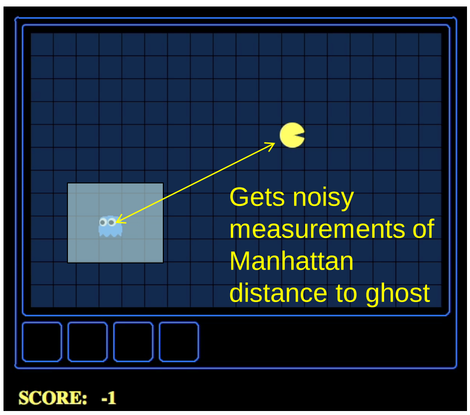
CSE 473

Lecture 23
(Chapters 15 & 18)

Particle Filters and Supervised Learning

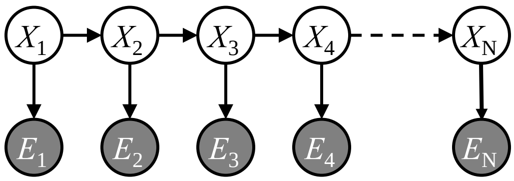
Ghostbusters

Pac-Man does not know true position of the ghost



Must infer probability distribution over true ghost position

Hidden Markov Model (HMM)



Hidden State at time t = 1, 2,…,N

Emissions (measurements) at time t = 1, 2,…,N

HMM is defined by 2 conditional probabilities:

$P(X_t \mid X_{t-1})$ Transition model $= P(X' \mid X)$

$P(E_t \mid X_t)$ Emission model $= P(E \mid X)$

(aka measurement/observation model)

plus initial state distribution $P(X_1)$

Ghostbusters HMM

- $P(X_1)$ = uniform

$P(X_1)$

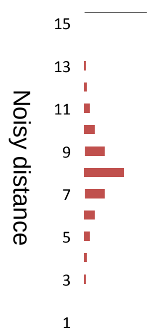
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

- P(X'|X) = ghost usually moves *clockwise*, but sometimes moves in a random direction or stays in place

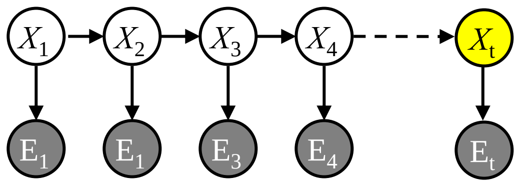
P(X'|X=<1,2>)

1/6	1/6	1/2
0	1/6	0
0	0	0

- P(E|X) = compute Manhattan distance to ghost from Pac-Man and emit a *noisy distance* given this true distance (see example for true distance = 8)



HMM Inference Problem



Where is the ghost now?
Compute posterior probability over X_t

- Given evidence $\boldsymbol{E_1, \ldots, E_t = E_{1:t} = e_{1:t}}$
- Inference problem (aka *Filtering*): Find posterior $P(X_t|e_{1:t})$ for current t

The "Forward" Algorithm for Filtering

$$P(X_t \mid e_1,\ldots,e_t) = \alpha\, P(e_t \mid X_t, e_1,\ldots,e_{t-1})\, P(X_t \mid e_1,\ldots,e_{t-1})$$ Bayes

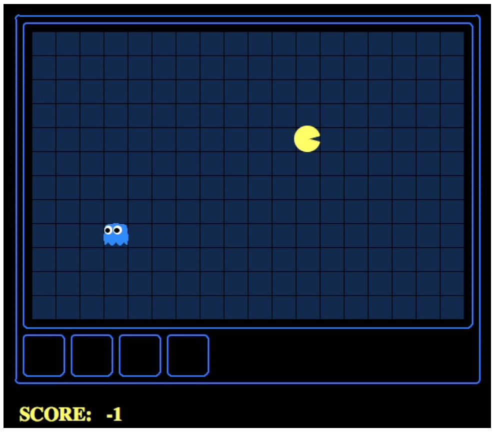
$$= \alpha\, P(e_t \mid X_t) \sum_{X_{t-1}} P(X_t \mid X_{t-1}) P(X_{t-1} \mid e_1,\ldots,e_{t-1})$$ Markov + Marginalize

