CSC411 Fall 2015
Machine Learning & Data Mining

Reinforcement Learning II

Slides from Rich Zemel

Formulating Reinforcement Learning

World described by a discrete, finite set of states and actions

At every time step t, we are in a state s_t, and we:

- Take an action a_t (possibly null action)
- Receive some reward r_{t+1}
- Move into a new state s_{t+1}

Decisions can be described by a policy – a selection of which action to take, based on the current state

Aim is to maximize the total reward we receive over time

Sometimes a future reward is discounted by γ^{k-1}, where k is the number of time-steps in the future when it is received

Basic Problems

Markov Decision Problem (MDP): tuple <S,A,P,γ>
where P is

$$P(s_{t+1} = s', r_{t+1} = r' \mid s_t = s, a_t = a)$$

Standard MDP problems:

1. Planning: given complete Markov decision problem as input, compute policy with optimal expected return

2. Learning: Only have access to experience in the MDP, learn a near-optimal strategy

MDP formulation

Goal: find policy π that maximizes expected accumulated future rewards $V^{\pi}(s_t)$, obtained by following π from state s_t:

$$V^{\pi}(s_t) \equiv r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots$$

$$= \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$

Game show example:

- assume series of questions, increasingly difficult, but increasing payoff
- choice: accept accumulated earnings and quit; or continue and risk losing everything

What to Learn

We might try to learn the function V (which we write as V*)

$$V^*(s) = \max_a [r(s,a) + \gamma V^*(\delta(s,a))]$$

We could then do a lookahead search to choose best action from any state s:

$$\pi^*(s) = \arg\max_a [r(s,a) + \gamma V^*(\delta(s,a))]$$

where
$$P(s_{t+1} = s', r_{t+1} = r' \mid s_t = s, a_t = a) =$$
$$P(s_{t+1} = s' \mid s_t = s, a_t = a) P(r_{t+1} = r' \mid s_t = s, a_t = a) =$$
$$\delta(s,a) r(s,a)$$

But there's a problem:

- This works well if we know δ() and r()
- But when we don't, we cannot choose actions this way

What to Learn

Let us first assume that δ() and r() are deterministic:

Remember:

At every time step t, we are in a state s_t, and we:

- Take an action a_t (possibly null action)
- Receive some reward r_{t+1}
- Move into a new state s_{t+1}

Reward function

$$r : (s,a) \rightarrow r$$

$$\delta : (s,a) \rightarrow s$$

Transition function

How can we do learning?

Q Learning

Define a new function very similar to V*

$$Q(s,a) \equiv r(s,a) + \gamma V^*(\delta(s,a))$$

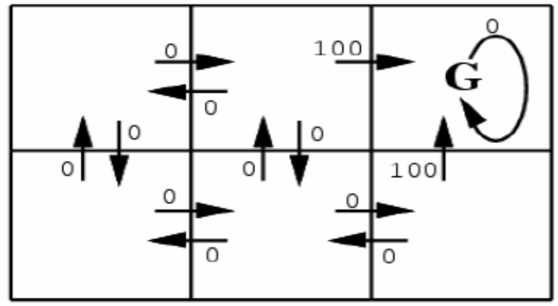
If we learn Q, we can choose the optimal action even without knowing δ!

$$\pi^*(s) = \arg\max_a [r(s,a) + \gamma V^*(\delta(s,a))]$$

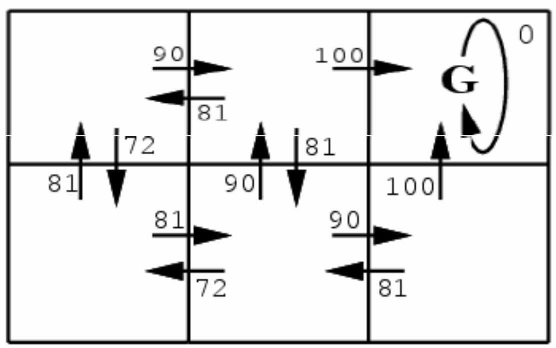
$$\pi^*(s) = \arg\max_a Q(s,a)$$

Q is then the evaluation function we will learn

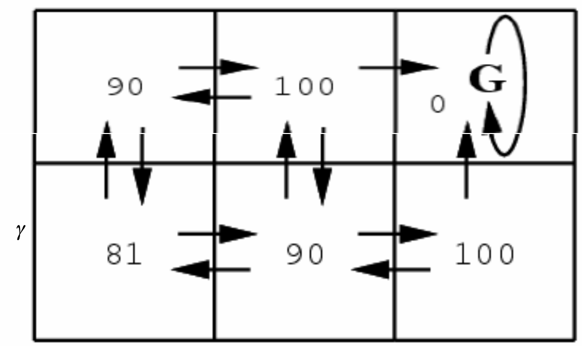
$$\gamma = 0.9$$



$r(s, a)$ (immediate reward) values

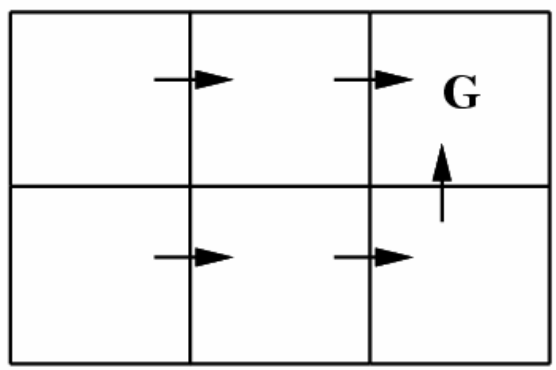


$Q(s, a)$ values



$V^*(s)$ values

$$V^*(s_5) = 0 + \gamma 100 + \gamma^2 0 + \ldots = 90$$



One optimal policy

Training Rule to Learn Q

Q and V* are closely related:

$$V^*(s) = \max_a Q(s,a)$$

So we can write Q recursively:

$$\begin{aligned} Q(s_t, a_t) &= r(s_t, a_t) + \gamma V^*(\delta(s_t, a_t)) \\ &= r(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a') \end{aligned}$$

