Combining Models

Oliver Schulte - CMPT 726

Bishop PRML Ch. 14

Outline

Combining Models: Some Theory

Boosting

Derivation of Adaboost from the Exponential Loss Function

Outline

Combining Models: Some Theory

Boosting

Derivation of Adaboost from the Exponential Loss Function

Combining Models

- Motivation: let's say we have a number of models for a problem
  - e.g. Regression with polynomials (different degree)
  - e.g. Classification with support vector machines (kernel type, parameters)
- Often, improved performance can be obtained by combining different models.
- But how do we combine classifiers?

Why Combining Works

Intuitively, two reasons.

1. *Portfolio Diversification*: if you combine options that on average perform equally well, you keep the same average performance but you lower your risk—*variance reduction*.
   - E.g., invest in Gold and in Equities.
2. The Boosting Theorem from computational learning theory.

Probably Approximately Correct Learning

1. We have discussed generalization error in terms of the *expected error* wrt a random test set.
2. PAC learning considers the *worst-case error* wrt a random test set.
   - Guarantees bounds on test error.
3. Intuitively, a PAC guarantee works like this, for a given learning problem:
   - The theory specifies a sample size n, s.t.
   - after seeing n i.i.d. data points, with high probability $(1 - \delta)$, a classifier with training error 0 will have test error no greater than ε on any test set.
   - Leslie Valiant, Turing Award 2011.

The Boosting Theorem

- Suppose you have a learning algorithm L with a PAC guarantee that is guaranteed to have test accuracy at least 50%.
- Then you can repeatedly run L and combine the resulting classifiers in such a way that with high confidence you can achieve *any* desired degree of accuracy <100%.

Committees

- A combination of models is often called a committee
- Simplest way to combine models is to just average them together:

$$y_{COM}(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x})$$

- It turns out this simple method is better than (or same as) the individual models on average (in expectation)
  - And usually slightly better
- Example: If the errors of 5 classifiers are *independent*, then averaging predictions reduces an error rate of 10% to 1%!

Error of Individual Models

- Consider individual models $y_m(\boldsymbol{x})$, assume they can be written as true value plus error:

$$y_m(\boldsymbol{x}) = h(\boldsymbol{x}) + \epsilon_m(\boldsymbol{x})$$

- Exercise: Show that the expected value of the error of an individual model is:

$$\mathbb{E}_{\boldsymbol{x}}[\{y_m(\boldsymbol{x}) - h(\boldsymbol{x})\}^2] = \mathbb{E}_{\boldsymbol{x}}[\epsilon_m(\boldsymbol{x})^2]$$

- The average error made by an individual model is then:

$$E_{AV} = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{\boldsymbol{x}}[\epsilon_m(\boldsymbol{x})^2]$$

Error of Individual Models

- Consider individual models $y_m(\boldsymbol{x})$, assume they can be written as true value plus error:

$$y_m(\boldsymbol{x}) = h(\boldsymbol{x}) + \epsilon_m(\boldsymbol{x})$$

- Exercise: Show that the expected value of the error of an individual model is:

$$\mathbb{E}_{\boldsymbol{x}}[\{y_m(\boldsymbol{x}) - h(\boldsymbol{x})\}^2] = \mathbb{E}_{\boldsymbol{x}}[\epsilon_m(\boldsymbol{x})^2]$$

- The average error made by an individual model is then:

$$E_{AV} = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{\boldsymbol{x}}[\epsilon_m(\boldsymbol{x})^2]$$

Error of Individual Models

- Consider individual models $y_m(\boldsymbol{x})$, assume they can be written as true value plus error:

$$y_m(\boldsymbol{x}) = h(\boldsymbol{x}) + \epsilon_m(\boldsymbol{x})$$

- Exercise: Show that the expected value of the error of an individual model is:

$$\mathbb{E}_{\boldsymbol{x}}[\{y_m(\boldsymbol{x}) - h(\boldsymbol{x})\}^2] = \mathbb{E}_{\boldsymbol{x}}[\epsilon_m(\boldsymbol{x})^2]$$

- The average error made by an individual model is then:

$$E_{AV} = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{\boldsymbol{x}}[\epsilon_m(\boldsymbol{x})^2]$$

Error of Committee

- Similarly, the committee

$$y_{COM}(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x})$$

has expected error

$$E_{COM} = \mathbb{E}_{\boldsymbol{x}}\left[\left\{\left(\frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x})\right) - h(\boldsymbol{x})\right\}^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\left\{\left(\frac{1}{M}\sum_{m=1}^{M} h(\boldsymbol{x}) + \epsilon_m(\boldsymbol{x})\right) - h(\boldsymbol{x})\right\}^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\left\{\left(\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\boldsymbol{x})\right) + h(\boldsymbol{x}) - h(\boldsymbol{x})\right\}^2\right] = \mathbb{E}_{\boldsymbol{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\boldsymbol{x})\right\}^2\right]$$

Error of Committee

- Similarly, the committee

$$y_{COM}(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x})$$

has expected error

$$E_{COM} = \mathbb{E}_{\boldsymbol{x}}\left[\left\{\left(\frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x})\right) - h(\boldsymbol{x})\right\}^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\left\{\left(\frac{1}{M}\sum_{m=1}^{M} h(\boldsymbol{x}) + \epsilon_m(\boldsymbol{x})\right) - h(\boldsymbol{x})\right\}^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\left\{\left(\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\boldsymbol{x})\right) + h(\boldsymbol{x}) - h(\boldsymbol{x})\right\}^2\right] = \mathbb{E}_{\boldsymbol{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\boldsymbol{x})\right\}^2\right]$$