Normalization constant
Emission model
Transition model
Previous estimate

New estimate

At each time step *t*, compute and maintain a table of *P* values over all possible values of *X*

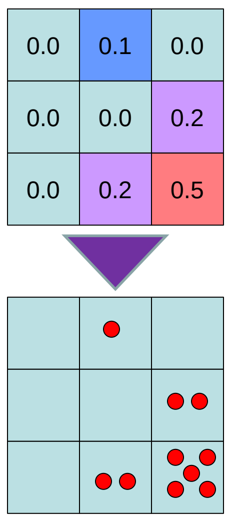
Filtering using the Forward Algorithm



$P(X_t \mid e_1,\ldots,e_t)$ is an array of $12 \times 18 = 216$ values
(one for each location)

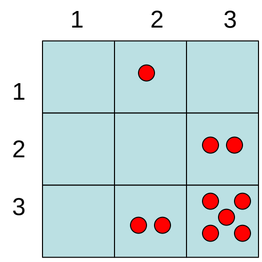
Particle Filtering

- Sometimes |X| is too big for exact inference
  - |X| may be too big to even store $P(X_t \mid e_{1:t})$
    E.g. when X is continuous
- Solution: Approximate inference
  - Track a set of *samples* of X
  - Samples are called ***particles***
  - Number of samples for X=x is proportional to probability of x



Representation: Particles

- Our representation of P(X) is now a list of N particles (samples)
  - ***Generally, N << |X|***
- P(x) approximated by number of particles with value x
  - Note: Many x will have P(x) = 0!
  - More particles, more accuracy
- Initially, all particles have a "weight" of 1



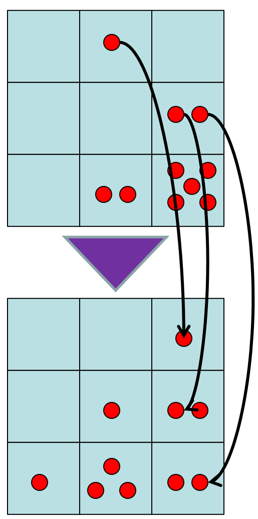
Particles:
(1,2) (3,3)
(2,3) (3,3)
(2,3) (3,3)
(3,2) (3,3)
(3,2) (3,3)

Particle Filtering
Step 1: Elapse Time

- Each particle x is moved by sampling its next position using the transition model

  $x' = \text{sample}(P(X'|x))$

  - Samples' frequencies reflect the transition probs
  - In example, most samples move clockwise, but some move in another direction or stay in place
- This step captures passage of time



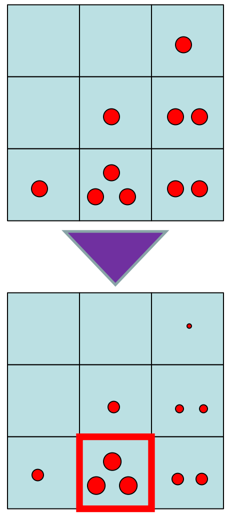
Particle Filtering
Step 2: Observe

Weight particles according to evidence

- ***Assign weights*** w ***to samples*** based on the new observed evidence e

$$w(x) = P(e|x)$$

- In example, true ghost position is shown in red outline; samples closer to ghost get higher weight (bigger size of circles) based on noisy distance emission model

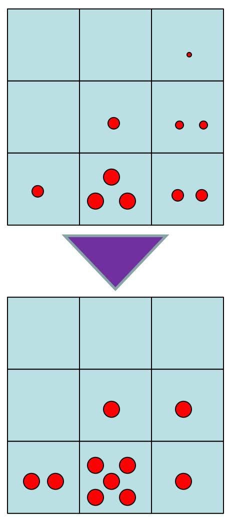


Particle Filtering
Step 3: Resample

- N times, we choose from our weighted sample distribution (i.e. randomly select with replacement)
  - Each sample selected with probability proportional to its weight
- Now the update is complete for this time step, continue with the next one

