). CSO is generated by observing the behaviors of cats, and composed of two sub-models, i.e., tracing mode and seeking mode, which model upon the behaviors of cats. Experimental results using six test functions demonstrate that CSO has much better performance than Particle Swarm Optimization (PSO).



# Cat Swarm Optimization

- Motivated by observations of cats

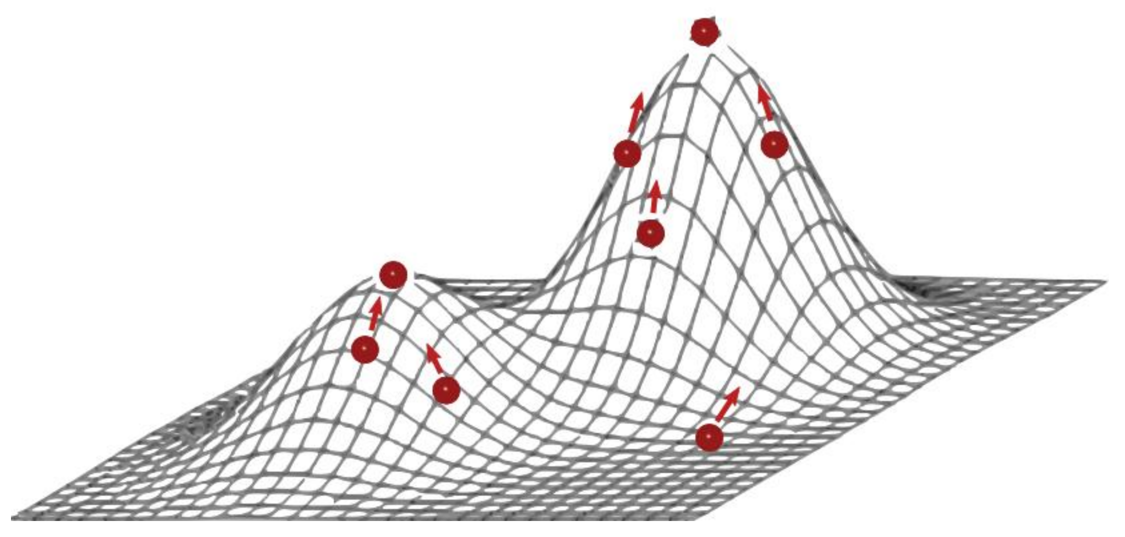
Seeking Mode
(Sleeping and Looking)



Tracing Mode
(Chasing Laser Pointers)

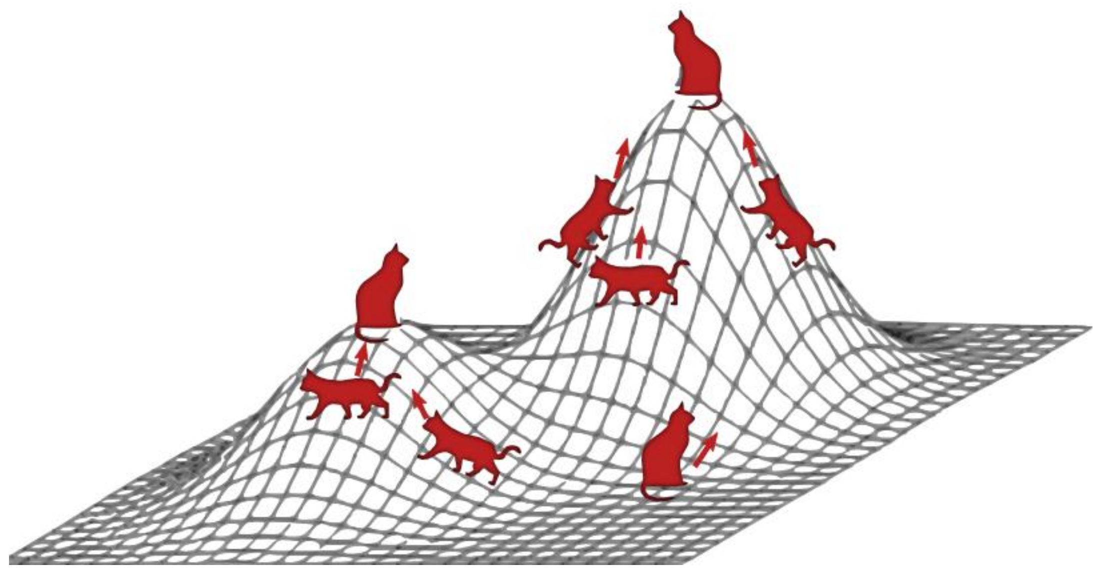
# Cat Swarm Optimization

- Particle swarm optimization
  - First sprinkle particles in an n-Dimensional space



# Cat Swarm Optimization

- Cat swarm optimization
  - First sprinkle cats in an n-Dimensional space*



*Check with your IRB first; trans-dimensional feline projection may be illegal or unethical.