Error of Committee

- Similarly, the committee

$$y_{COM}(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x})$$

has expected error

$$E_{COM} = \mathbb{E}_{\boldsymbol{x}}\left[\left\{\left(\frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x})\right) - h(\boldsymbol{x})\right\}^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\left\{\left(\frac{1}{M}\sum_{m=1}^{M} h(\boldsymbol{x}) + \epsilon_m(\boldsymbol{x})\right) - h(\boldsymbol{x})\right\}^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\left\{\left(\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\boldsymbol{x})\right) + h(\boldsymbol{x}) - h(\boldsymbol{x})\right\}^2\right] = \mathbb{E}_{\boldsymbol{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\boldsymbol{x})\right\}^2\right]$$

Committee Error vs. Individual Error

- Multiplying out the inner sum over m, the committee error is

$$E_{COM} = \mathbb{E}_{\boldsymbol{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M}\epsilon_m(\boldsymbol{x})\right\}^2\right] = \frac{1}{M^2}\sum_{m=1}^{M}\sum_{n=1}^{M}\mathbb{E}_{\boldsymbol{x}}\left[\epsilon_m(\boldsymbol{x})\epsilon_n(\boldsymbol{x})\right]$$

- If we assume errors are uncorrelated, $\mathbb{E}_{\boldsymbol{x}}\left[\epsilon_m(\boldsymbol{x})\epsilon_n(\boldsymbol{x})\right] = 0$ when $m \neq n$, then:

$$E_{COM} = \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}_{\boldsymbol{x}}\left[\epsilon_m(\boldsymbol{x})^2\right] = \frac{1}{M}E_{AV}$$

- However, errors are rarely uncorrelated
  - For example, if all errors are the same, $\epsilon_m(\boldsymbol{x}) = \epsilon_n(\boldsymbol{x})$, then $E_{COM} = E_{AV}$
  - Using Jensen's inequality (convex functions), can show $E_{COM} \leq E_{AV}$

Committee Error vs. Individual Error

- Multiplying out the inner sum over m, the committee error is

$$E_{COM} = \mathbb{E}_{\boldsymbol{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M}\epsilon_m(\boldsymbol{x})\right\}^2\right] = \frac{1}{M^2}\sum_{m=1}^{M}\sum_{n=1}^{M}\mathbb{E}_{\boldsymbol{x}}\left[\epsilon_m(\boldsymbol{x})\epsilon_n(\boldsymbol{x})\right]$$

- If we assume errors are uncorrelated, $\mathbb{E}_{\boldsymbol{x}}\left[\epsilon_m(\boldsymbol{x})\epsilon_n(\boldsymbol{x})\right] = 0$ when $m \neq n$, then:

$$E_{COM} = \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}_{\boldsymbol{x}}\left[\epsilon_m(\boldsymbol{x})^2\right] = \frac{1}{M}E_{AV}$$

- However, errors are rarely uncorrelated
  - For example, if all errors are the same, $\epsilon_m(\boldsymbol{x}) = \epsilon_n(\boldsymbol{x})$, then $E_{COM} = E_{AV}$
  - Using Jensen's inequality (convex functions), can show $E_{COM} \leq E_{AV}$

Committee Error vs. Individual Error

- Multiplying out the inner sum over m, the committee error is

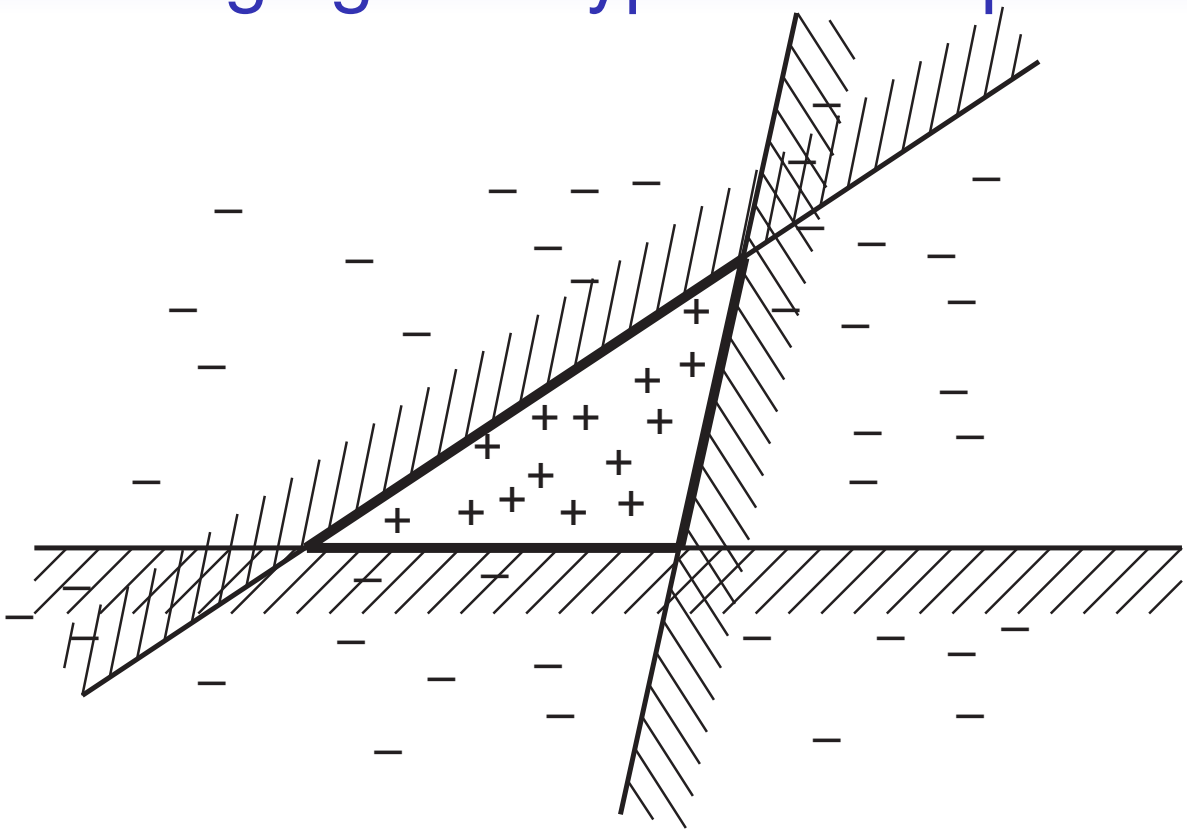
$$E_{COM} = \mathbb{E}_{\boldsymbol{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M}\epsilon_m(\boldsymbol{x})\right\}^2\right] = \frac{1}{M^2}\sum_{m=1}^{M}\sum_{n=1}^{M}\mathbb{E}_{\boldsymbol{x}}\left[\epsilon_m(\boldsymbol{x})\epsilon_n(\boldsymbol{x})\right]$$