Old Particles:
(1,3) w=0.1
(3,2) w=0.9
(3,2) w=0.9
(3,1) w=0.4
(2,3) w=0.3
(2,2) w=0.4
(3,3) w=0.4
(3,3) w=0.4
(3,2) w=0.9
(2,3) w=0.3

New Particles:
(3,2) w=1
(3,2) w=1
(3,2) w=1
(2,3) w=1
(2,2) w=1
(3,2) w=1
(3,1) w=1
(3,3) w=1
(3,2) w=1
(3,1) w=1



Particle Filtering Summary

- Represent current belief P(X | evidence to date) as set of *N* samples (actual values x)
- For each new observation e:
  1. *Sample transition*, once for each current particle x

     $$x' = \text{sample}(P(X'|x))$$

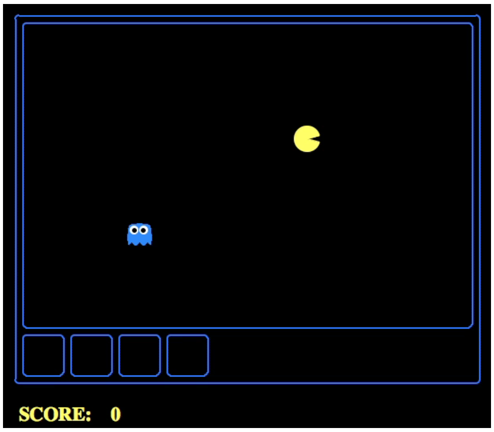
  2. For each new sample x', *compute importance weights* for the new evidence e:

     $$w(x') = P(e|x')$$

  3. Finally, *resample* the importance weights to create N new particles

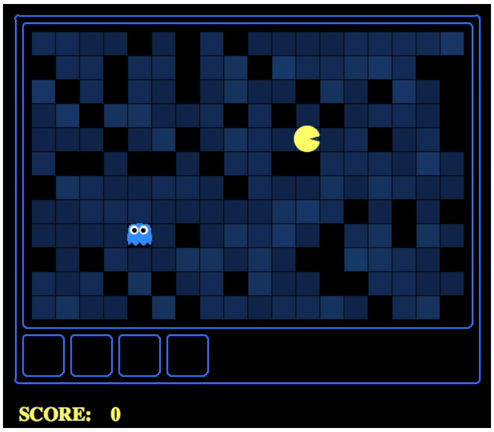
Example 1

Particle filter, uniform initial beliefs, 25 particles



Example 2

Particle filter, uniform initial beliefs, 300 particles



Yesterday's headline:

The New York Times

November 23, 2012

Scientists See Promise in Deep-Learning Programs

By JOHN MARKOFF

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

The advances have led to widespread enthusiasm among researchers who design software to perform human activities li They offer the promise of machines that converse with humans and perform tasks like driving cars and working in factori 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, w Communications' speech recognition service, and in Google's Street View, which uses machine vision to identify specific ad

But what is new in recent months is the growing speed and accuracy of deep-learning programs, often called artificial neu for their resemblance to the neural connections in the brain.

Hao Zhang/The New York Times

A voice recognition program translated a speech given by Richard F. Rashid, Microsoft's top scientist, into Mandarin Chinese.

Machine learning everywhere!

The Seattle Times | Local News
Winner of a 2012 Pulitzer Prize

Home | News | Business & Tech | Sports | Entertainment | Living | H

IN THE NEWS: Holiday events | Bellevue light rail | Apple Cup aftermath | Charter schools | Fun

Originally published June 27, 2012 at 7:30 PM | Page modified June 27, 2012 at 8:31 PM

UW recruits superstars of computer-science world

The University of Washington has landed four new faculty members considered among the brightest in the world of computer science.

