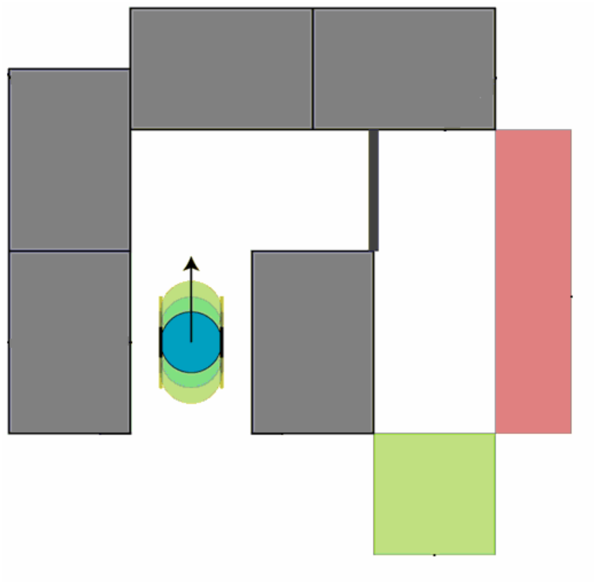
# Self-improvement for dummies
## (Machine Learning)

COS 116
4/23/2007
Instructor: Umar Syed

# Recall your final Scribbler lab

- Task: Program Scribbler to navigate a maze.
  - Avoid walls, avoid “lava”, head towards the goal.



- Seemed simple. So why was this so challenging?

# *Teach* Scribbler to navigate a maze

Start with a simple program:

1. Run the maze.
2. Label this trial GOOD or BAD, depending on whether goal was reached.
3. Submit data from the trial to a “learning algorithm”, which uses it to devise a better program.
4. Repeat as needed.

- Is this how you learned to drive a car?


SPAM®
Ingredients: Pork with Ham, Salt, Water, Sugar, Sodium Nitrite.
GET
SPAM
STUFF
SEE BACK FOR DETAILS
NET WT. 12 OZ. (340g)
U.S. INSPECTED AND PASSED BY DEPARTMENT OF AGRICULTURE
Hormel Foods
Serving
Sugges

# Spam filtering


- How would you define Spam to a computer?
- Descriptive approach:
  - “Any email in ALL CAPS, unless it’s from my kid brother, or that contains the word ‘mortgage’, unless it’s from my real estate agent, …”
  - Difficult to come up with an good description!
- Learning approach:
  - “Train” the computer with labeled examples of spam and non-spam (a.k.a. ham) email.
  - Easy to find examples of spam – you probably get hundreds a day!

# Today’s lecture: Machine Learning

- Machine learning = “Programming by example.”
- <u>Show</u> the computer what to do, without explaining <u>how</u> to do it.
- The computer programs itself!

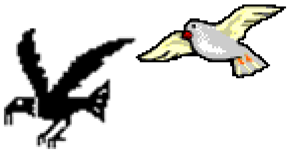
# Machine Learning (ML):

- A subfield within Artificial Intelligence (AI).
- Algorithms that <u>improve their performance</u> with experience/data.
- Closely related to *Data Mining.*
  - Data mining = Finding patterns and relationships in data.

# ML is **not** concerned with modeling human intelligence.

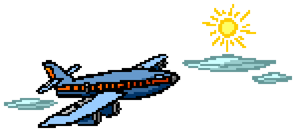
- Imitating nature may not be the best strategy anyway:

Birds



vs

Airplanes



Cheetahs

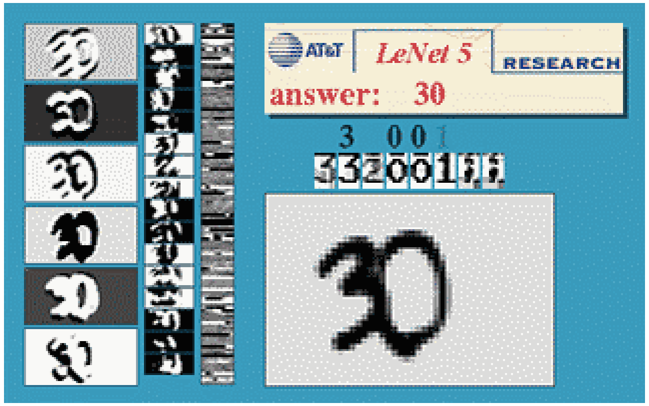
vs

Race cars

# Examples of Machine Learning

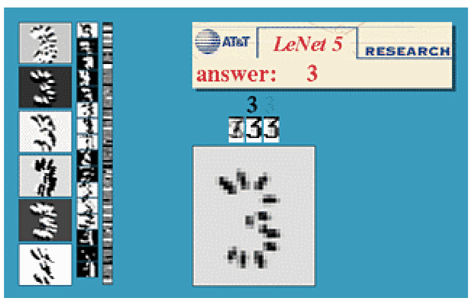
# Handwriting recognition
[LeCun et al, AT&T, 1998]

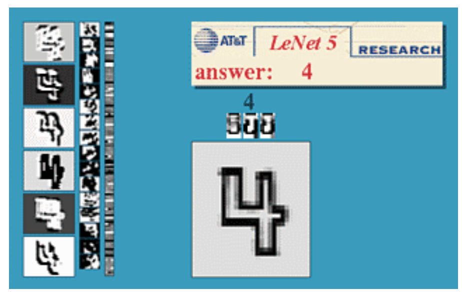
- The LeNet-5 system
  - Trained on a database of 60,000 handwritten digits.
  - Reads about 10% of all the checks cashed in the USA.

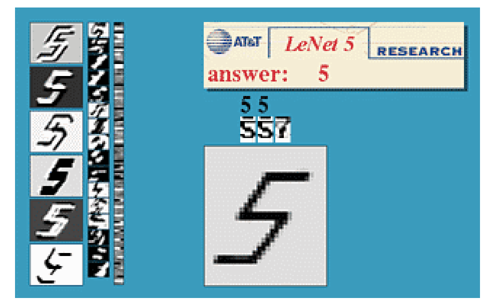


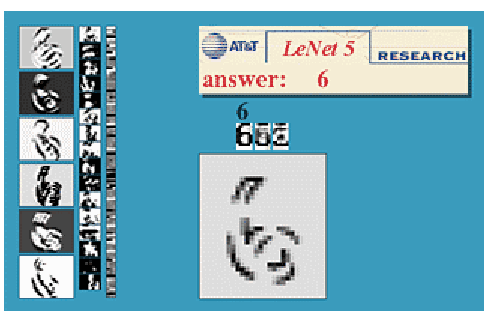
# Handwriting recognition: LeNet-5

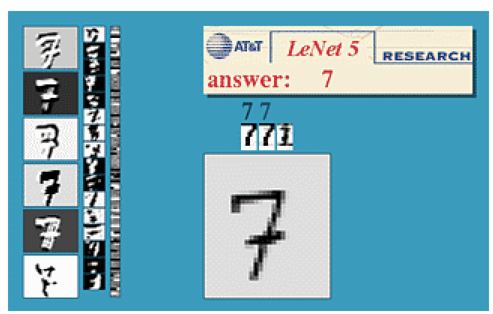
- Can recognize weird styles:

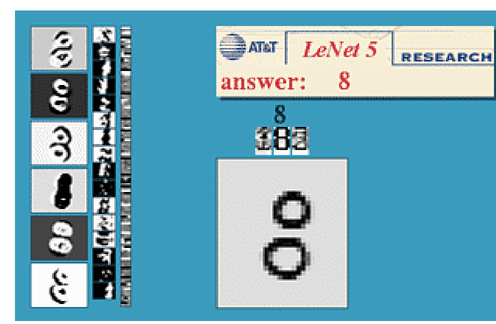


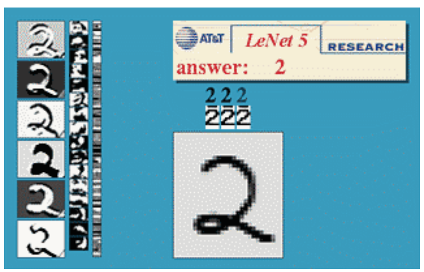



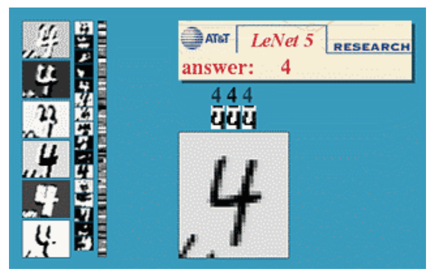



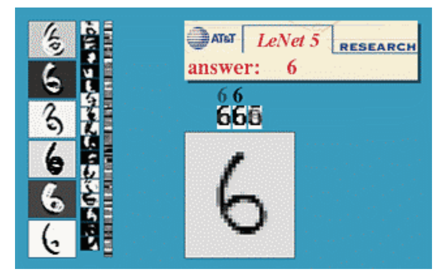



# Handwriting recognition: LeNet-5

- Can handle stray marks and deformations:




- Mistakes are usually ambiguous anyway:

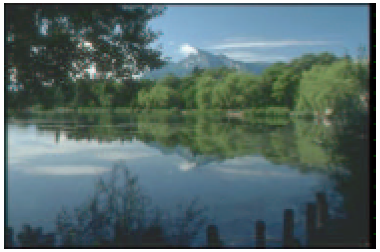
# Image labeling

- Problem: Searching an image database is hard.
  - How does the computer know what a "sunset" looks like?
- Solution: Add captions to the images.
  - Google uses the nearby text on the web page.
  - Remember the Image Labeling Game from Lab 1?

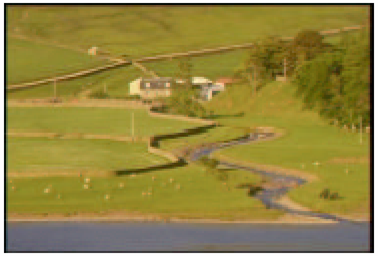
# Image labeling [Blei et al, 2003]

Princeton prof!

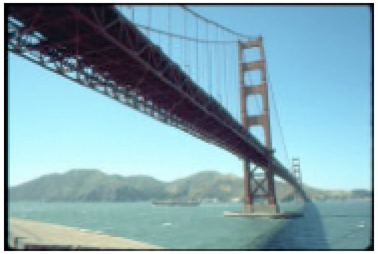
- Another solution: Learn captions from examples.
  - System trained on a Corel database of 6,000 images with captions.
  - Applied to images without captions.