Let Q^ denote the learner's current approximation to Q

Consider training rule

$$\hat{Q}(s,a) \leftarrow r(s,a) + \gamma \max_{a'} \hat{Q}(s',a')$$

where s' is state resulting from applying action a in state s

Q Learning for Deterministic World

For each *s,a* initialize table entry $Q\hat{}(s,a) \leftarrow 0$

Start in some initial state *s*

Do forever:

- Select an action *a* and execute it
- Receive immediate reward *r*
- Observe the new state *s*'
- Update the table entry for $Q\hat{}(s,a)$ using Q learning rule:

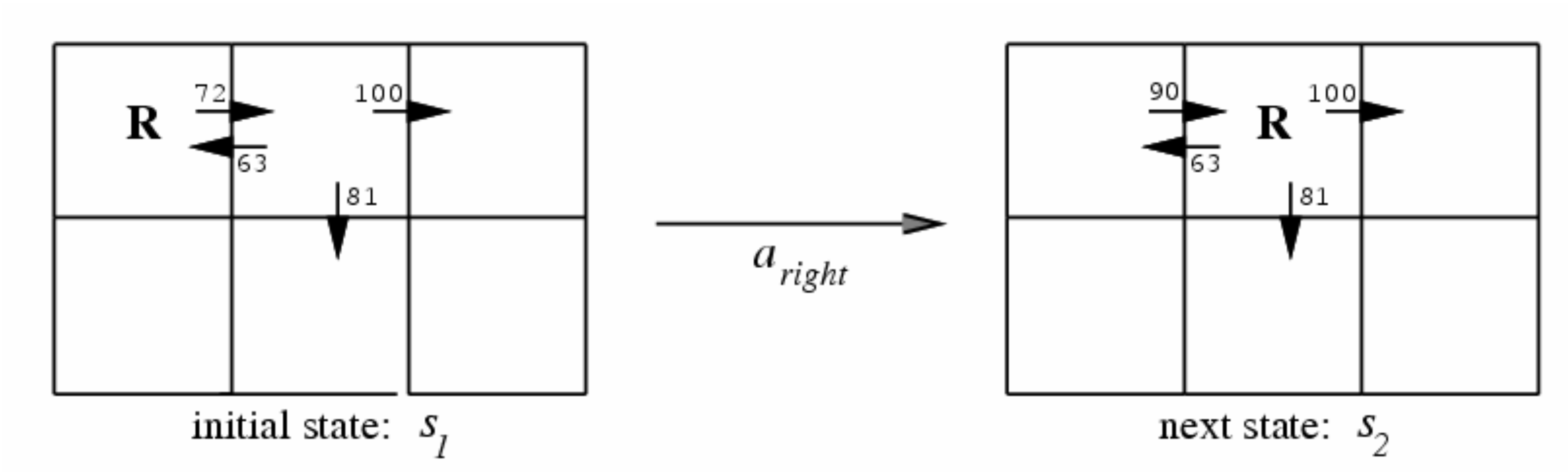
$$\hat{Q}(s,a) \leftarrow r(s,a) + \gamma \max_{a'} \hat{Q}(s',a')$$

- $s \leftarrow s'$

If get to absorbing state, restart to initial state, and run thru "Do forever" loop until reach absorbing state

Updating Estimated Q

Assume Robot is in state s_1; some of its current estimates of Q are as shown; executes rightward move



$$\hat{Q}(s_1, a_{right}) \leftarrow r + \gamma \max_{a'} \hat{Q}(s_2, a')$$

$$\leftarrow r + 0.9 \max_{a'} \{63, 81, 100\} \leftarrow 90$$

Notice that if rewards are non-negative, then Q^ values only increase from 0, approach true Q

Q Learning: Summary

training set consists of series of intervals (episodes): sequence of (state, action, reward) triples, end at absorbing state

Each executed action a results in transition from state s_i to s_j; algorithm updates Q^(s_i,a) using the learning rule

Intuition for simple grid world, reward only upon entering goal state → Q estimates improve from goal state back

1. All Q^(s,a) start at 0
2. First episode – only update Q^(s,a) for transition leading to goal state
3. Next episode – if go thru this next-to-last transition, will update Q^(s,a) another step back
4. Eventually propagate information from transitions with non-zero reward throughout state-action space

Q Learning: Convergence Proof

Q^(s,a) converges to *Q(s,a)*

Consider deterministic world, each (s,a) visited ∞ly often.

Proof: Define full interval as interval during which each (s,a) visited. During each full interval largest error in Q^ table reduced by factor of γ.

Let $\hat{Q}_n$ be table after n updates, Δ_n be max. error in $\hat{Q}_n$

$$\Delta_n = \max_{s,a} |\hat{Q}(s,a) - Q(s,a)|$$

Q Learning: Convergence Proof

Let Q^$_n$ be table after n updates, Δ_n be max. error in Q^$_n$

$$\Delta_n = \max_{s,a} |\hat{Q}(s,a) - Q(s,a)|$$

For any entry updated on interval n+1, error in new estimate:

$$
\begin{aligned}
|\hat{Q}_{n+1}(s,a) - Q(s,a)| &= |(r + \gamma \max_{a'} \hat{Q}_n(s',a')) - (r + \gamma \max_{a'} Q(s',a'))| \\
&= \gamma\, |\max_{a'} \hat{Q}_n(s',a') - \max_{a'} Q(s',a')| \\
&\le \gamma \max_{a'} |\hat{Q}_n(s',a') - Q(s',a')| \\
&\le \gamma \max_{s'',a'} |\hat{Q}_n(s'',a') - Q(s'',a')| \le \gamma \Delta_n
\end{aligned}
$$

Q Learning: Convergence Proof (cont.)

Largest error in initial table is bounded, since values of Q_n^(s,a) and Q(s,a) are bounded for all s,a

Largest error in table after one interval will be at most $\gamma\Delta_0$

After k intervals, error will be at most $\gamma^k\Delta_0$

Since $0 \leq \gamma, < 1$ error → 0 as n → ∞

Q Learning: Exploration/Exploitation

Have not specified how actions chosen (during learning)

Can choose actions to maximize Q^(s,a)

Good idea?

Can instead employ stochastic action selection (policy):

$$P(a_i \mid s) = \frac{\exp(k\hat{Q}(s,a_i))}{\sum_j \exp(k\hat{Q}(s,a_j))}$$

Can vary *k* during learning – more exploration early on, shift towards exploitation

Nondeterministic Case

What if reward and next state are non-deterministic?