- If we assume errors are uncorrelated, $\mathbb{E}_{\boldsymbol{x}}\left[\epsilon_m(\boldsymbol{x})\epsilon_n(\boldsymbol{x})\right] = 0$ when $m \neq n$, then:

$$E_{COM} = \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}_{\boldsymbol{x}}\left[\epsilon_m(\boldsymbol{x})^2\right] = \frac{1}{M}E_{AV}$$

- However, errors are rarely uncorrelated
  - For example, if all errors are the same, $\epsilon_m(\boldsymbol{x}) = \epsilon_n(\boldsymbol{x})$, then $E_{COM} = E_{AV}$
  - Using Jensen's inequality (convex functions), can show $E_{COM} \leq E_{AV}$

Enlarging the Hypothesis space



- Classifier committees are more expressive than a single classifier.
- Example: classify as positive if all three threshold classifiers classify as positive.
- Figure Russell and Norvig 18.32.

Outline

Combining Models: Some Theory

Boosting

Derivation of Adaboost from the Exponential Loss Function

Outline

Combining Models: Some Theory

Boosting

Derivation of Adaboost from the Exponential Loss Function

Boosting

- Boosting is a technique for combining classifiers into a committee
  - We describe AdaBoost (adaptive boosting), the most commonly used variant. (Freund and Schapire 1995, Gödel Prize 2003).
- Boosting is a meta-learning technique
  - Combines a set of classifiers trained using their own learning algorithms
  - Magic: can work well even if those classifiers only perform slightly better than random!

Boosting Model

- We consider two-class classification problems, training data $(\boldsymbol{x}_i, t_i)$, with $t_i \in \{-1, 1\}$
- In boosting we build a "linear" classifier of the form:

$$y(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m y_m(\boldsymbol{x})$$

- A committee of classifiers, with weights
- In boosting terminology:
  - Each $y_m(\boldsymbol{x})$ is called a weak learner or base classifier
  - Final classifier $y(\boldsymbol{x})$ is called strong learner
- Learning problem: how do we choose the weak learners $y_m(\boldsymbol{x})$ and weights α_m?

Boosting Model

- We consider two-class classification problems, training data $(\boldsymbol{x}_i, t_i)$, with $t_i \in \{-1, 1\}$
- In boosting we build a "linear" classifier of the form:

$$y(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m y_m(\boldsymbol{x})$$

- A committee of classifiers, with weights
- In boosting terminology:
  - Each $y_m(\boldsymbol{x})$ is called a weak learner or base classifier
  - Final classifier $y(\boldsymbol{x})$ is called strong learner
- Learning problem: how do we choose the weak learners $y_m(\boldsymbol{x})$ and weights α_m?

Boosting Model

- We consider two-class classification problems, training data $(\boldsymbol{x}_i, t_i)$, with $t_i \in \{-1, 1\}$
- In boosting we build a "linear" classifier of the form:

$$y(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m y_m(\boldsymbol{x})$$

- A committee of classifiers, with weights
- In boosting terminology:
  - Each $y_m(\boldsymbol{x})$ is called a weak learner or base classifier
  - Final classifier $y(\boldsymbol{x})$ is called strong learner
- Learning problem: how do we choose the weak learners $y_m(\boldsymbol{x})$ and weights α_m?

Community Notes on Boosting

- Boosting with Decision Trees was used by Dugan O'Neill (SFU, Physics) to find evidence for the top quark. (Yes, this is a big deal.) `http://www.phy.bnl.gov/edg/samba/oneil_summary.pdf`.
- Boosting demo `http://cseweb.ucsd.edu/~yfreund/adaboost/index.html`.

