Introduction to Boosting and Joint Boosting

Hsuan-Tien Lin

Learning Systems Group, Caltech

2005/04/26, Presentation in EE150

Outline

1. Introduction to Boosting
   - Intuition of Boosting
   - Adaptive Boosting (AdaBoost)
2. Joint Boosting
   - Independent Boosting
   - Joint Boosting

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.

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.

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.


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.



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.

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.

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.



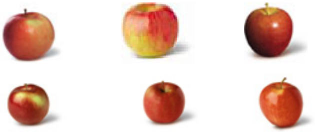



Apple Recognition Problem

- 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.

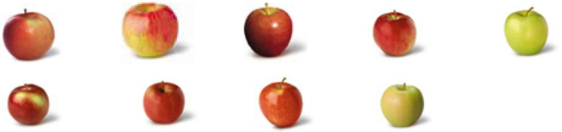
Apple Recognition Problem

- 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.



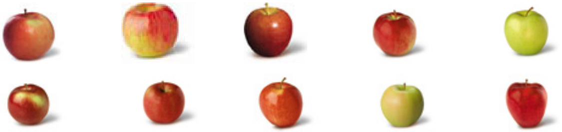
Apple Recognition Problem

- 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.



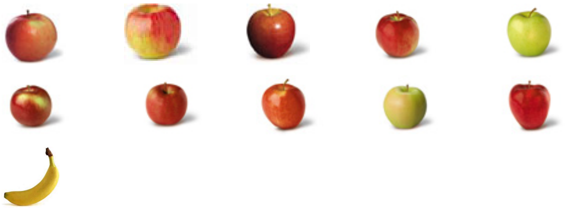
Apple Recognition Problem

- 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.



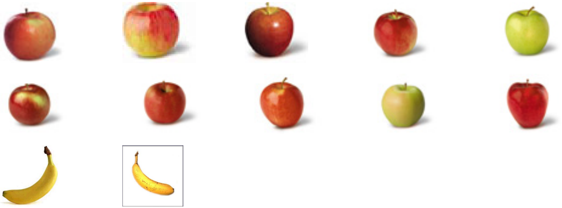
Apple Recognition Problem

- 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.



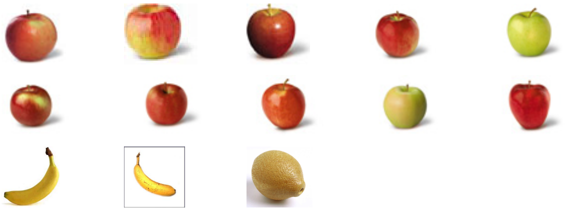
Apple Recognition Problem

- 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.



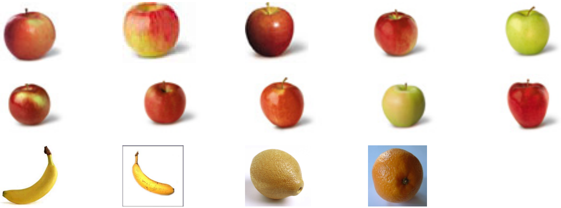
Apple Recognition Problem

- 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.



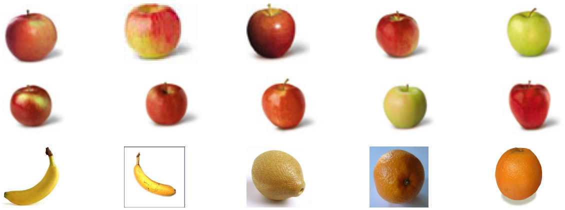
Apple Recognition Problem

- 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.



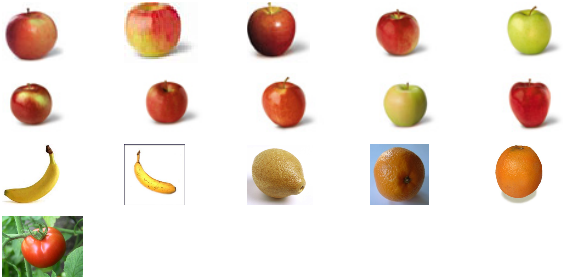
Apple Recognition Problem

- 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.