We redefine V,Q based on probabilistic estimates, expected values of them:

$$V^{\pi}(s) \equiv E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots]$$

$$= E[\sum_{i=0}^{\infty} \gamma^i r_{t+i}]$$

$$Q(s,a) \equiv E[r(s,a) + \gamma V^*(\delta(s,a))]$$

$$= E[r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) \max_{a'} Q(s',a')]$$

Nondeterministic Case: Learning Q

Training rule does not converge (can keep changing Q^ even if initialized to true Q values)

So modify training rule to change more slowly

$$\hat{Q}_n(s,a) \leftarrow (1-\alpha_n)\hat{Q}_{n-1}(s,a) + \alpha_n[r + \gamma \max_{a'} \hat{Q}_{n-1}(s',a')]$$

where s’ is the state land in after s, and a’ indexes the actions that can be taken in state s’

$$\alpha_n = 1/(1 + visits_n(s,a))$$

where visits is the number of times action a is taken in state s

Summary

- What to study?
  - Material covered in lectures and tutorial
  - Use the books/readings as back-up, to help understand the methods and derivations

- Focus mainly on material since the mid-term

- The exam is closed book and notes
  - Do not focus on memorizing formulas, but instead main ideas and methods

Topics to Study

- Unsupervised Learning
  - what is the difference between hard/soft clustering?
  - Gaussian mixture models / EM:
    - what is a mixture?
    - what does it mean that this is a generative model?
    - what is E step?
    - what is M step?
    - EM vs. gradient descent?
    - is convergence guaranteed?
    - what are responsibilities?
    - understand (but not memorize) eqns, objective
  - PCA and autoencoders:
    - what is PCA used for?
    - what is the objective function(s)?
    - what is a principal component?
    - PCA vs. clustering?
    - How does PCA compare to autoencoders

Topics to Study (cont.)

- Support Vector Machines
    - what is the kernel trick?
    - when can the kernel trick be applied?
    - what is its purpose
    - how is an SVM similar and different than a linear classifier?
    - what is a support vector?
    - What is the objective function?
    - Primal vs. dual formulation
- Reinforcement Learning
    - Compare to other forms of learning
    - Q learning algorithm: updates, objective
    - Exploration/exploitation

Topics to Study (cont.)

Ensemble Methods

- Basic motivation, approach
- Bagging, boosting – compare and contrast
- AdaBoost: steps of algorithm
- Mixture of experts: compare/contrast to others

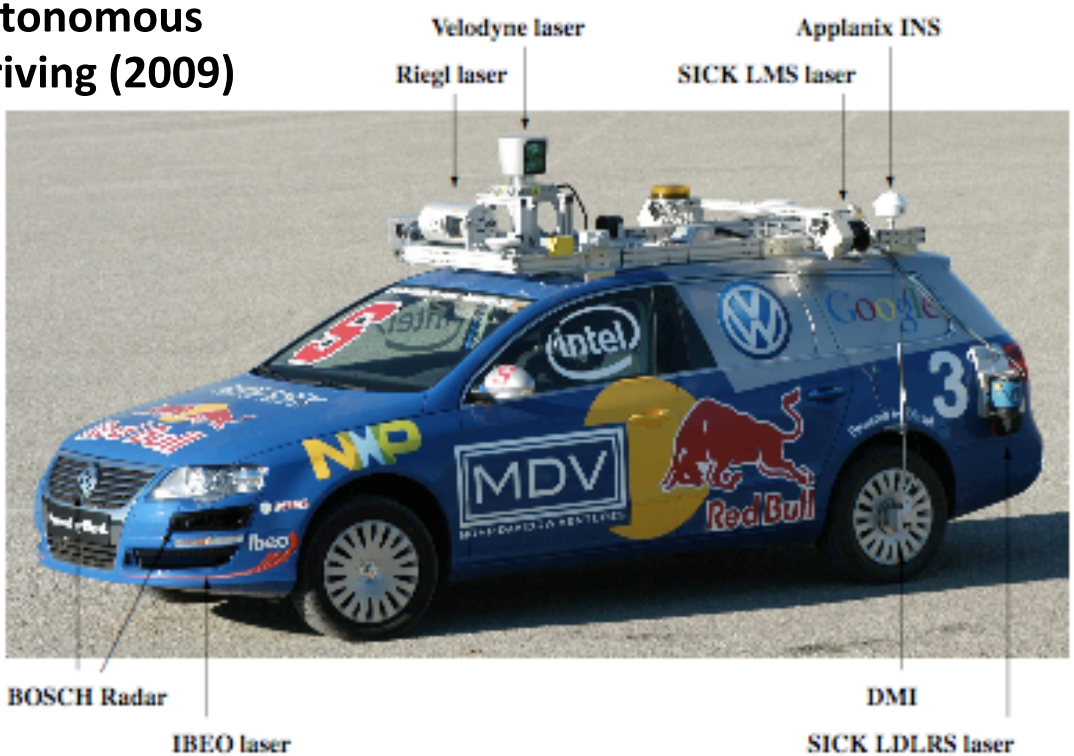
Bayesian Methods

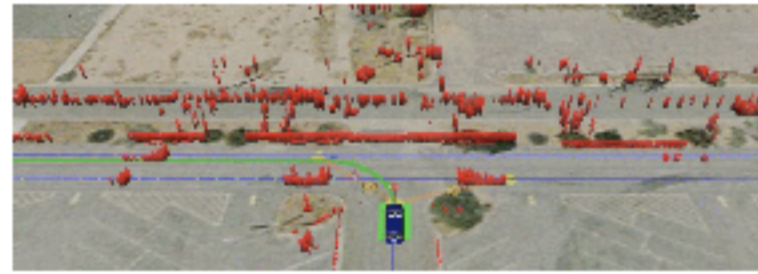
- Motivation
- Posterior predictive distribution
- Learning & prediction

