Boosting as a Product of Experts

Narayanan U. Edakunni, Gavin Brown and Tim Kovacs
University of Bristol and University of Manchester, UK

Introduction

► We establish a generic probabilistic model of boosting.
► The class probability is represented as a Product of Experts (PoE).
► Boosting is modeled as addition of new experts to the PoE with the addition of an expert increasing the likelihood at each iteration.
► Condition that the likelihood does not decrease with addition of an expert leads to the weak learning criterion.
► We show that for a specific parameterization of the expert probabilities, PoE = AdaBoost.
► Using a specific form of expert probability, POEBoost.CS is developed, which uses base hypotheses with probabilistic outputs.
► The comparisons of POEBoost.CS with existing boosting algorithm illustrates the advantages of having a probabilistic model for boosting.

Product of Experts [Hinton, 2002]

The conditional class probability of label y given input x is given by
\[ P(Y = y | X = x, h_1, \ldots h_j) = \prod_{i=1}^{j} P(Y = y | X = x, h_i) \]
and
\[ P(Y = y | X = x, h_1, \ldots h_j) = \prod_{i=1}^{j} P(Y = y | X = x, h_i) \]
where Y and X are random variables for output and input, Y is the complement of y and h the hypothesis.

Incremental expansion of PoE

Addition of an expert \( P(y|x, h) \) to a PoE \( P_{j-1} \) with \( j - 1 \) experts gives
\[ P_j(y|x) = \frac{P_j(y|x, h_j)P_j(y|x) + P_j(y|x, h_j)P_j(y|x)}{P_j(y|x)} \]
where \( P_j \) is a PoE with j experts.

Monotonic increase in likelihood

► The log likelihood is
\[ L_j = \log P(y_1, \ldots, y_N|x_1, \ldots, x_N, h_1, \ldots, h_j) \]
which the data points (\( x_1, y_1 \)) \ldots (\( x_N, y_N \)) are assumed to be IID.
► With every addition of an expert, the likelihood does not decrease if
\[ \Delta L_j = L_j - L_{j-1} \geq 0 \]
It leads to a constraint on the new expert
\[ \sum_{i=1}^{N} D_{i}^{j-1} = 1 \]
where
\[ \sum_{i=1}^{N} D_{i}^{j-1} = P(Y = y | x_i, h_{j-1}) \]
and
\[ \sum_{i=1}^{N} D_{i}^{j-1} = P(Y = y | x_i, h_{j-1}) \]
We can now come up with a procedure to expand the PoE incrementally using eq. (1) to add new hypotheses, with each hypothesis satisfying the constraint given in eq. (2).

Algorithm : Incremental learning in PoE

Training data : \( S = (x_1, y_1) \ldots (x_N, y_N) \)
\[ D_{i}^{j} = \frac{1}{N} \text{ for } i = 1 \ldots N \]
for \( j = 1 \) to \( M \) do
Obtain a hypothesis \( h_j \) such that
\[ \sum_{i=1}^{N} D_{i}^{j-1} \leq \frac{1}{2} \]
\[ D_{i}^{j-1} = P(Y = y | x_i, h_j)D_{i}^{j} \]
\[ \text{end for} \]

POEBoost.CS : Continuous Symmetric expert

► We now slightly adapt the assumptions of POEBoost.CS to obtain a variant.
► Symmetric class error probability
\[ \text{Error probability parameterization} \]
\[ e^{-\alpha} \]
We can now derive an algorithm that can handle base hypotheses with probabilistic class predictions, without a large modification in the algorithm.

POEBoost.CS : Discrete Symmetric expert

► We assume a certain form for the expert probability.
► Parametrize expert probability
\[ P(Y = y | x, h) = P(Y = y | x, h)P(Z = y | x, h) + P(Y = y | x, \bar{h})P(Z = y | x, \bar{h}) \]
Symmetric class error probability
\[ P(Y = y | x, h)P(Z = y | x, h) = P(Y = y | x, \bar{h})P(Z = y | x, \bar{h}) \]
Determining class error probability
\[ P(Y = y | x, h)P(Z = y | x, h) \in \{0, 1\} \]
Error probability parameterization
\[ P_{j} = e^{-\alpha} \]
We show that these assumptions on the expert reduces incremental learning in a PoE to AdaBoost.

Product of Experts = majority voting

Conditional class probability is given by
\[ P(Y = y | x, h) = e^{-\alpha} \sum_{i=1}^{N} y_i h_i(x_i) \]
this is equivalent to prediction rule of AdaBoost [Freund and Schapire, 1997].

Incremental expansion of PoE = updating weights

The weights on the data points \( D_{i}^{j-1} \) is updated with the addition of an expert \( h_j \) with voting weight \( \alpha_j \) as
\[ D_{i}^{j} = e^{-\alpha_j \sum_{i=1}^{N} h_i(x_i)}D_{i}^{j-1} \]
similar to that of the AdaBoost algorithm.

Monotonic increase in likelihood = weak learning condition

► The likelihood does not decrease with the addition of an expert if
\[ \epsilon_j \leq \frac{1}{2} \]
where
\[ \epsilon_j = \sum_{i=1}^{N} h_i(x_i)D_{i}^{j-1} \]
The likelihood constraint on \( h_j \) equivalent to that of weak learning condition of AdaBoost [Freund and Schapire, 1997, Schapire, 1990].

► The update rule for \( \alpha_j \)
\[ 0 \leq \alpha_j \leq \frac{1}{2} \log \frac{1 - \epsilon_j}{\epsilon_j} \]
AdaBoost uses \( \frac{1}{2} \log \frac{1 - \epsilon_j}{\epsilon_j} \) for \( \alpha_j \).

Weak learning condition

The likelihood does not decrease with the addition of an expert if
\[ \delta_j \leq P_{\delta} \leq \frac{1}{2} \]
where
\[ \delta_j = \sum_{i=1}^{N} D_{i}^{j-1} \]
The likelihood constraint on \( h_j \) to AdaBoost [Freund and Schapire, 1997, Schapire, 1990].

► The update rule for \( \alpha_j \)
\[ 0 \leq \alpha_j \leq \frac{1}{2} \log \frac{1 - \epsilon_j}{\epsilon_j} \]
AdaBoost uses \( \frac{1}{2} \log \frac{1 - \epsilon_j}{\epsilon_j} \) for \( \alpha_j \).

Conclusions

► We have presented a novel interpretation of boosting algorithms as a PoE.
► The sequential distribution updates in the boosting explained as an iterative model selection process.
► The model we have presented can incorporate arbitrary probabilistic models as the experts, including real-valued predictions naturally.
► In a particular case, the model reduces to an AdaBoost-like algorithm.

References

A decision-theoretic generalization of on-line learning and an application to boosting.
Training products of experts by minimizing contrastive divergence.
The strength of weak learnability.
Improved boosting algorithms using confidence-rated predictions.

http://www.cs.bris.ac.uk/~nara
nara@cs.bris.ac.uk