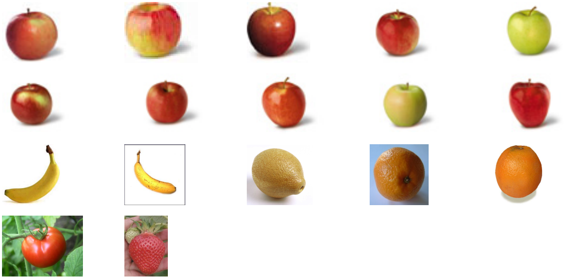
Apple Recognition Problem

- 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.



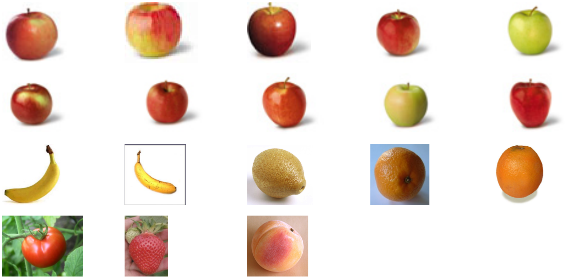
Apple Recognition Problem

- 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.



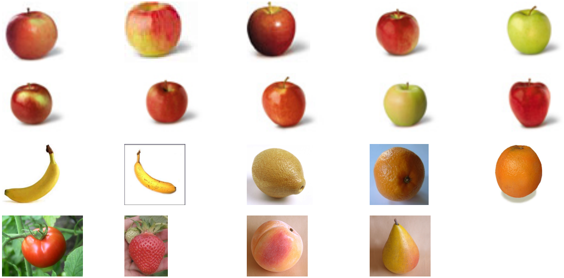
Apple Recognition Problem

- 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.



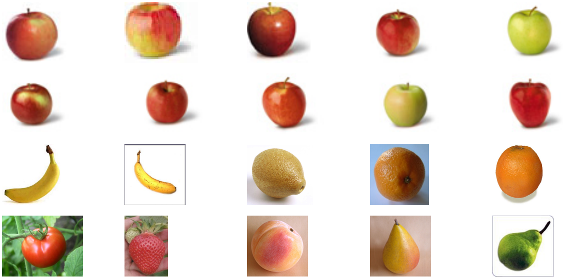
Apple Recognition Problem

- 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.



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.

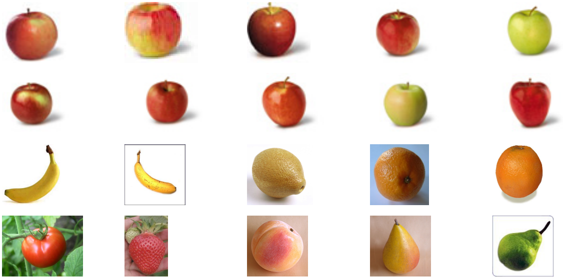


Our Fruit Class Begins

Teacher: How would you describe an apple? Michael?

Michael: I think apples are circular.

(Class): Apples are circular.

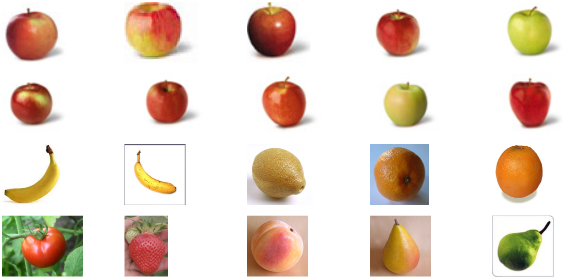


Our Fruit Class Begins

Teacher: How would you describe an apple? Michael?

Michael: I think apples are circular.

(Class): Apples are circular.