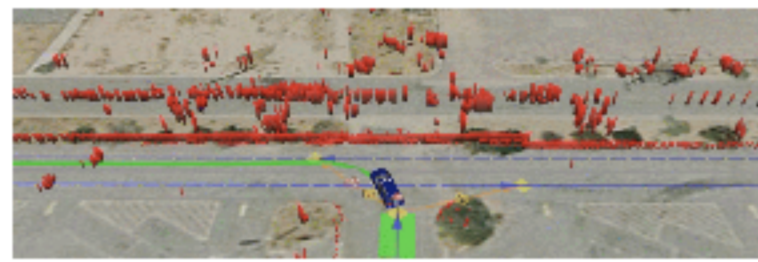
Future Looks Bright

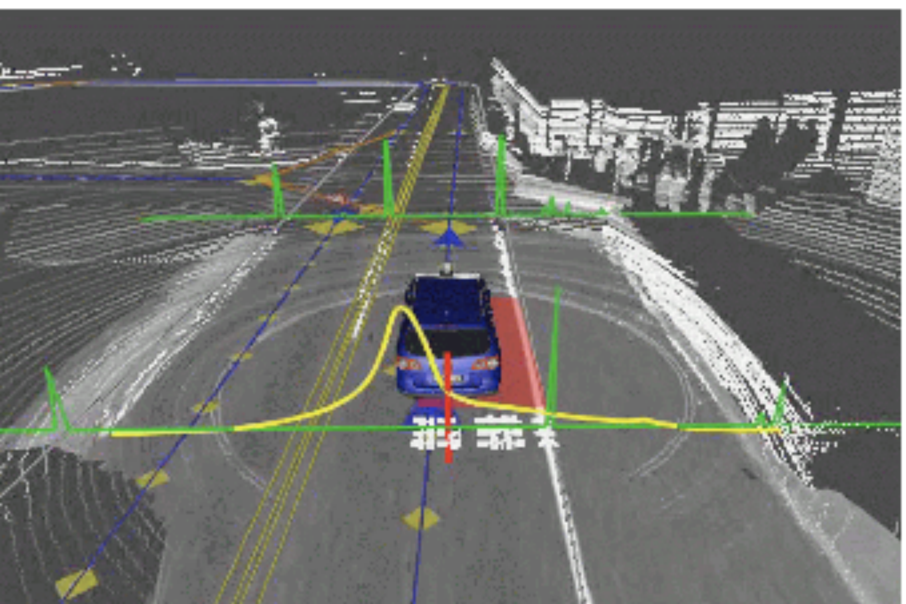
- Data is everywhere! It's an exciting time to know how to make the most of it.
  - Internet
  - Web traffic
  - Store purchases
  - Online ads
  - Social connections (Facebook, Twitter, etc)
  - Etc., etc., etc., etc., …
  - Robotics and Computer Vision
  - Images, videos, range scans

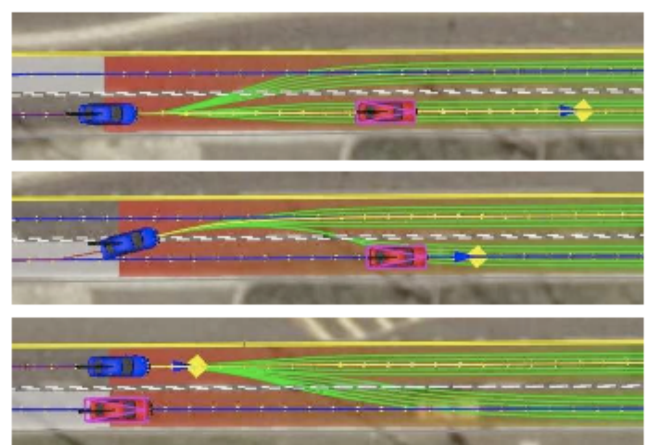
Autonomous Driving (2009)



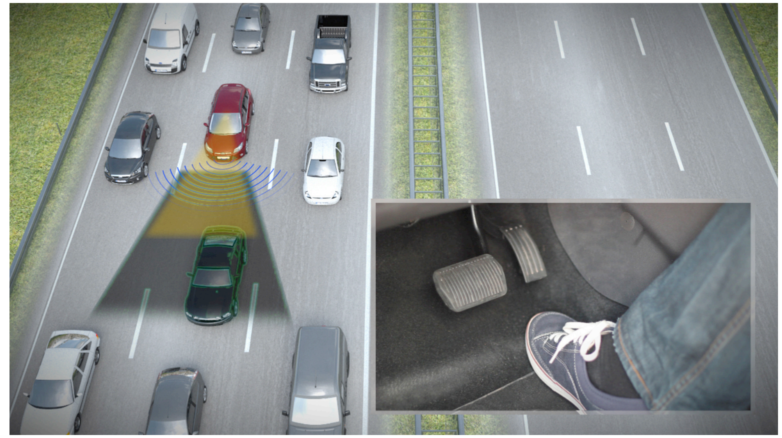








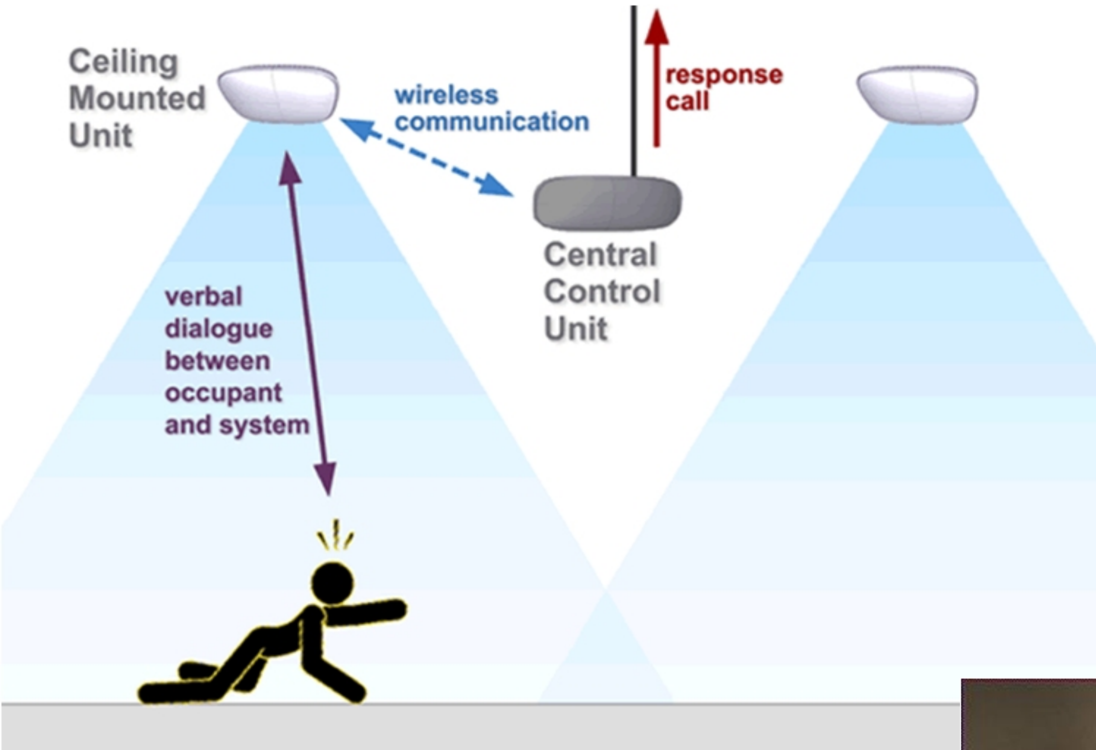
Autonomous driving (2012)