SKY WATER TREE
MOUNTAIN PEOPLE

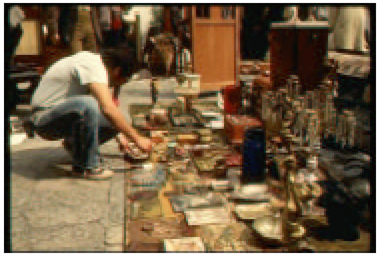


SCOTLAND WATER
FLOWER HILLS TREE



SKY WATER BUILDING
PEOPLE WATER

FISH WATER OCEAN
TREE CORAL

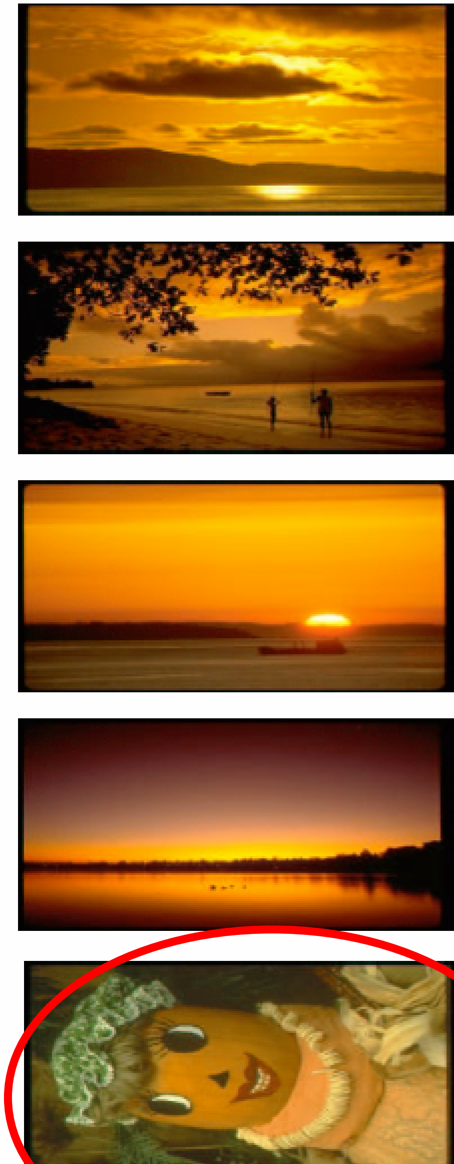


PEOPLE MARKET PATTERN
TEXTILE DISPLAY

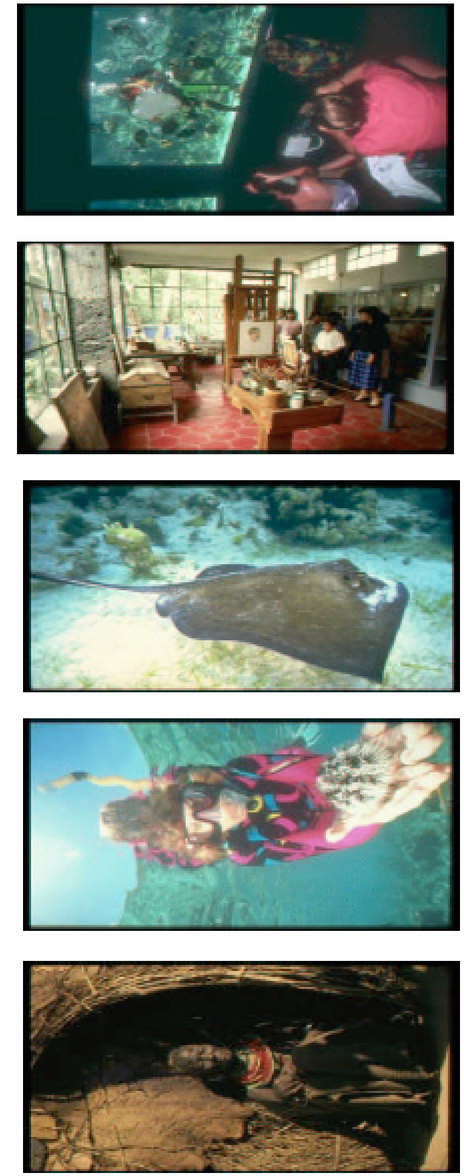
BIRDS NEST TREE
BRANCH LEAVES

Candy

Sunset

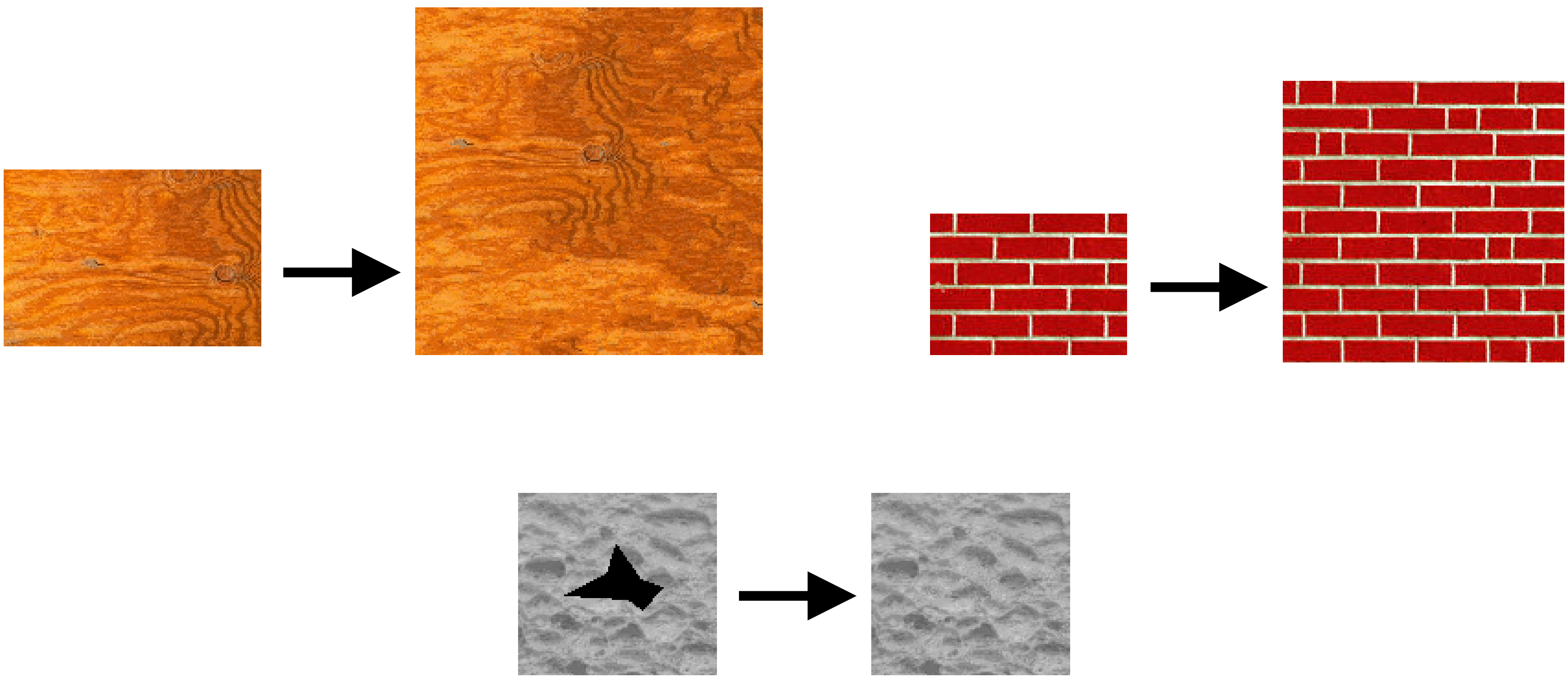


People & Fish



# Texture synthesis [Efros and Leung 2001]

- Idea: Use examples of a texture to generate similar textures.



# Text synthesis

- Idea: Use example text to generate similar text.
  - Input: 2007 State of the Union Address.
  - Output: “This war is more competitive by strengthening math and science skills. The lives of our nation was attacked, I ask you to make the same standards, and a prompt up-or-down vote on the work we've done and reduce gasoline usage in the NBA.”
- Even with <u>no</u> knowledge of grammar or semantics, output can look (sort of) sensible.

# SAT Analogies

- Remember these?
  - Bird : Feathers :: Fish : ____
- Could a computer possibly solve these? Seems to require understanding.
- Douglas Hofstadter, cognitive scientist, author of *Gödel, Escher, Bach*, on cognition:
  - "[A]nalogy is <u>everything</u>, or very nearly so, in my view."