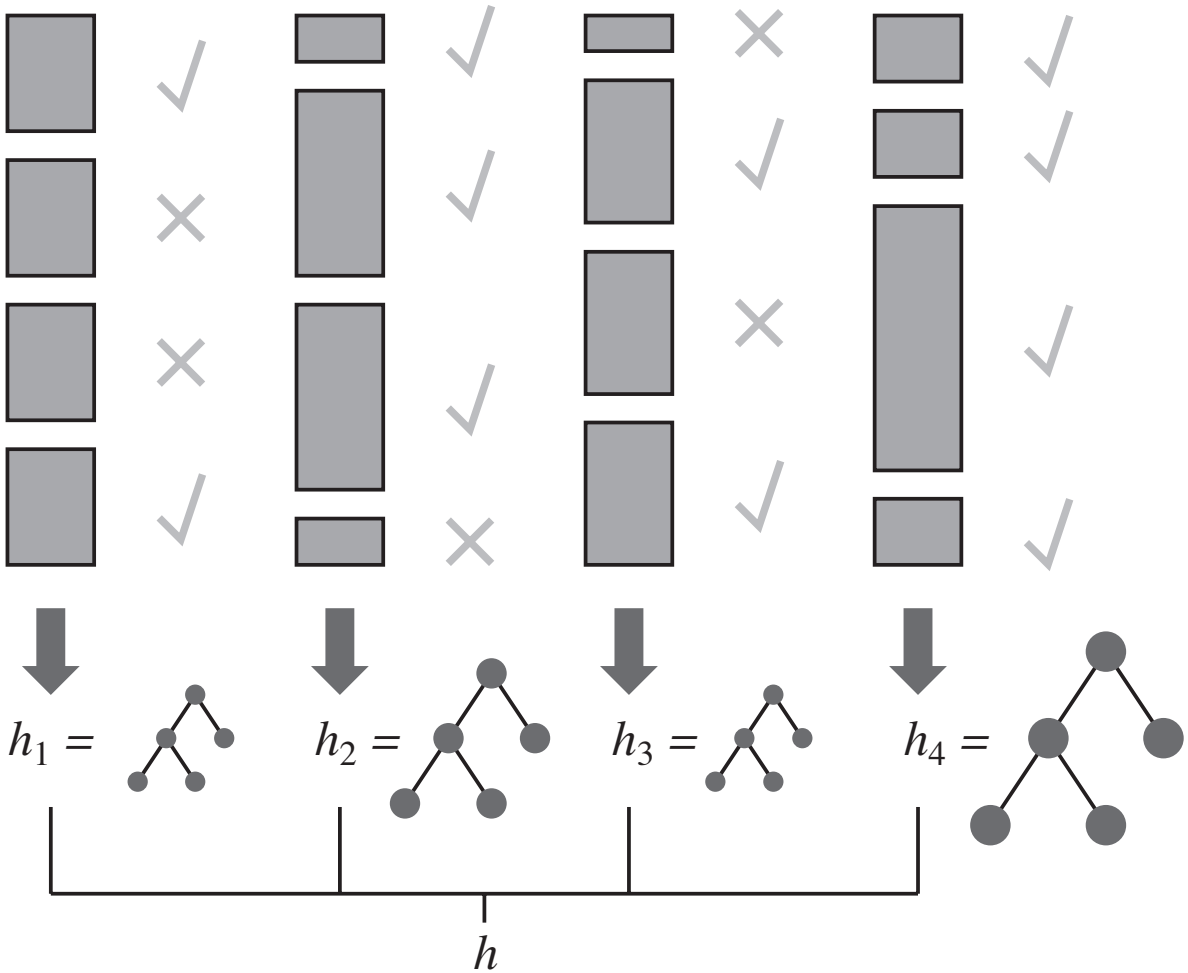
Boosting Intuition

- The weights α_k reflect the training error of the different classifiers.
- Classifier α_{k+1} is trained on *weighted examples*, where instances misclassified by the committee

$$y^k(\boldsymbol{x}) = \sum_{m=1}^{k} \alpha_m y_m(\boldsymbol{x})$$

  receive higher weight.
- The instance weights can be interpreted as *resampling*: build a new sample where instances with higher weight occur more frequently.

Example - Boosting Decision Trees



- Shaded rectangle: classification example
- Sizes of rectangles, trees = weight

Example - Thresholds

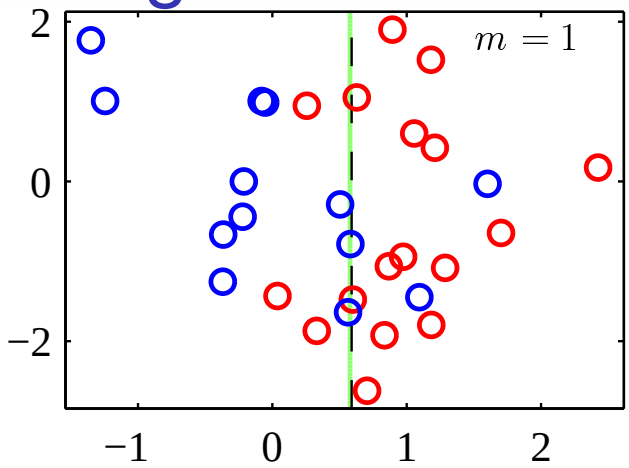
- Let's consider a simple example where weak learners are thresholds
- i.e. Each $y_m(\boldsymbol{x})$ is of the form:

$$y_m(\boldsymbol{x}) = x_i > \theta$$

- To allow different directions of threshold, include $p \in \{-1, +1\}$:

$$y_m(\boldsymbol{x}) = px_i > p\theta$$

Choosing Weak Learners



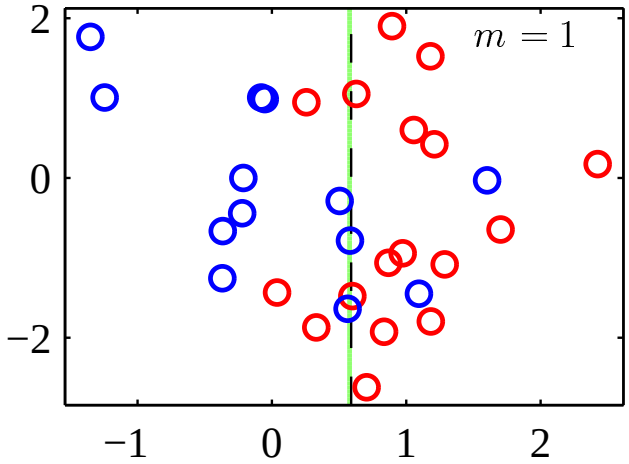
- Boosting is a greedy strategy for building the strong learner

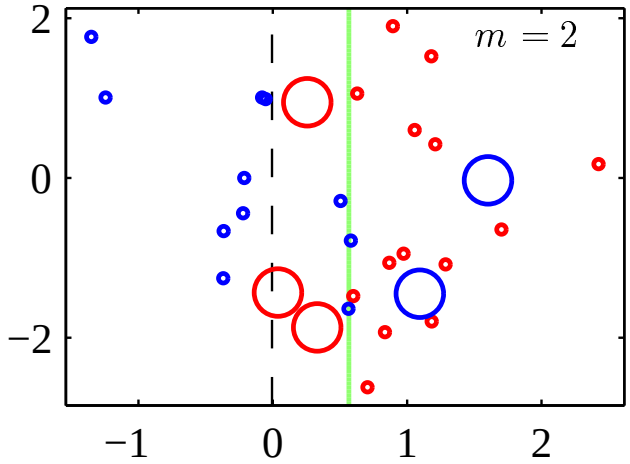
$$y(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m y_m(\boldsymbol{x})$$