Videos:

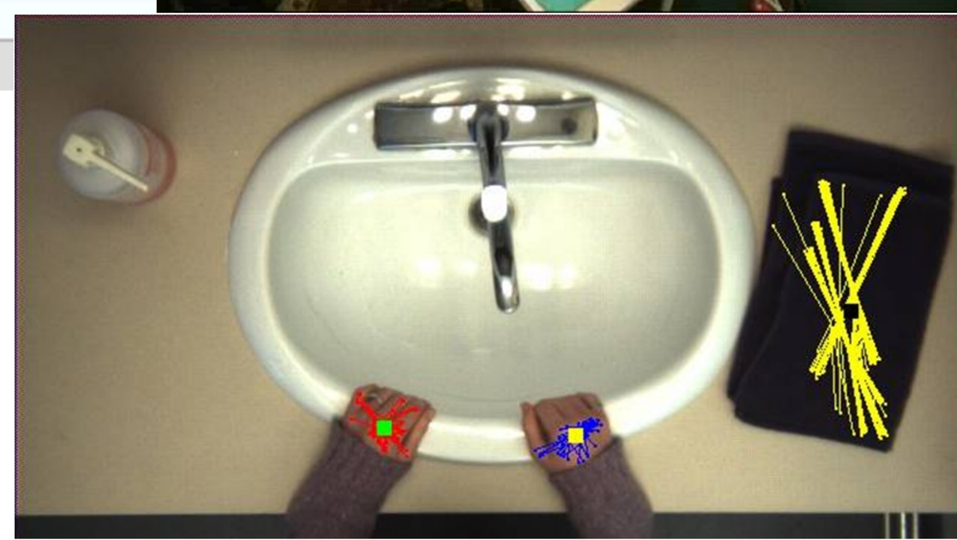
- [- Google car touring]
- [- Google car racing]

