# Introduction to Adaptive Boosting

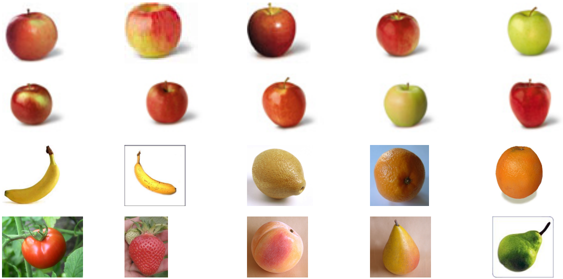
Hsuan-Tien Lin

National Taiwan University

Machine Learning, Fall 2009

# Apple Recognition Problem

- Is this a picture of an apple?
- We want to teach a class of 6 year olds.
- Gather photos from NY Apple Asso. and Google Image.

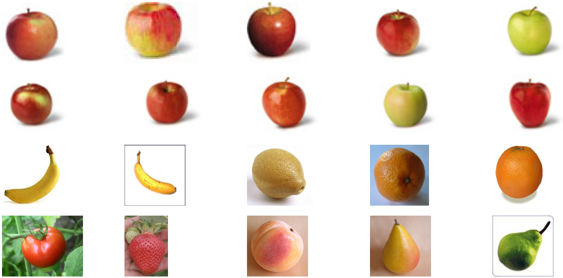


# Our Fruit Class Begins

Teacher: How would you describe an apple? Michael?

Michael: I think apples are circular.

(Class): Apples are circular.

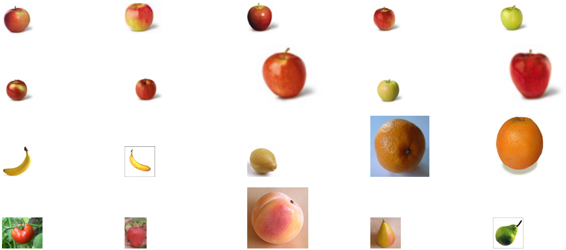


# Our Fruit Class Continues

**Teacher:** Being circular is a good feature for the apples. However, if you only say circular, you could make several mistakes. What else can we say for an apple? Tina?

**Tina:** It looks like apples are red.

**(Class):** Apples are somewhat circular and somewhat red.

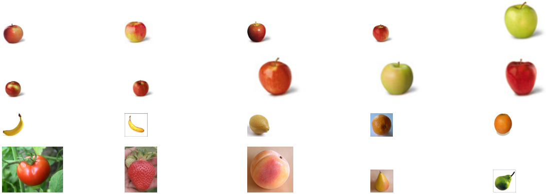


# Our Fruit Class Continues

**Teacher:** Yes. Many apples are red. However, you could still make mistakes based on circular and red. Do you have any other suggestions, Joey?

**Joey:** Apples could also be green.

**(Class):** Apples are somewhat circular and somewhat red and possibly green.

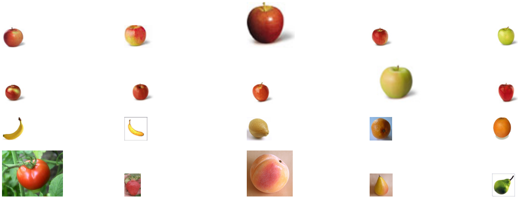


# Our Fruit Class Continues

**Teacher:** Yes. It seems that apples might be circular, red, green. But you may confuse them with tomatoes or peaches, right? Any more suggestions, Jessica?

**Jessica:** Apples have stems at the top.

**(Class):** Apples are somewhat circular, somewhat red, possibly green, and may have stems at the top.



# Put Intuition to Practice

Intuition

- Combine simple rules to approximate complex function.
- Emphasize incorrect data to focus on valuable information.

AdaBoost Algorithm (Freund and Schapire 1997)

- Input: training examples $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule $h_t$ from emphasized training examples.
  - Get the confidence $\alpha_t$ of such rule
  - Emphasize the training examples that do not agree with $h_t$.
- Output: combined function $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right)$

# Some More Details

AdaBoost Algorithm

- Input: training examples $Z = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule $h_t$ from emphasized training examples.
    - How? Choose a $h_t \in \mathcal{H}$ with minimum emphasized error.
  - Get the confidence $\alpha_t$ of such rule
    - How? An $h_t$ with lower error should get higher $\alpha_t$.
  - Emphasize the training examples that do not agree with $h_t$.
    - How? Maintain an emphasis value $u_n$ per example.
- Output: combined function $H(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right)$
- Let's see some demos.

## The Final Version

- Input: $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$. **Set** $u_n = \frac{1}{N}$ **for all** $n$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule $h_t$ such that $h_t$ solves
    $$\min_h \sum_{n=1}^N u_n \cdot [y_n \neq h(\mathbf{x}_n)].$$
  - Compute the error $\epsilon_t = \sum_{n=1}^N \frac{u_n}{\sum_{m=1}^N u_m} \cdot I[y_n \neq h(\mathbf{x}_n)]$ and the confidence
    $$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$$
  - Emphasize the training examples that do not agree with $h_t$:
    $$u_n = u_n \cdot \exp\Big(-\alpha_t y_n h_t(\mathbf{x}_n)\Big).$$
- Output: combined function $H(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x})\right)$