# SAT Analogies

- Bird : Feathers :: Fish : ____

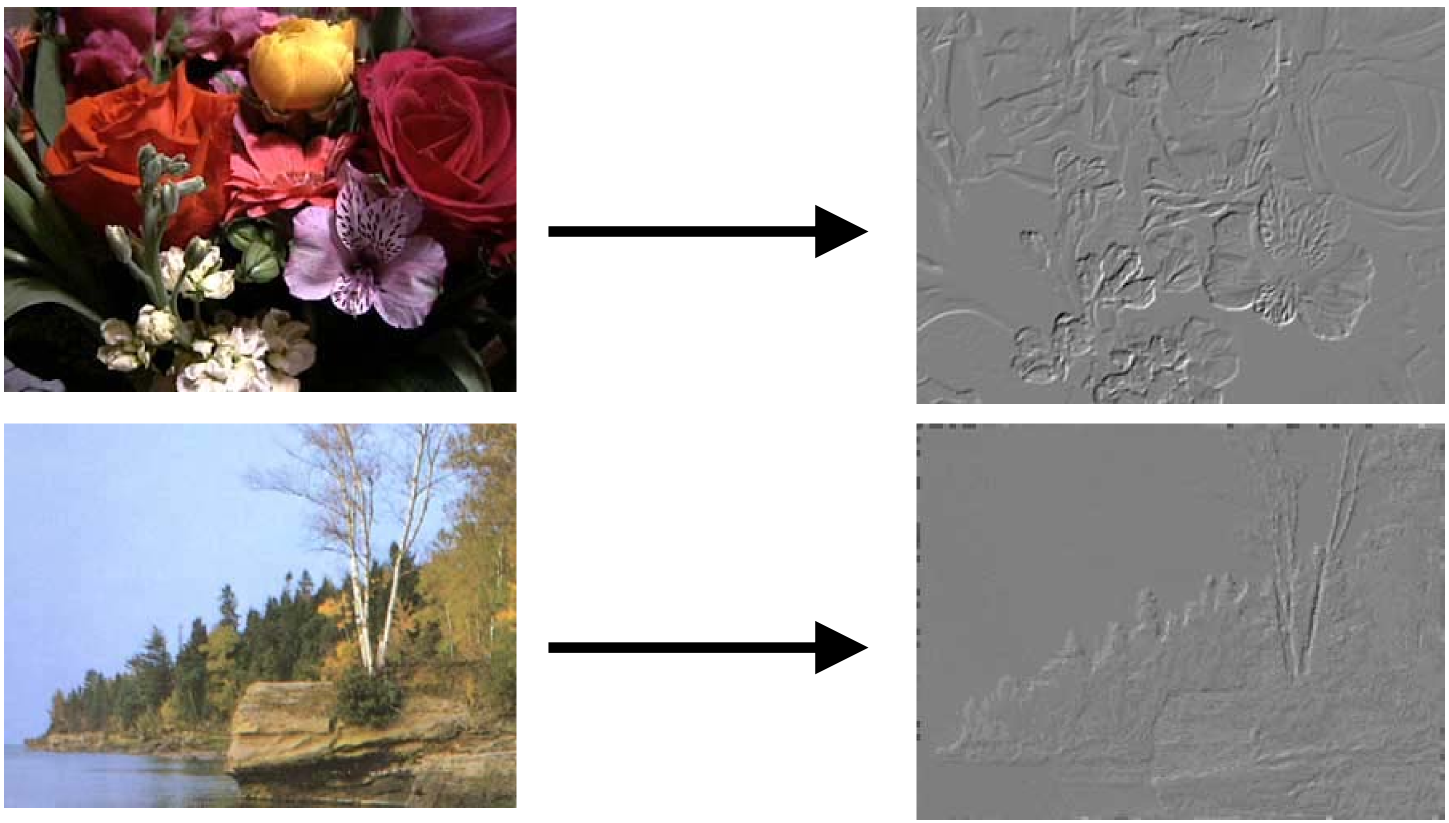
- Idea: Search the web to learn relationships between words. [Turney 2004]
  - Example: Is the answer above “water” or “scales”?
    - Most common phrases on the web: “bird *has* feathers”, “bird *in* air”, “fish *has* scales”, “fish *in* water”.
    - Conclusion: Right answer is “scales”.

# SAT Analogies [Turney 2004]

- On a set of 374 multiple-choice SAT analogies, this approach got 56% correct.
- High-school seniors on the same set:
  - 57% (!)
- Q: So are computers intelligent now?

# Image Analogies [Hertzmann, et al 2001]

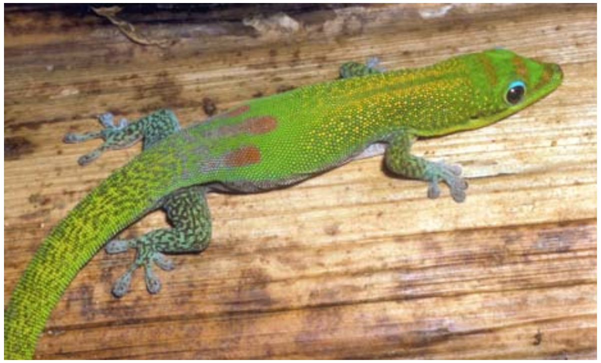
# Image Analogies [Hertzmann, et al 2001]



# Species Habitat Modeling

[Dudik, Schapire, et al] ← At Princeton!