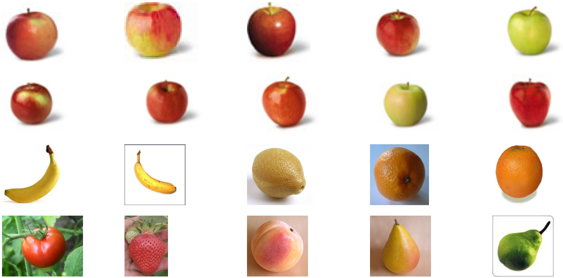


Our Fruit Class Begins

Teacher: How would you describe an apple? Michael?

Michael: I think apples are circular.

(Class): Apples are circular.

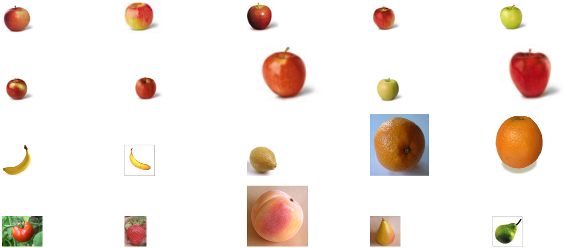


Our Fruit Class Continues

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.

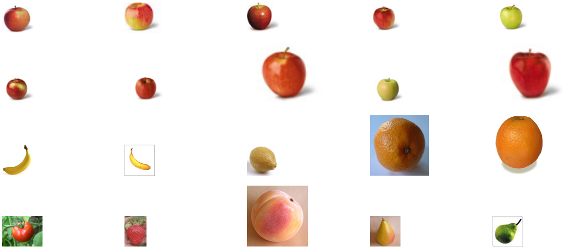


Our Fruit Class Continues

Teacher: 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.

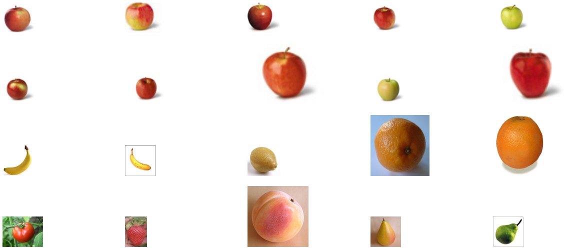


Our Fruit Class Continues

Teacher: 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.

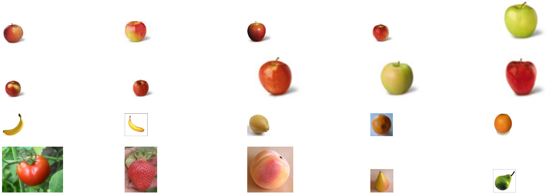


Our Fruit Class Continues

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.

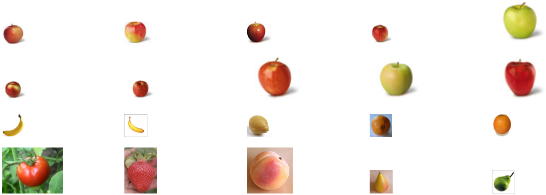


Our Fruit Class Continues

Teacher: Yes. 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.

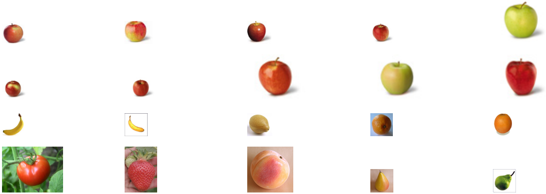


Our Fruit Class Continues

Teacher: Yes. 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.

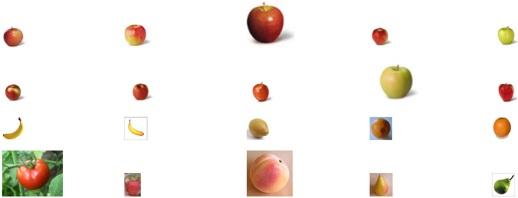


Our Fruit Class Continues

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.