Assistive Technology



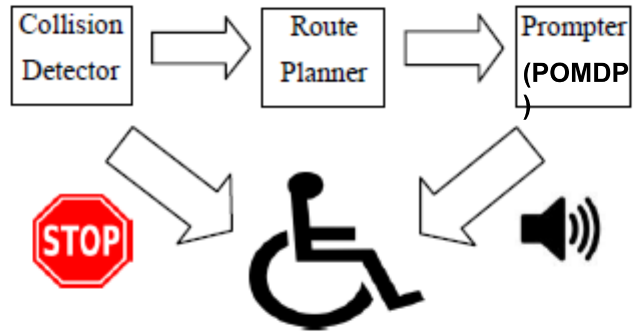
Fall Detection

Hand Washing



Intelligent Assistive Technology and Systems Lab
University of Toronto

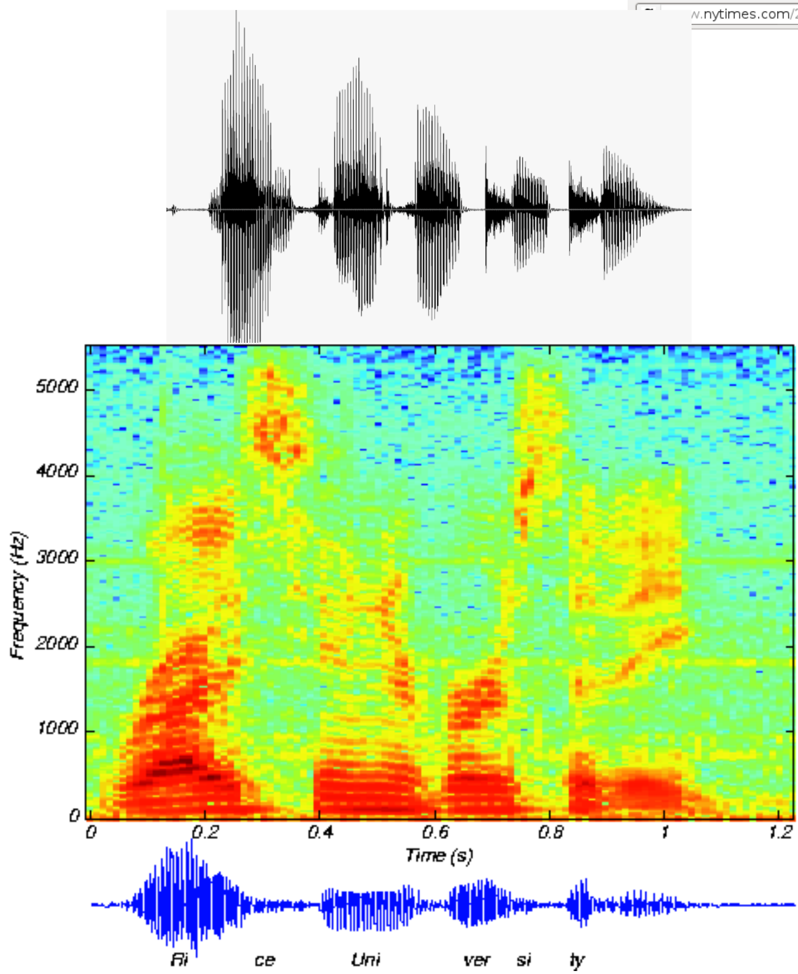
Navigation and Obstacle Avoidance Help

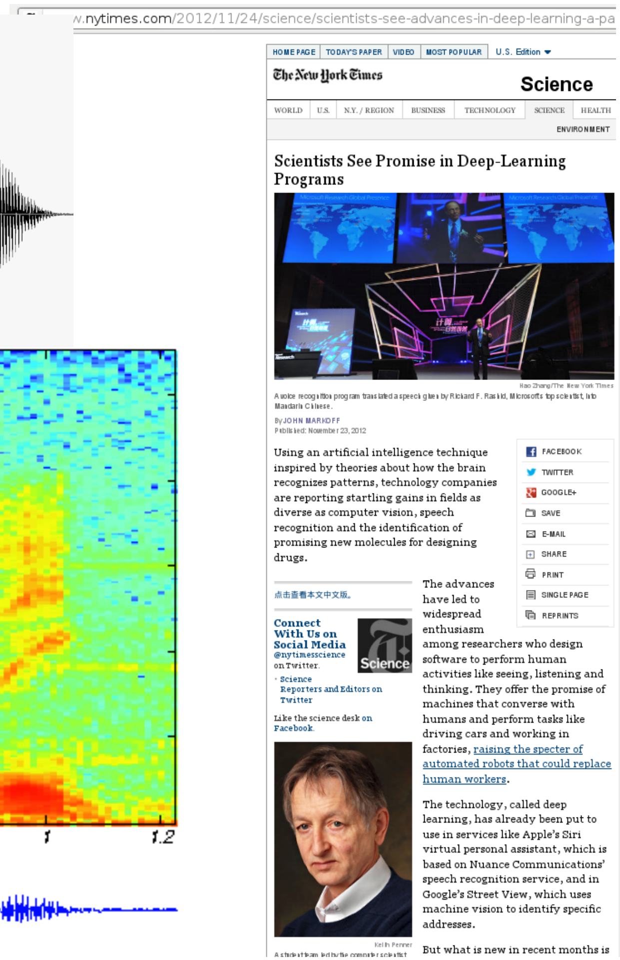


System prevented user from driving into detected obstacles, audio prompts for wayfinding assistance ("off-route – turn left!", "move forward", etc.)

Tested with six cognitively-impaired older adults in Toronto: Single-Subject Research Design: A-B (B-A) trials with training session prior to each phase

Speech Recognition (thanks to deep learning)




w.nytimes.com/2012/11/24/science/scientists-see-advances-in-deep-learning-a-pa

HOME PAGE | TODAY'S PAPER | VIDEO | MOST POPULAR | U.S. Edition

The New York Times — Science

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | ENVIRONMENT

Scientists See Promise in Deep-Learning Programs

By JOHN MARKOFF
Published: November 23, 2012

Using an artificial intelligence technique inspired by theories about how the brain recognizes patterns, technology companies are reporting startling gains in fields as diverse as computer vision, speech recognition and the identification of promising new molecules for designing drugs.

FACEBOOK | TWITTER | GOOGLE+ | SAVE | E-MAIL | SHARE | PRINT | SINGLE PAGE | REPRINTS

点击查看本文中文版。

Connect With Us on Social Media
@nytimesscience on Twitter.
· Science Reporters and Editors on Twitter
Like the science desk on Facebook.

The advances have led to widespread enthusiasm among researchers who design software to perform human activities like seeing, listening and thinking. They offer the promise of machines that converse with humans and perform tasks like driving cars and working in factories, raising the specter of automated robots that could replace human workers.

The technology, called deep learning, has already been put to use in services like Apple's Siri virtual personal assistant, which is based on Nuance Communications' speech recognition service, and in Google's Street View, which uses machine vision to identify specific addresses.

But what is new in recent months is


THE NEW YORKER

SUBSCRIBE | THIS WEEK'S ISSUE | NEWS | CULTURE | POLITICS | BOOKS & FICTION

DOUBLE TAKE | PHOTO BOOTH | DAILY SHOUTS | PAGE-TURNER | DAILY COMMENT | AMY DAVIDSON

THE NEW YORKER | ONLINE ONLY

NEWS DESK
Reporting the latest on Washington and the world.

« How Susan Rice Sees the World | Main | Moral Machines »

NOVEMBER 25, 2012

IS "DEEP LEARNING" A REVOLUTION IN ARTIFICIAL INTELLIGENCE?