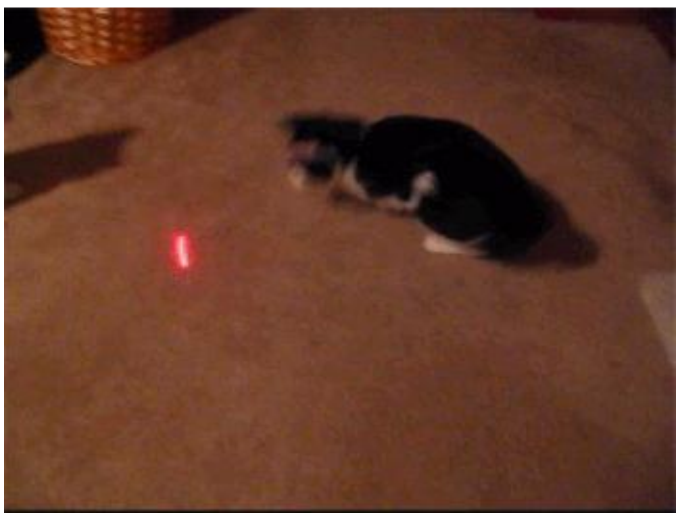
# Seeking Mode

- Sitting around looking at things

Seeking mode at a local minimum

# Tracing Mode

- Chasing after things



In a region of convergence


Scurrying between minima

# Results – Distinguished Leaders



# More Results – Great Scientists



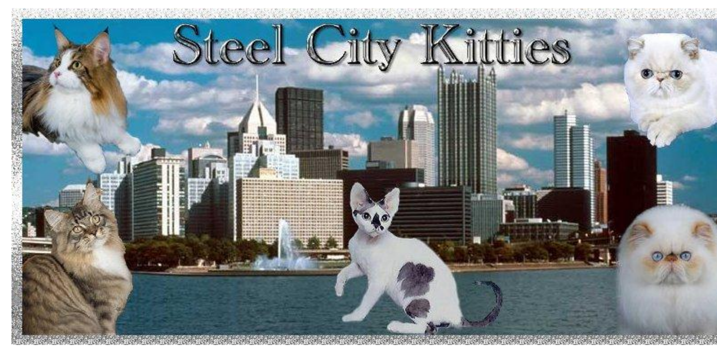
# Future Work

# Future Work

Cat NIPS 2013:

- In conjunction with: NIPS 2013
- Colocated with: Steel City Kitties 2013

N I P S™ Neural Information Processing Systems Foundation

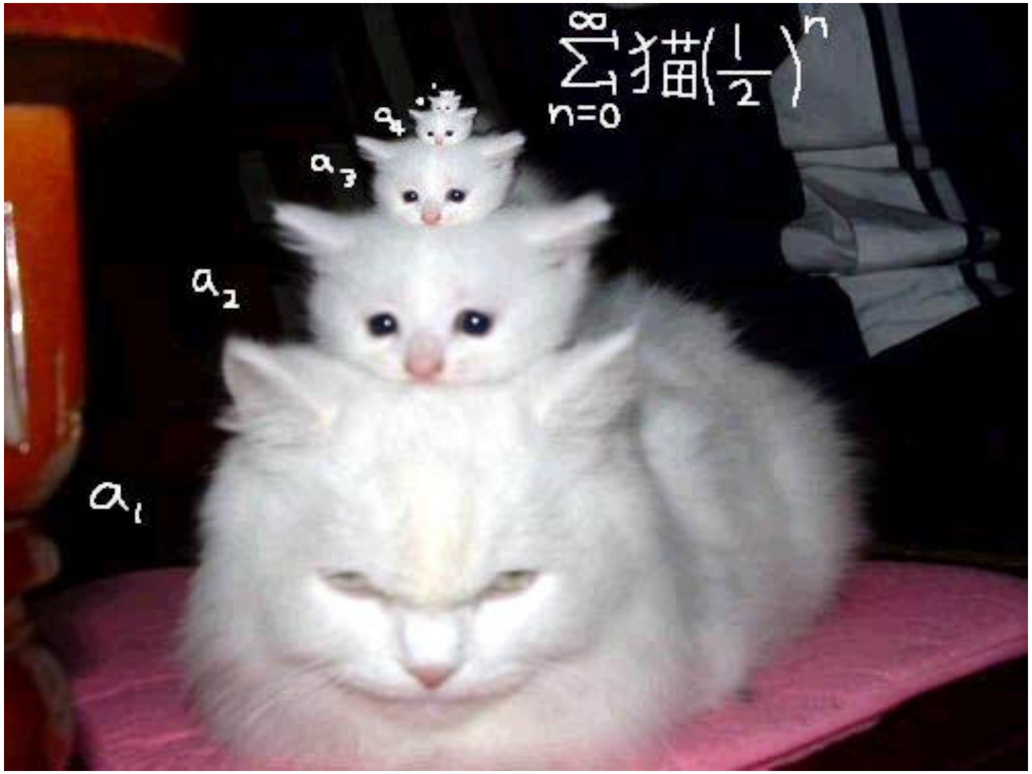


# Future Work

## Deep Cat Basis

# Future Work

## Hierarchical Felines



# Future Work

## Convex Relaxations for Cats

# Future Work

Random Furrests

# Questions?