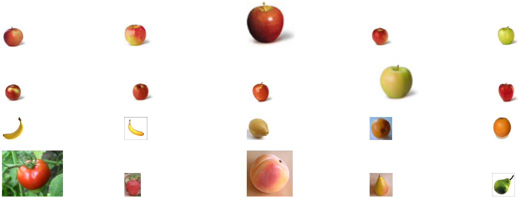


Our Fruit Class Continues

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.

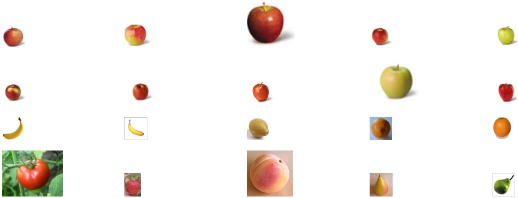


Our Fruit Class Continues

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 data $Z = (x_i, y_i)_{i=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t from emphasized training data.
  - Get the confidence w_t of such rule
  - Emphasize the training data that do not agree with h_t.
- Output: combined function $H(x) = \sum_{t=1}^{T} w_t h_t(x)$ with normalized w.

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 data $Z = (x_i, y_i)_{i=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t from emphasized training data.
  - Get the confidence w_t of such rule
  - Emphasize the training data that do not agree with h_t.
- Output: combined function $H(x) = \sum_{t=1}^{T} w_t h_t(x)$ with normalized w.

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 data $Z = (x_i, y_i)_{i=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t from emphasized training data.
  - Get the confidence w_t of such rule
  - Emphasize the training data that do not agree with h_t.
- Output: combined function $H(x) = \sum_{t=1}^{T} w_t h_t(x)$ with normalized w.

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 data $Z = (x_i, y_i)_{i=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t from emphasized training data.
  - Get the confidence w_t of such rule
  - Emphasize the training data that do not agree with h_t.
- Output: combined function $H(x) = \sum_{t=1}^{T} w_t h_t(x)$ with normalized w.

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 data $Z = (x_i, y_i)_{i=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t from emphasized training data.
  - Get the confidence w_t of such rule
  - Emphasize the training data that do not agree with h_t.
- Output: combined function $H(x) = \sum_{t=1}^{T} w_t h_t(x)$ with normalized w.

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 data $Z = (x_i, y_i)_{i=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t from emphasized training data.
  - Get the confidence w_t of such rule
  - Emphasize the training data that do not agree with h_t.
- Output: combined function $H(x) = \sum_{t=1}^{T} w_t h_t(x)$ with normalized w.

Some More Details

AdaBoost Algorithm

- Input: training data $Z = (x_i, y_i)_{i=1}^N$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t from emphasized training data.
    - How? Choose a $h_t \in \mathcal{H}$ with minimum emphasized error.
    - For example, $\mathcal{H}$ could be a set of decision stumps $h_{\theta,d,s}(x) = s \cdot I[(x)_d > \theta]$.
  - Get the confidence w_t of such rule
    - How? An h_t with lower error should get higher w_t.
  - Emphasize the training data that do not agree with h_t.
- Output: combined function $H(x) = \sum_{t=1}^T w_t h_t(x)$ with normalized w.
- Let's see some demos.

Some More Details

AdaBoost Algorithm

- Input: training data $Z = (x_i, y_i)_{i=1}^N$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t from emphasized training data.
    - How? Choose a $h_t \in \mathcal{H}$ with minimum emphasized error.
    - For example, $\mathcal{H}$ could be a set of decision stumps $h_{\theta,d,s}(x) = s \cdot I[(x)_d > \theta]$.
  - Get the confidence w_t of such rule
    - How? An h_t with lower error should get higher w_t.
  - Emphasize the training data that do not agree with h_t.
- Output: combined function $H(x) = \sum_{t=1}^T w_t h_t(x)$ with normalized w.
- Let's see some demos.