POSTED BY GARY MARCUS

Recommend 598 | Tweet 317 | +1 89

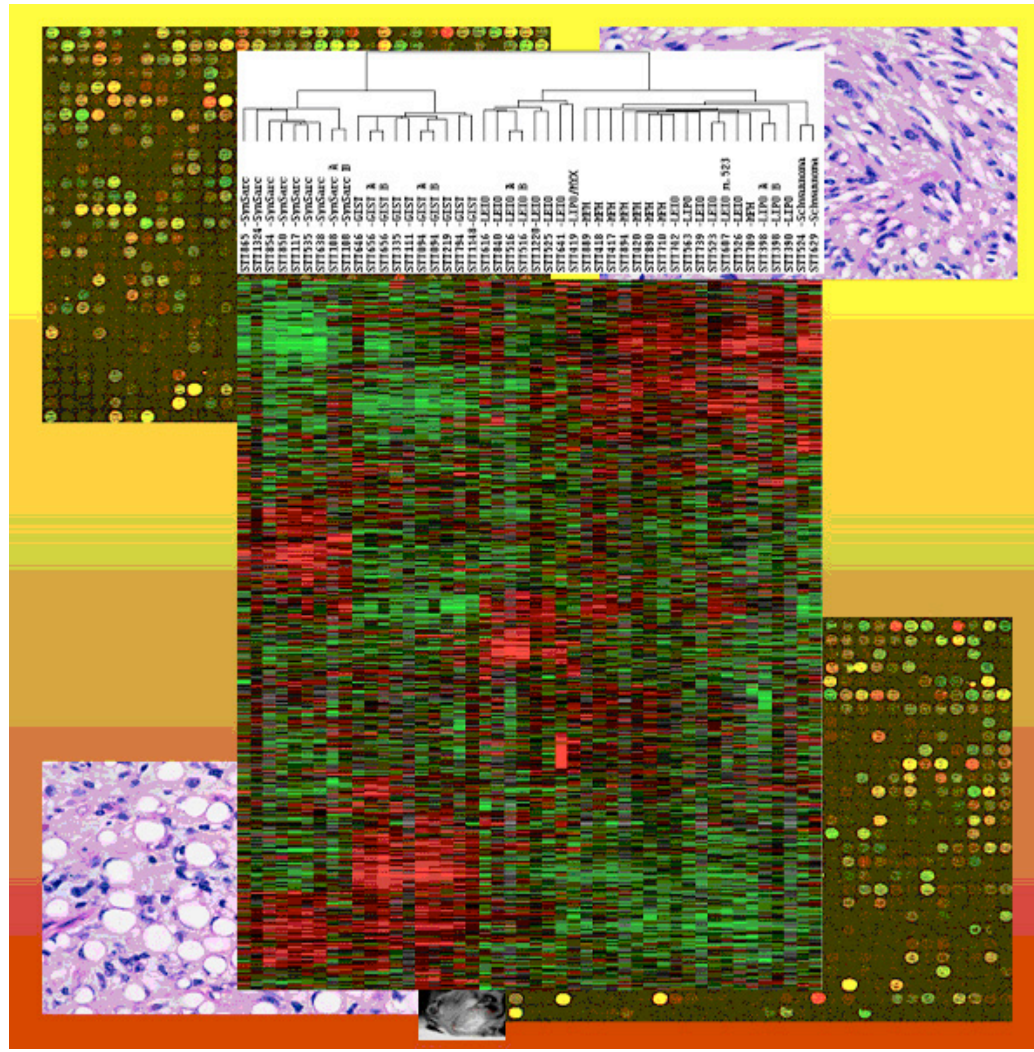
Can a new technique known as deep learning revolutionize artificial intelligence, as yesterday's front-

FOLLOW @TNYNEWSDESK ON TWITTER
E-MAIL | RSS
JOHN CASSIDY — TWITTER | RSS
AMY DAVIDSON — E-MAIL | TWITTER | RSS
HENDRIK HERTZBERG — E-MAIL | TWITTER | RSS
ALEX KOPPELMAN — E-MAIL | TWITTER
EVAN OSNOS — E-MAIL | TWITTER |

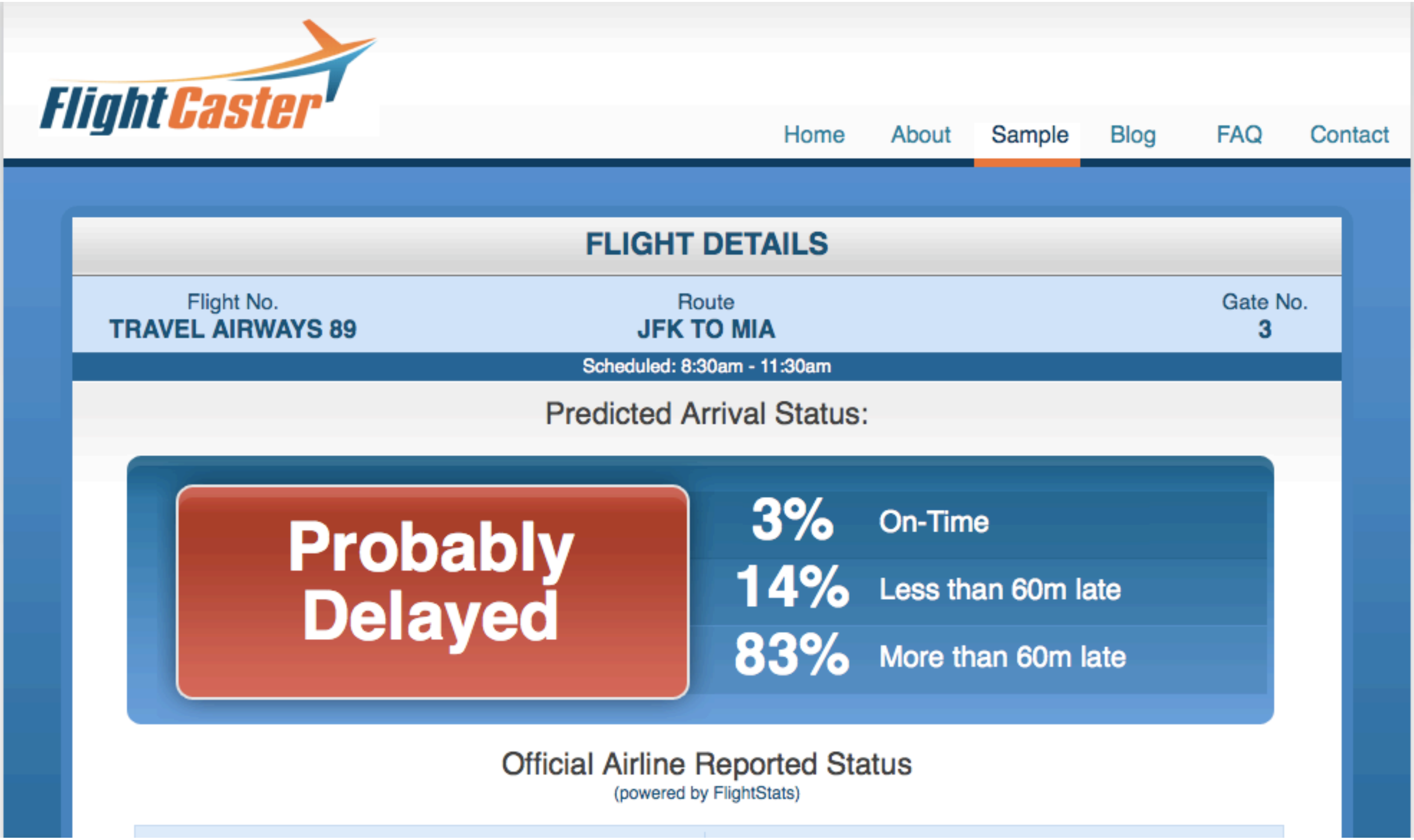
4. Chinese Media Retreat After Reports of Unexpected 'Black Jail' Verdict
5. GADGETWISE — Q&A: Opening a New Page in a Browser Tab
6. DOT EARTH — Exploring Humanity's Evolving 'Global Brain'
7. For Second Opinion, Consult a Computer?
8. OPINION — Many More Images, Much Less Meaning
9. UNBOXED — Taking a Stand for Office Ergonomics
10. MEDIA DECODER — Book by 2 From Google Takes a Deep Look at the Web

Computational Biology

- •Protein folding
- •Gene expression
- •HIV/AID vaccines
- •Machine Learning
- •in Comp. Biology
  Workshops at NIPS
- •Etc.