- Start by choosing the best weak learner, use it as $y_1(\boldsymbol{x})$
  - Best is defined as that which minimizes number of mistakes made (0-1 classification loss)
- i.e. Search over all p, θ, i to find best

$$y_1(\boldsymbol{x}) = px_i > p\theta$$

Choosing Weak Learners

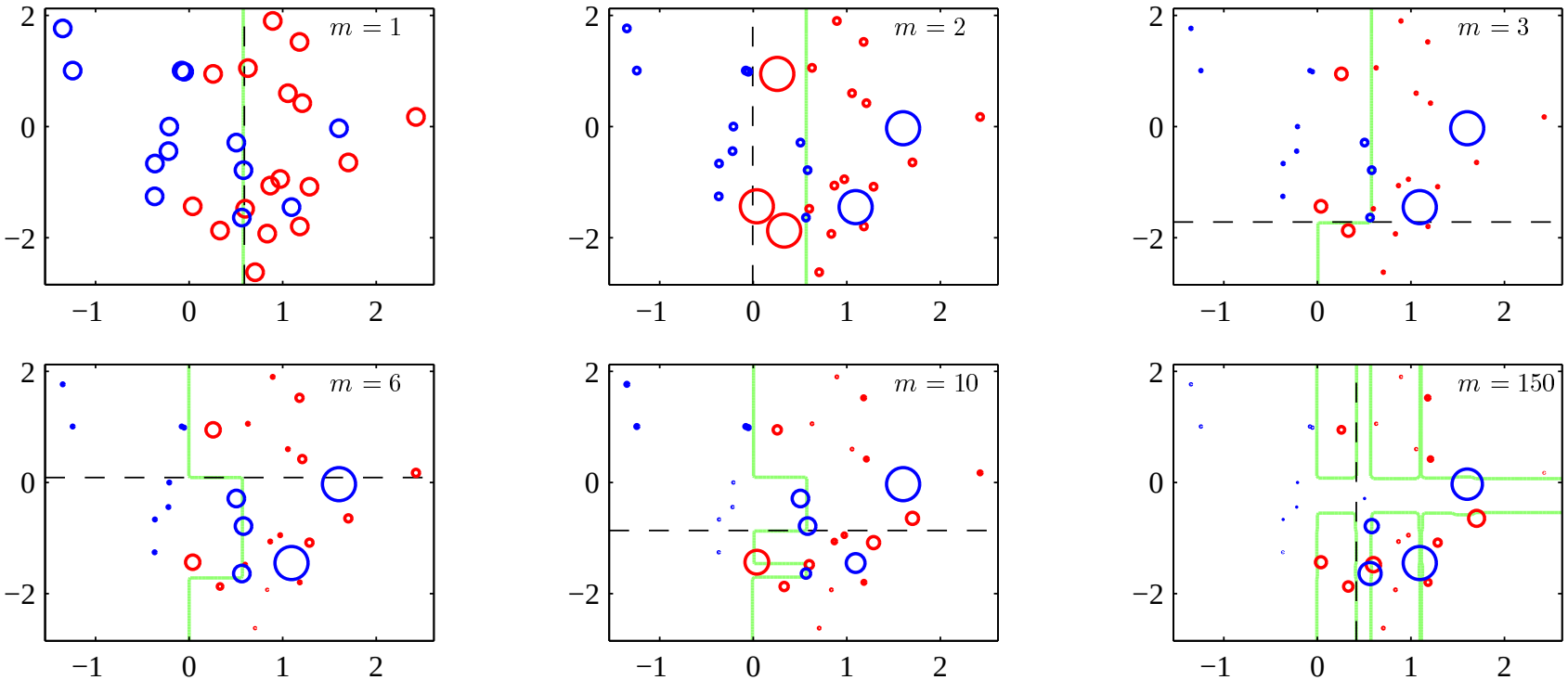




- The first weak learner $y_1(\boldsymbol{x})$ made some mistakes
- Choose the second weak learner $y_2(\boldsymbol{x})$ to try to get those ones correct
  - Best is now defined as that which minimizes weighted number of mistakes made
  - Higher weight given to those $y_1(\boldsymbol{x})$ got incorrect
- Strong learner now

$$y(\boldsymbol{x}) = \alpha_1 y_1(\boldsymbol{x}) + \alpha_2 y_2(\boldsymbol{x})$$

Choosing Weak Learners



- Repeat: reweight examples and choose new weak learner based on weights
- Green line shows decision boundary of strong learner

What About Those Weights?

- So exactly how should we choose the weights for the examples when classified incorrectly?
- And what should the α_m be for combining the weak learners $y_m(\boldsymbol{x})$?
- Original approach: make sure the strong learner satisfies the PAC guarantee.
- Alternative view: define a loss function, and choose parameters to minimize it.

AdaBoost Algorithm

- Initialize weights $w_n^{(1)} = 1/N$
- For $m = 1, \ldots, M$ (and while $\epsilon_m < 1/2$)
  - Find weak learner $y_m(\boldsymbol{x})$ with minimum weighted error

$$\epsilon_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n)$$

  - With normalized weights, $\epsilon_m =$ probability of mistake.
  - Set $\alpha_m = \frac{1}{2} \ln \frac{1-\epsilon_m}{\epsilon_m}$
  - Update weights $w_n^{(m+1)} = w_n^{(m)} \exp\{-\alpha_m t_n y_m(\boldsymbol{x}_n)\}$
  - Normalize weights to sum to one
- Final classifier is

$$y(\boldsymbol{x}) = sign\left(\sum_{m=1}^{M} \alpha_m y_m(\boldsymbol{x})\right)$$