Some More Details

AdaBoost Algorithm

- Input: training data $Z = (x_i, y_i)_{i=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t from emphasized training data.
    - How? Choose a $h_t \in \mathcal{H}$ with minimum emphasized error.
    - For example, $\mathcal{H}$ could be a set of decision stumps $h_{\theta,d,s}(x) = s \cdot I[(x)_d > \theta]$.
  - Get the confidence w_t of such rule
    - How? An h_t with lower error should get higher w_t.
  - Emphasize the training data that do not agree with h_t.
- Output: combined function $H(x) = \sum_{t=1}^{T} w_t h_t(x)$ with normalized w.
- Let's see some demos.

Some More Details

AdaBoost Algorithm

- Input: training data $Z = (x_i, y_i)_{i=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t from emphasized training data.
    - How? Choose a $h_t \in \mathcal{H}$ with minimum emphasized error.
    - For example, $\mathcal{H}$ could be a set of decision stumps $h_{\theta,d,s}(x) = s \cdot I[(x)_d > \theta]$.
  - Get the confidence w_t of such rule
    - How? An h_t with lower error should get higher w_t.
  - Emphasize the training data that do not agree with h_t.
- Output: combined function $H(x) = \sum_{t=1}^{T} w_t h_t(x)$ with normalized w.
- Let's see some demos.

Some More Details

AdaBoost Algorithm

- Input: training data $Z = (x_i, y_i)_{i=1}^N$.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t from emphasized training data.
    - How? Choose a $h_t \in \mathcal{H}$ with minimum emphasized error.
    - For example, $\mathcal{H}$ could be a set of decision stumps $h_{\theta,d,s}(x) = s \cdot I[(x)_d > \theta]$.
  - Get the confidence w_t of such rule
    - How? An h_t with lower error should get higher w_t.
  - Emphasize the training data that do not agree with h_t.
- Output: combined function $H(x) = \sum_{t=1}^T w_t h_t(x)$ with normalized w.
- Let's see some demos.

Why Boosting Works?

- Our intuition is correct.
- Provably, if each h_t is better than a random guess (has error $< 1/2$), the combined function $H(x)$ could make **no error** at all!
- Besides, boosting obtains large $y_i H(x_i)$ value on each data: $H(x)$ could separate the data as **clearly** as possible.

Why Boosting Works?

- Our intuition is correct.
- Provably, if each h_t is better than a random guess (has error $< 1/2$), the combined function $H(x)$ could make **no error** at all!
- Besides, boosting obtains large $y_i H(x_i)$ value on each data: $H(x)$ could separate the data as **clearly** as possible.

Why Boosting Works?

- Our intuition is correct.
- Provably, if each h_t is better than a random guess (has error $< 1/2$), the combined function $H(x)$ could make **no error** at all!
- Besides, boosting obtains large $y_i H(x_i)$ value on each data: $H(x)$ could separate the data as **clearly** as possible.

Multi-Class Boosting (Independent Boosting)

Not very different from binary boosting.

- Input: training data $Z = (x_i, y_i)_{i=1}^N$.
- For $t = 1, 2, \cdots, T$,
  - For $c = 1, 2, \cdots, C$
    - Learn a rule alone with confidence $h_t(x, c)$ from emphasized training data.
    - Emphasize the training data that do not agree with $h_{c,t}$.
- Output: combined function $H(x, c) = \sum_{t=1}^T h_t(x, c)$.

Separate each class with the rest **independently**.

Multi-Class Boosting (Independent Boosting)

Not very different from binary boosting.

- Input: training data $Z = (x_i, y_i)_{i=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - For $c = 1, 2, \cdots, C$
    - Learn a rule alone with confidence $h_t(x, c)$ from emphasized training data.
    - Emphasize the training data that do not agree with $h_{c,t}$.
- Output: combined function $H(x, c) = \sum_{t=1}^{T} h_t(x, c)$.

Separate each class with the rest **independently**.