Flight Delays



Political Campaigns

...In our own campaign, polling was just one way we viewed how we were doing in a state in the general election. We had a lot of voter identification work. We had a lot of field data. So we'd put all that together and model out the election in those states every week. So we'd say, okay, if the election were held this week **based on all our data, put it all in a blender, where are we?** ...**It makes you enormously agile.**

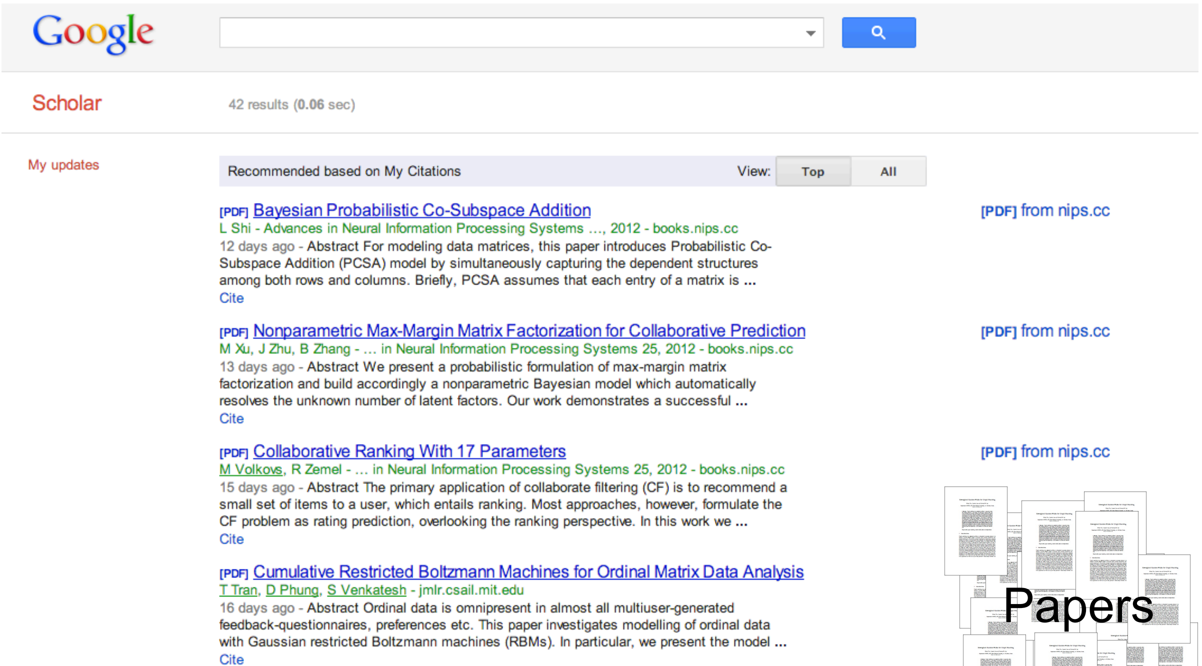
-David Plouffe, Campaign Manager, Obama for America 2008

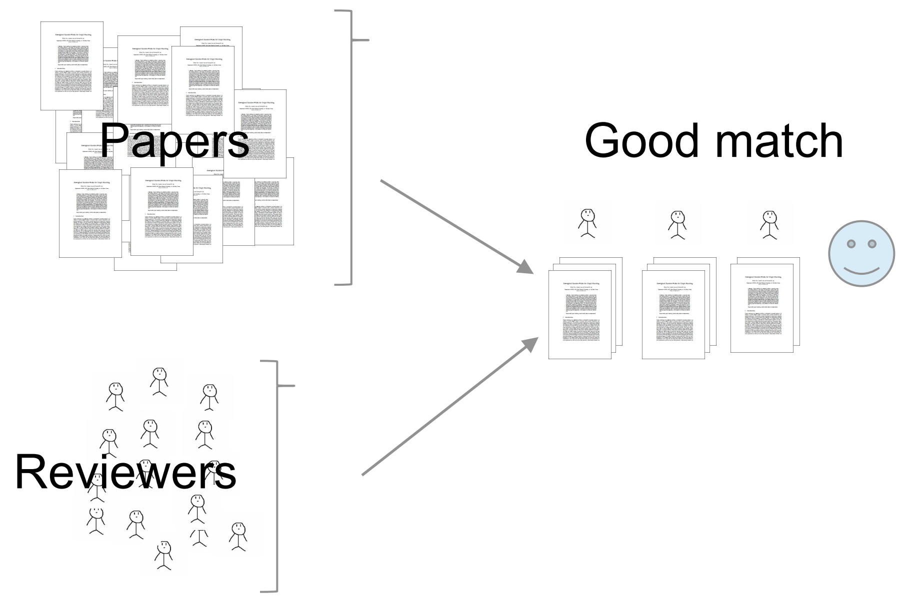
Video: How We Used Data to Win the Presidential Election
– Dan Siroker, Director of Analytics for the 2008 Obama Presidential Campaign

… We could [predict] people who were going to give online.
We could model people who were going to give through mail.
We could model volunteers," said one of the senior advisers about the predictive profiles built by the data. "In the end, modeling became something way bigger for us in '12 than in '08 because it made our time more efficient…

-Senior adviser to the Obama 2012 campaign

Paper recommendations





Machine Learning for Sustainability

- Emerging topic (NIPS Mini Symposium)
  - **Machine learning for the NYC power grid: lessons learned and the future**
  - **What it takes to win the carbon war. Why even AI is needed.**
  - **Ecological Science and Policy: Challenges for Machine Learning**
  - **Optimizing Information Gathering in Environmental Monitoring**
  - **Approximate Dynamic Programming in Energy Resource Management**