Outline

Combining Models: Some Theory

Boosting

Derivation of Adaboost from the Exponential Loss Function

Exponential Loss

- Boosting attempts to minimize the exponential loss

$$E_n = \exp\{-t_n y(\boldsymbol{x}_n)\}$$

  error on n^{th} training example
- Exponential loss is differentiable approximation to 0/1 loss
  - Better for optimization
- Total error

$$E = \sum_{n=1}^{N} \exp\{-t_n y(\boldsymbol{x}_n)\}$$

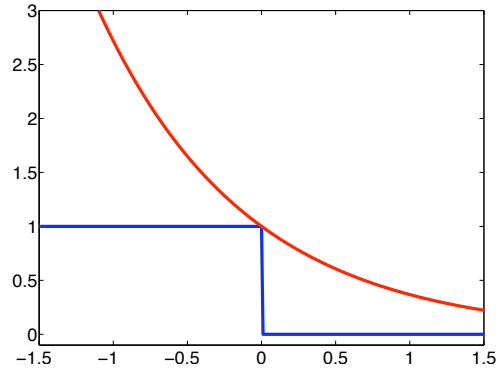


figure from G. Shakhnarovich

Exponential Loss

- Boosting attempts to minimize the exponential loss

$$E_n = \exp\{-t_n y(\boldsymbol{x}_n)\}$$

  error on n^{th} training example
- Exponential loss is differentiable approximation to 0/1 loss
  - Better for optimization
- Total error

$$E = \sum_{n=1}^{N} \exp\{-t_n y(\boldsymbol{x}_n)\}$$

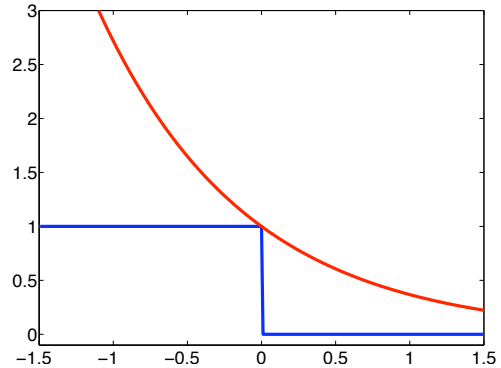


figure from G. Shakhnarovich

Minimizing Exponential Loss

- Let's assume we've already chosen weak learners $y_1(\boldsymbol{x}), \ldots, y_{m-1}(\boldsymbol{x})$ and their weights $\alpha_1, \ldots, \alpha_{m-1}$
  - Define $f_{m-1}(\boldsymbol{x}) = \alpha_1 y_1(\boldsymbol{x}) + \ldots + \alpha_{m-1} y_{m-1}(\boldsymbol{x})$
- Just focus on choosing $y_m(\boldsymbol{x})$ and α_m
  - Greedy optimization strategy
- Total error using exponential loss is:

$$
\begin{aligned}
E &= \sum_{n=1}^{N} \exp\{-t_n y(\boldsymbol{x}_n)\} = \sum_{n=1}^{N} \exp\{-t_n [f_{m-1}(\boldsymbol{x}_n) + \alpha_m y_m(\boldsymbol{x})]\} \\
&= \sum_{n=1}^{N} \exp\{-t_n f_{m-1}(\boldsymbol{x}_n) - t_n \alpha_m y_m(\boldsymbol{x})\} \\
&= \sum_{n=1}^{N} \underbrace{\exp\{-t_n f_{m-1}(\boldsymbol{x}_n)\}}_{\text{weight } w_n^{(m)}} \exp\{-t_n \alpha_m y_m(\boldsymbol{x})\}
\end{aligned}
$$

Minimizing Exponential Loss

- Let's assume we've already chosen weak learners $y_1(\boldsymbol{x}), \ldots, y_{m-1}(\boldsymbol{x})$ and their weights $\alpha_1, \ldots, \alpha_{m-1}$
  - Define $f_{m-1}(\boldsymbol{x}) = \alpha_1 y_1(\boldsymbol{x}) + \ldots + \alpha_{m-1} y_{m-1}(\boldsymbol{x})$
- Just focus on choosing $y_m(\boldsymbol{x})$ and α_m
  - Greedy optimization strategy
- Total error using exponential loss is:

$$
\begin{aligned}
E &= \sum_{n=1}^{N} \exp\{-t_n y(\boldsymbol{x}_n)\} = \sum_{n=1}^{N} \exp\{-t_n [f_{m-1}(\boldsymbol{x}_n) + \alpha_m y_m(\boldsymbol{x})]\} \\
&= \sum_{n=1}^{N} \exp\{-t_n f_{m-1}(\boldsymbol{x}_n) - t_n \alpha_m y_m(\boldsymbol{x})\} \\
&= \sum_{n=1}^{N} \underbrace{\exp\{-t_n f_{m-1}(\boldsymbol{x}_n)\}}_{\text{weight } w_n^{(m)}} \exp\{-t_n \alpha_m y_m(\boldsymbol{x})\}
\end{aligned}
$$

Minimizing Exponential Loss

- Let's assume we've already chosen weak learners $y_1(\boldsymbol{x}), \ldots, y_{m-1}(\boldsymbol{x})$ and their weights $\alpha_1, \ldots, \alpha_{m-1}$
  - Define $f_{m-1}(\boldsymbol{x}) = \alpha_1 y_1(\boldsymbol{x}) + \ldots + \alpha_{m-1} y_{m-1}(\boldsymbol{x})$
- Just focus on choosing $y_m(\boldsymbol{x})$ and α_m
  - Greedy optimization strategy
- Total error using exponential loss is:

$$
\begin{aligned}
E &= \sum_{n=1}^{N} \exp\{-t_n y(\boldsymbol{x}_n)\} = \sum_{n=1}^{N} \exp\{-t_n [f_{m-1}(\boldsymbol{x}_n) + \alpha_m y_m(\boldsymbol{x})]\} \\
&= \sum_{n=1}^{N} \exp\{-t_n f_{m-1}(\boldsymbol{x}_n) - t_n \alpha_m y_m(\boldsymbol{x})\} \\
&= \sum_{n=1}^{N} \underbrace{\exp\{-t_n f_{m-1}(\boldsymbol{x}_n)\}}_{\text{weight } w_n^{(m)}} \exp\{-t_n \alpha_m y_m(\boldsymbol{x})\}
\end{aligned}
$$