PRWeb
Online Visibility from Vocus

HOME | NEWS CENTER | BLOG

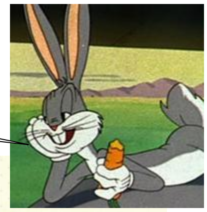
Front Page | Arts | Business | Education | Environment | Government | Indu

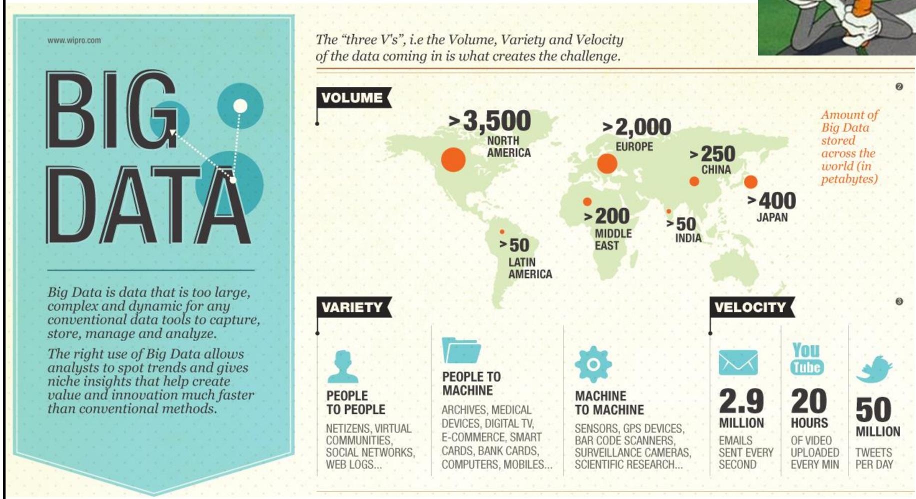
Sunday, November 25, 2012

Beware of the New "Machine Learning" IRS Tax Audit Advises Former IRS Manager

"The public should be aware that the IRS has begun using a new audit method, the" Machine Learning Tax Audit", states Michael Sullivan of Fresh Start Tax and a former IRS Agent. "This new system will allow the IRS to conduct more audits on a yearly basis, which will create more IRS tax problems and tax debt for individuals."

Why all the hubbub, bub?





Varieties of Machine Learning

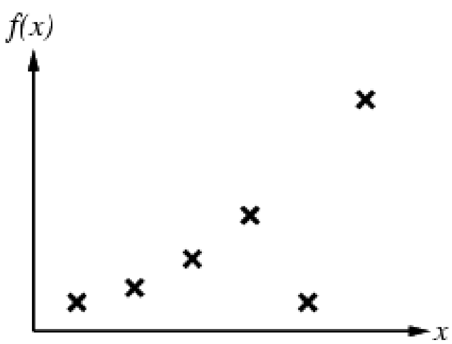
- Supervised learning: correct answers for each input is provided, goal is to *generalize* to new data
  - E.g., decision trees, neural networks
- Unsupervised learning: correct answers not given, must discover patterns in input data
  - E.g., clustering, principal component analysis
- Reinforcement learning: occasional rewards (or punishments) given to guide behavior
  - We've covered this already! (Q-learning, MDPs)

Supervised Learning

- Classification
  - Decision trees
  - K-nearest neighbor
  - Linear Classifiers
  - Support Vector Machines (SVMs)
  - Cross validation
- Regression
  - Linear regression and Neural networks
    - Backpropagation learning algorithm

Supervised learning

- Goal: Construct a function *h* from training data to approximate the hidden function *f* that is generating the data
  - *h* is consistent if it agrees with *f* on all training examples

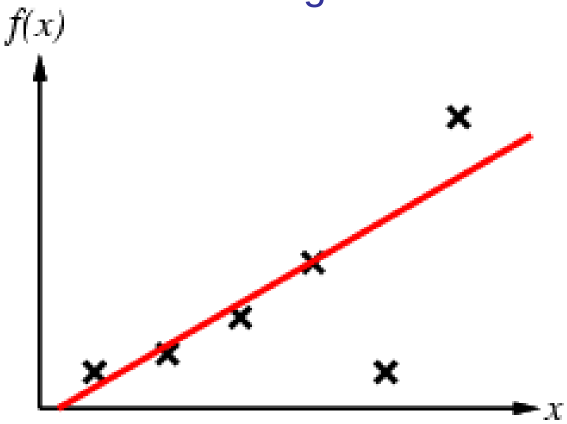


E.g., curve fitting (aka regression)