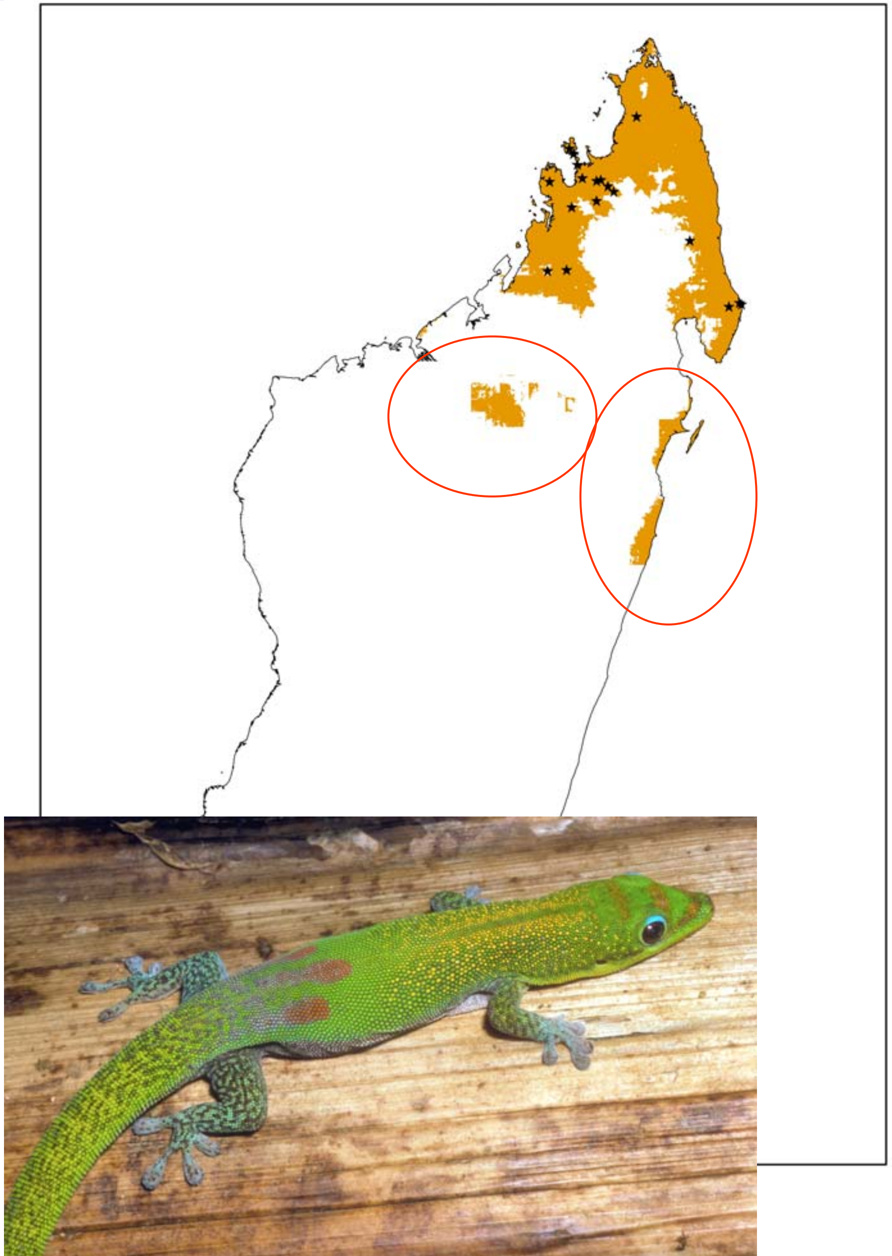
- Problem: Determine where these live:



- Solution: Use the features of their known habitats to infer new habitats.
    - Example: If we always observe them in warm, rainy places, then infer that that's where they tend to live.