Problem of Independent Boosting

- Number of rules for good performance: $O(C)$. For a budget of M rules, can only use M/C rules per class.
- For example, for fruits, many of the M rules (for apple, orange, tomato, etc.) would be “it is circular.”: waste of budget.
- The rules separate each class clearly: not contain mutual information between classes.
- For example, if we separate apples with other fruits, we have no idea that apples and tomatoes look similar.
- Independent Boosting: **each class resides in its own, budget-wasting rules.**

Problem of Independent Boosting

- Number of rules for good performance: $O(C)$. For a budget of M rules, can only use M/C rules per class.
- For example, for fruits, many of the M rules (for apple, orange, tomato, etc.) would be "it is circular.": waste of budget.
- The rules separate each class clearly: not contain mutual information between classes.
- For example, if we separate apples with other fruits, we have no idea that apples and tomatoes look similar.
- Independent Boosting: **each class resides in its own, budget-wasting rules.**

Problem of Independent Boosting

- Number of rules for good performance: $O(C)$. For a budget of M rules, can only use M/C rules per class.
- For example, for fruits, many of the M rules (for apple, orange, tomato, etc.) would be "it is circular.": waste of budget.
- The rules separate each class clearly: not contain mutual information between classes.
- For example, if we separate apples with other fruits, we have no idea that apples and tomatoes look similar.
- Independent Boosting: **each class resides in its own, budget-wasting rules.**

Problem of Independent Boosting

- Number of rules for good performance: $O(C)$. For a budget of M rules, can only use M/C rules per class.
- For example, for fruits, many of the M rules (for apple, orange, tomato, etc.) would be “it is circular.”: waste of budget.
- The rules separate each class clearly: not contain mutual information between classes.
- For example, if we separate apples with other fruits, we have no idea that apples and tomatoes look similar.
- Independent Boosting: **each class resides in its own, budget-wasting rules.**

Problem of Independent Boosting

- Number of rules for good performance: $O(C)$. For a budget of M rules, can only use M/C rules per class.
- For example, for fruits, many of the M rules (for apple, orange, tomato, etc.) would be "it is circular.": waste of budget.
- The rules separate each class clearly: not contain mutual information between classes.
- For example, if we separate apples with other fruits, we have no idea that apples and tomatoes look similar.
- Independent Boosting: **each class resides in its own, budget-wasting rules.**

Joint Boosting

Try to have joint rules.

- Input: training data $Z = (x_i, y_i)_{i=1}^{N}$.
- For $t = 1, 2, \cdots, T$,
  - For $S \subseteq \{1, 2, \cdots, C\}$
    - Learn a rule alone with confidence $h_t(x, S)$ using the classes in S combined together.
  - Pick the rule $h_t(x, S_t)$ that achieves the best overall criteria.
  - Emphasize the training data that do not agree with $h_t(x, S_t)$.
- Output: combined function $H(x, c) = \sum_{c \in S_t} h_t(x, S_t)$.

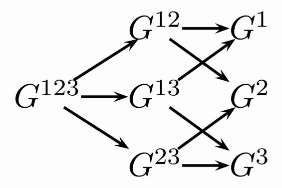
Separate a cluster of class **jointly** with the rest.

Joint Boosting

Try to have joint rules.

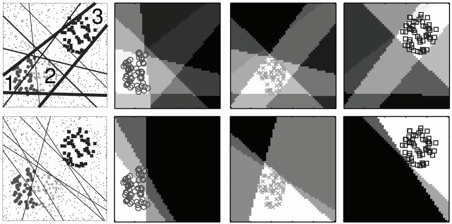
- Input: training data $Z = (x_i, y_i)_{i=1}^N$.
- For $t = 1, 2, \cdots, T$,
  - For $S \subseteq \{1, 2, \cdots, C\}$
    - Learn a rule alone with confidence $h_t(x, S)$ using the classes in S combined together.
  - Pick the rule $h_t(x, S_t)$ that achieves the best overall criteria.
  - Emphasize the training data that do not agree with $h_t(x, S_t)$.
- Output: combined function $H(x, c) = \sum_{c \in S_t} h_t(x, S_t)$.