Weighted Loss

- On the m^{th} iteration of boosting, we are choosing y_m and α_m to minimize the weighted loss:

$$E = \sum_{n=1}^{N} w_n^{(m)} \exp\{-t_n \alpha_m y_m(\boldsymbol{x})\}$$

  where $w_n^{(m)} = \exp\{-t_n f_{m-1}(\boldsymbol{x}_n)\}$
- Can define these as weights since they are constant wrt y_m and α_m

Minimization wrt y_m

- Consider the weighted loss

$$E = \sum_{n=1}^{N} w_n^{(m)} e^{-t_n \alpha_m y_m(\boldsymbol{x})} = e^{-\alpha_m} \sum_{n \in \mathcal{T}_m} w_n^{(m)} + e^{\alpha_m} \sum_{n \in \mathcal{M}_m} w_n^{(m)}$$

  where $\mathcal{T}_m$ is the set of points correctly classified by the choice of $y_m(\boldsymbol{x})$, and $\mathcal{M}_m$ those that are not

$$E = e^{\alpha_m} \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n) + e^{-\alpha_m} \sum_{n=1}^{N} w_n^{(m)} (1 - I(y_m(\boldsymbol{x}_n) \neq t_n))$$

$$= (e^{\alpha_m} - e^{-\alpha_m}) \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n) + e^{-\alpha_m} \sum_{n=1}^{N} w_n^{(m)}$$

- Since the second term is a constant wrt y_m and $e^{\alpha_m} - e^{-\alpha_m} > 0$ if $\alpha_m > 0$, best y_m minimizes weighted 0-1 loss $\sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n)$.

Minimization wrt y_m

- Consider the weighted loss

$$E = \sum_{n=1}^{N} w_n^{(m)} e^{-t_n \alpha_m y_m(\boldsymbol{x})} = e^{-\alpha_m} \sum_{n \in \mathcal{T}_m} w_n^{(m)} + e^{\alpha_m} \sum_{n \in \mathcal{M}_m} w_n^{(m)}$$

where $\mathcal{T}_m$ is the set of points correctly classified by the choice of $y_m(\boldsymbol{x})$, and $\mathcal{M}_m$ those that are not

$$\begin{aligned} E &= e^{\alpha_m} \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n) + e^{-\alpha_m} \sum_{n=1}^{N} w_n^{(m)} (1 - I(y_m(\boldsymbol{x}_n) \neq t_n)) \\ &= (e^{\alpha_m} - e^{-\alpha_m}) \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n) + e^{-\alpha_m} \sum_{n=1}^{N} w_n^{(m)} \end{aligned}$$

- Since the second term is a constant wrt y_m and $e^{\alpha_m} - e^{-\alpha_m} > 0$ if $\alpha_m > 0$, best y_m minimizes weighted 0-1 loss $\sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n)$.

Minimization wrt y_m

- Consider the weighted loss

$$E = \sum_{n=1}^{N} w_n^{(m)} e^{-t_n \alpha_m y_m(\boldsymbol{x})} = e^{-\alpha_m} \sum_{n \in \mathcal{T}_m} w_n^{(m)} + e^{\alpha_m} \sum_{n \in \mathcal{M}_m} w_n^{(m)}$$

  where $\mathcal{T}_m$ is the set of points correctly classified by the choice of $y_m(\boldsymbol{x})$, and $\mathcal{M}_m$ those that are not

$$\begin{aligned} E &= e^{\alpha_m} \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n) + e^{-\alpha_m} \sum_{n=1}^{N} w_n^{(m)} (1 - I(y_m(\boldsymbol{x}_n) \neq t_n)) \\ &= (e^{\alpha_m} - e^{-\alpha_m}) \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n) + e^{-\alpha_m} \sum_{n=1}^{N} w_n^{(m)} \end{aligned}$$

- Since the second term is a constant wrt y_m and $e^{\alpha_m} - e^{-\alpha_m} > 0$ if $\alpha_m > 0$, best y_m minimizes weighted 0-1 loss $\sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n)$.

Choosing α_m

- So best y_m minimizes weighted 0-1 loss regardless of α_m
- How should we set α_m given this best y_m?
- Recall from above:

$$\begin{aligned} E &= e^{\alpha_m} \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n) + e^{-\alpha_m} \sum_{n=1}^{N} w_n^{(m)} (1 - I(y_m(\boldsymbol{x}_n) \neq t_n)) \\ &= e^{\alpha_m} \epsilon_m + e^{-\alpha_m} (1 - \epsilon_m) \end{aligned}$$

  where we define ϵ_m to be the weighted error of y_m
- Calculus: $\alpha_m = \frac{1}{2} \ln \frac{1-\epsilon_m}{\epsilon_m}$ minimizes E.

Choosing α_m

- So best y_m minimizes weighted 0-1 loss regardless of α_m
- How should we set α_m given this best y_m?
- Recall from above:

$$\begin{aligned} E &= e^{\alpha_m} \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n) + e^{-\alpha_m} \sum_{n=1}^{N} w_n^{(m)} (1 - I(y_m(\boldsymbol{x}_n) \neq t_n)) \\ &= e^{\alpha_m} \epsilon_m + e^{-\alpha_m} (1 - \epsilon_m) \end{aligned}$$

  where we define ϵ_m to be the weighted error of y_m

- Calculus: $\alpha_m = \frac{1}{2} \ln \frac{1-\epsilon_m}{\epsilon_m}$ minimizes E.