Given: Data points $(x,f(x))$
(training examples)

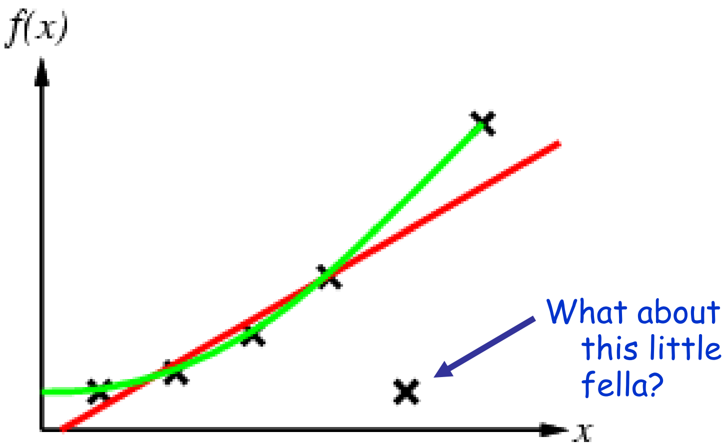
Supervised learning example

h = Straight line?



Supervised learning example

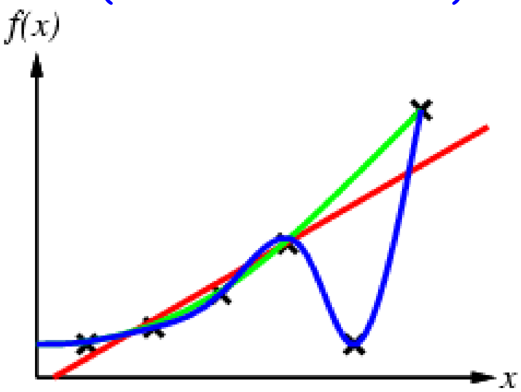
What about a quadratic function?



Supervised learning example

Finally, a function that satisfies all!

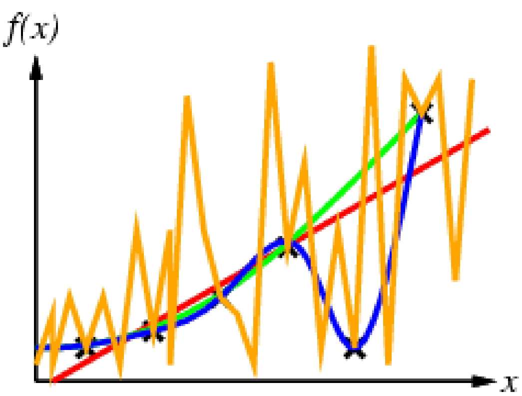
(consistent function)



23

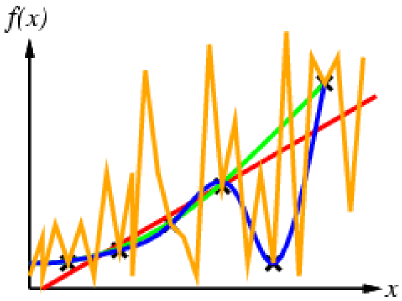
Supervised learning example

But so does this one…



24

Ockham's Razor Principle



Prefer the simplest hypothesis consistent with data

- Related to KISS principle ("keep it simple stupid")
- *Smooth* blue function preferable over wiggly yellow one
- If noise known to exist in data, even linear might be better (the lowest x might be due to noise)

25

Next Time

- Learning Decision Trees from data
- Nearest Neighbor classification
- To Do:
  - Project 4
  - Read Chapter 18

26