Separate a cluster of class **jointly** with the rest.

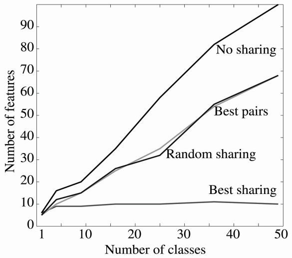


Pros of Joint Boosting

- A rule from a cluster of classes: meaningful and often stable.



- Number of rules for good performance: $O(\log C)$. Use the budget efficiently.



Cons of Joint Boosting

- The algorithm is **very slow**: $S \subseteq \{1, 2, \cdots, C\}$ is a loop of size 2^C.
- Replace the loop by a greedy search.
  - Add the best single class to the cluster.
  - Greedily combine a class to the cluster. $\cdots$
- Trace $O(C^2)$ subsets instead of $O(2^C)$.
- Still slow in general, but could speed up when $\mathcal{H}$ is simple. For example, the regression stumps

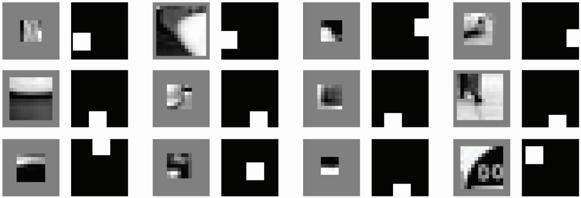
$$aI[(x)_d > \theta] + b.$$

Experiment Framework

- Goal: detect 21 objects (13 indoor, 6 outdoor, 2 both) in the picture.

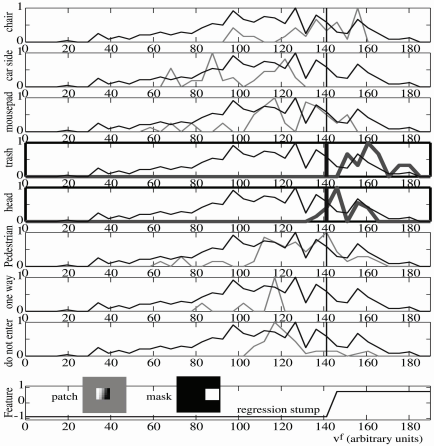
Experiment Framework (Cont'd)

- Extract feature with the following steps
  - Scale the image by σ.
  - Filter (by normalized correlation) with a patch g_f.
  - Mark the region to average response by a mask w_f.
  - Take the p-norm of average response in the region.
- Patches: small parts of the known objects – randomly generated 2000.
- Example: a feature for the stem of an apple would be a patch (matched filter to stem) with mask at the top portion.



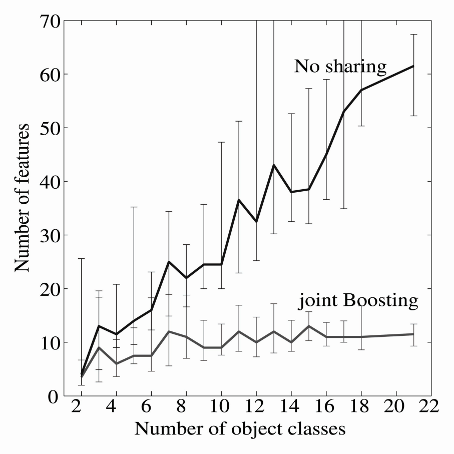
Experiment Results

- Similarity between combined classes (head and trash can).