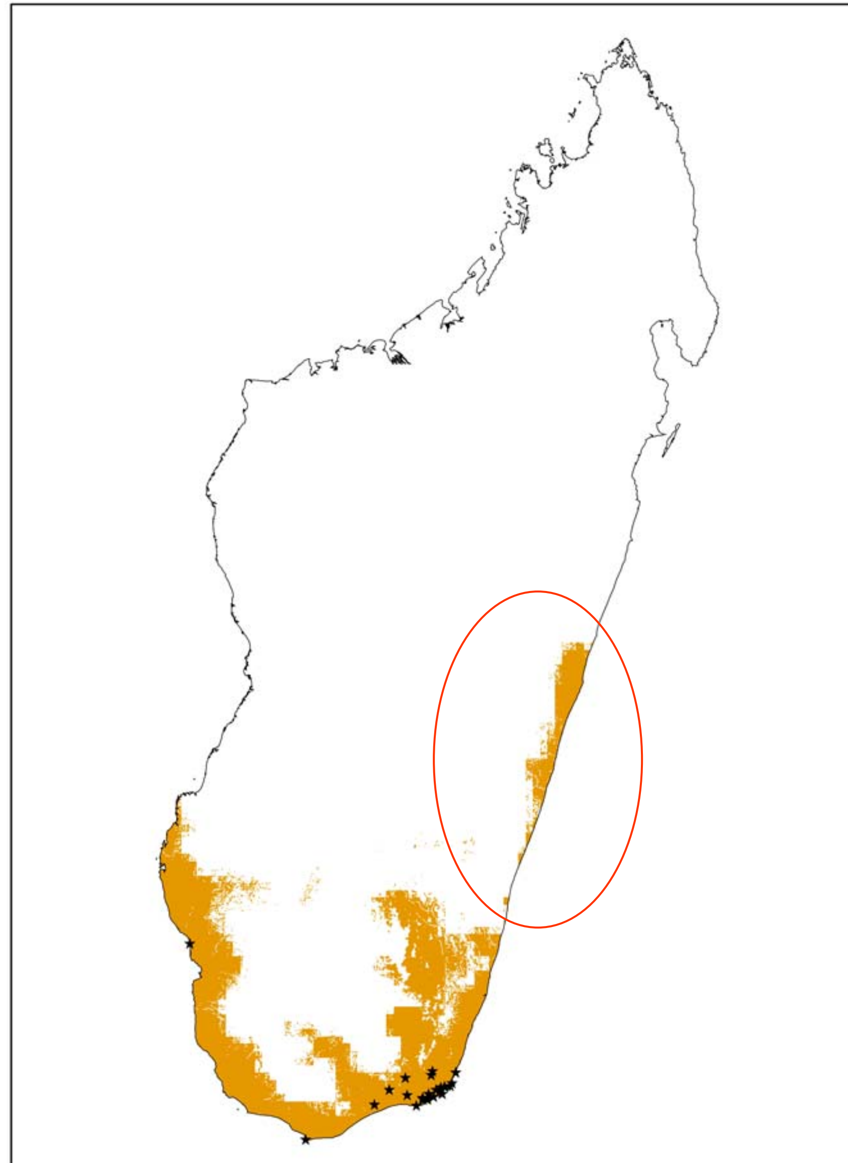
# Phelsuma laticauda

Maxent model, lpt



# Phelsuma modesta

Maxent model, lpt



# Helicopter flight [Abbeel et al 2005]

- Idea: Algorithm learns to pilot a helicopter by observing a human pilot.
- Results: Even <u>better</u> than the human pilot.

# Twenty Questions

http://www.20q.net

# Machine Learning Algorithms -- Details

# Spam Filtering


- Given: A spam corpus and ham corpus.
- Goal: Determine whether a new email is spam or ham.

- Step 1: Assign a "spam score" to each *word:*
  - $F_{spam}$(*word*) = Fraction of emails in spam corpus that contain *word*.
  - $F_{ham}$(*word*) = Fraction of emails in ham corpus that contain *word*.

$$SpamScore(word) = \frac{F_{spam}(word)}{F_{ham}(word)}$$

  - Observe:
    - SpamScore(*word*) > 1 if *word* is more prevalent in spam.
    - SpamScore(*word*) < 1 if *word* is more prevalent in ham.

# Spam Filtering


- Step 2: Assign a “spam score” to the *email*:
  - SpamScore(*email*) = SpamScore($word_1$) x … x SpamScore($word_n$), where $word_i$ is the $i^{th}$ word in *email*.
  - Observe:
    - SpamScore(*email*) >> 1 if *email* contains many spammy words.
    - SpamScore(*email*) << 1 if *email* contains many hammy words.
- Step 3: Declare *email* to be spam if SpamScore(*email*) is high enough.

# Spam Filtering


- Advantages of this type of spam filter:
  - Though simple, catches 90+% of spam!
  - No explicit definition of spam required.
  - Customized for your email.
  - Adaptive – as spam changes, so does the filter.

# Text synthesis

- Idea: Use example text to generate similar text.
  - Input: 2007 State of the Union Address.
  - Output: “This war is more competitive by strengthening math and science skills. The lives of our nation was attacked, I ask you to make the same standards, and a prompt up-or-down vote on the work we've done and reduce gasoline usage in the NBA.”

# Text synthesis

- How it works: Output one word at a time.
  1. Let (v, *w)* be the last two words outputted.
  2. Find all occurrences of (v, *w)* in the input text.
  3. Of the words following the occurrences of (v, w), output one at random.
  4. Repeat.
- Variants: Last *k* words instead of last two words.

# Conclusion

- Machine learning = Pretty great.

- Questions?

- Next week's lab: You'll experiment with spam filtering and text generation.