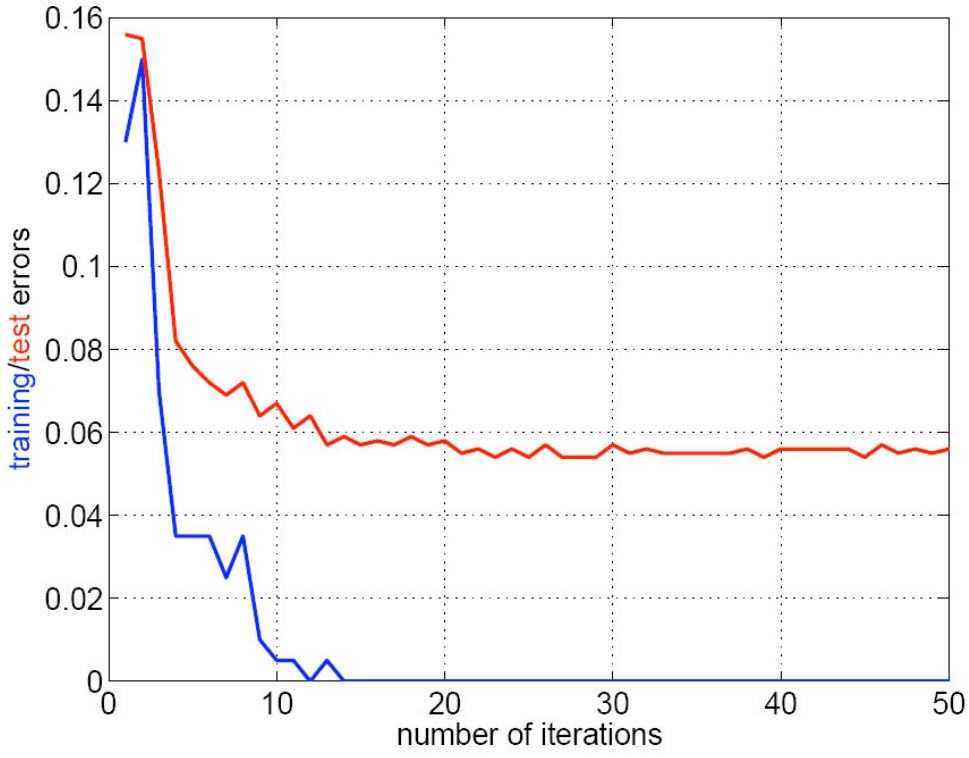
Choosing α_m

- So best y_m minimizes weighted 0-1 loss regardless of α_m
- How should we set α_m given this best y_m?
- Recall from above:

$$\begin{aligned} E &= e^{\alpha_m} \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n) + e^{-\alpha_m} \sum_{n=1}^{N} w_n^{(m)} (1 - I(y_m(\boldsymbol{x}_n) \neq t_n)) \\ &= e^{\alpha_m} \epsilon_m + e^{-\alpha_m} (1 - \epsilon_m) \end{aligned}$$

  where we define ϵ_m to be the weighted error of y_m
- Calculus: $\alpha_m = \frac{1}{2} \ln \frac{1-\epsilon_m}{\epsilon_m}$ minimizes E.

AdaBoost Behaviour



- Typical behaviour:
  - Test error decreases even after training error is flat (even zero!)
  - Tends not to overfit

from G. Shakhnarovich

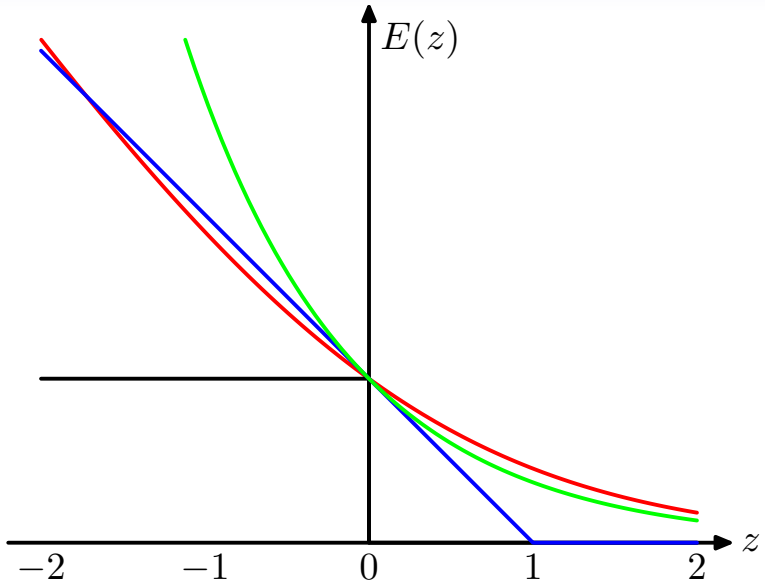
Boosting the Margin

- Define the margin of an example:

$$\gamma(\boldsymbol{x}_i) = t_i \frac{\alpha_1 y_1(\boldsymbol{x}_i) + \ldots + \alpha_m y_m(\boldsymbol{x}_i)}{\alpha_1 + \ldots + \alpha_m}$$

- Margin is 1 iff all y_i classify correctly, -1 if none do
- Iterations of AdaBoost increase the margin of training examples (even after training error is zero)
- Intuitively, classifier becomes more “definite”.

Loss Functions for Classification



- We revisit a graph from earlier: 0-1 loss, SVM hinge loss, logistic regression cross-entropy loss, and AdaBoost exponential loss are shown
- All are approximations (upper bounds) to 0-1 loss
- Exponential loss leads to simple greedy optimization scheme
- But it has problems with outliers: note different behaviour compared to logistic regression cross-entropy loss for badly mis-classified examples.

Conclusion

- Readings: Ch. 14.3, 14.4
- Methods for combining models
  - Simple averaging into a committee
  - Greedy selection of models to minimize exponential loss (AdaBoost)