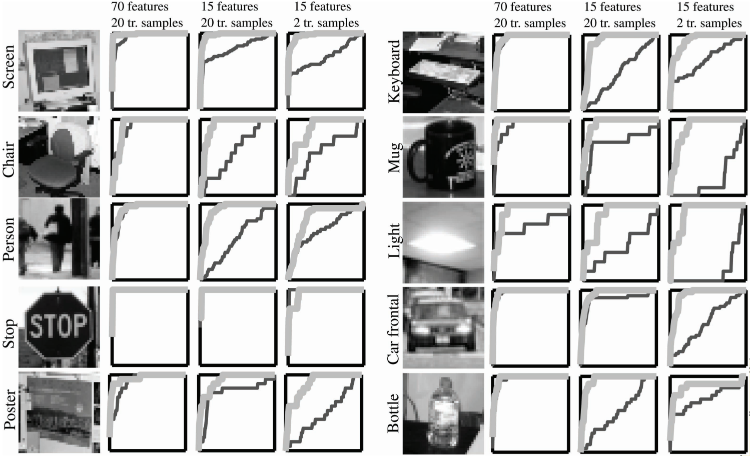
Experiment Results (Cont'd)

- Save budgets for rules.



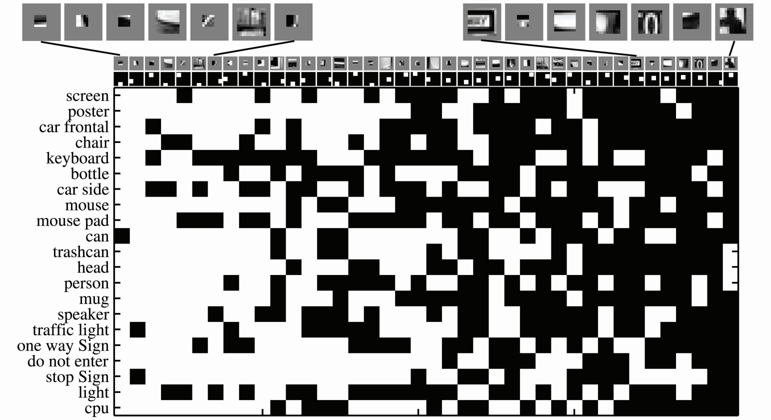
Experiment Results (Cont'd)

- Save needed data.



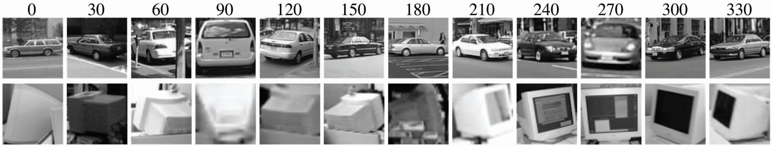
Experiment Results (Cont'd)

- Simple rules are shared by more classes.



Application: Multiview detection

- Multiview detection: usually consider each view as a class.

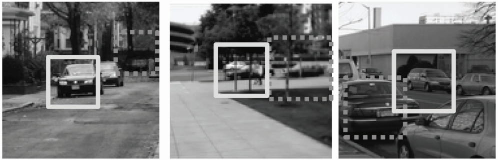


- Independent boosting: cannot allow too many classes (views).
- Views often share similar rules: joint boosting benefits.

Result: Multiview detection

- Less false alarms in detection.

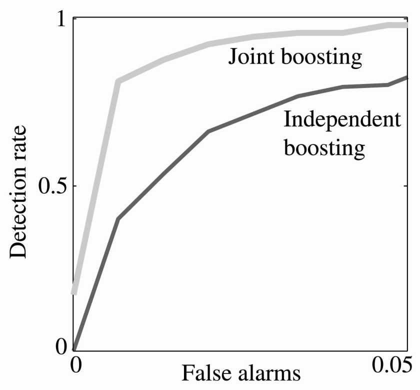
a) No sharing between views.



b) Sharing between views.

Result: Multiview detection (Cont'd)

- Significantly better ROC.



Summary

- Boosting: reweight examples and combine rules.
- Independent boosting: separate each class with the rest independently.
- Joint boosting: find best joint cluster to separate with the rest.
  - More complex algorithm.
  - More meaningful and robust classifiers.
- Utility of joint boosting:
  - When some of the classes share common rules: e.g. fruits.
  - In multiview object detection: e.g. views of cars.