Notes on Boosting

Geoff Gordon
ggordon@cs.cmu.edu

October 18, 1999

Overview

Boosting, online learning, linear/convex programming

AdaBoost, WM, SVM

All related

Related reading

"Drifting Games," Rob Schapire

"Boosting as Entropy Projection," Jyrki Kivinen and Manfred Warmuth

"Additive Models, Boosting, and Inference for Generalized Divergences," John Lafferty

All the above are in COLT99 (`www.acm.org/dl/`)

"A Tutorial on Boosting," Freund and Schapire (`www.research.att.com/~{yoav,schapire}`)

Boosting

Training data D (instance $\mapsto \{0, 1\}$)

Weak learner: distribution on D $\mapsto$ weak hypothesis (w.h.p.)

Weak hypothesis = right more often than not $\left(\frac{1+\gamma}{2}\right)$

Boosting cont'd

Run weak learner many times with different distributions

Final hypothesis = (weighted) majority vote

Hopefully final hypothesis is better than weak hypotheses

Boosting as LP

Problem of combining weak learners is a (semi-infinite) LP

Write $X = (x_{ij})$ for mistake matrix:

$$x_{ij} = \begin{cases} +1 & \text{hypothesis } j \text{ gets sample } i \text{ right} \\ -1 & j \text{ gets } i \text{ wrong} \end{cases}$$

Want weights w so $Xw > 0$

Boosting as LP cont'd

X is short and fat:

$$
\begin{array}{cc}
 & \text{Infinitely many weak hypotheses} \\
\begin{array}{c} \\ n \\ \\ \mathsf{e} \\ \mathsf{x} \\ \mathsf{s} \\ \\ \end{array} &
\left(\begin{array}{ccccccc}
+1 & +1 & -1 & +1 & -1 & +1 & \ldots \\
+1 & -1 & -1 & -1 & -1 & -1 & \ldots \\
-1 & -1 & +1 & -1 & +1 & -1 & \ldots \\
+1 & -1 & +1 & +1 & +1 & +1 & \ldots \\
-1 & +1 & +1 & +1 & -1 & +1 & \ldots \\
+1 & +1 & +1 & +1 & +1 & -1 & \ldots \\
-1 & +1 & -1 & +1 & +1 & +1 & \ldots
\end{array}\right)
\end{array}
$$

Boosting as LP cont'd

Either minimize $\|w\|$ for $Xw \geq 1$

Or bound $\|w\|$ and maximize ϵ for $Xw \geq \epsilon$

ϵ is called margin—compare SVM

Boosting as LP cont'd

Problem: infinitely many weak hypotheses

Solution: column generation

Alternately:

- Solve for finite number of columns
- Pick a new column to add

Picking a new column

Current mistake matrix is $X^{(t)}$

Have solution w and ϵ with

$$X^{(t)} w \geq \epsilon$$

Also have dual variables $d \geq 0$ with

$$\sum_i d_i = 1$$

$$d^{\top} X^{(t)} \approx 0$$

Picking a new column cont'd

d is distribution on which previous weak hypotheses' performance is as close as possible to chance

Want to add column $x^{(t)}$ with $d^\mathsf{T} x^{(t)}$ far from 0

Feed d as distribution to weak learner

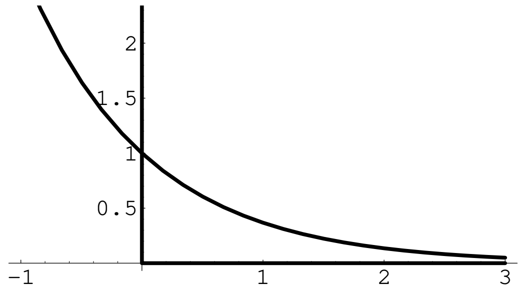
Guaranteed to get $d^\mathsf{T} x^{(t)} \geq \gamma$

Intuition: force weak learner to learn something new

AdaBoost

Most popular boosting algorithm

Only approximates above program



Minimizes $\sum_i \exp(-[Xw]_i)$ (proxy for optimizing margin)

AdaBoost cont'd

Only considers one weight at a time:

- leave previous weights fixed
- optimize only weight of $x^{(t)}$

Consequences:

- $d^{\mathsf{T}} x^{(t)} = 0$
- Can solve for optimal w and d in closed form

In practice

AdaBoost quickly reduces training error to zero

Also increases “margin,” even after training error is zero

AdaBoost has been more successful than

- SVM-like algorithms for boosting
- Algorithms which adjust all weights at once

Connection to WM

Optimal d computed from d^{old} by scaling down the probabilities of examples the last weak hypothesis got right

Looks strangely like WM, but in reverse:

- In WM, fixed number of experts, one example at a time
- In AdaBoost, fixed number of examples, one expert at a time
- Weights in WM are distribution in AdaBoost
- Weights in AdaBoost have no counterpart in WM

Connection to WM was part of original motivation

An iteration of AdaBoost

Have $X^{(t)}, x^{(t)}, w^{\text{old}}, d^{\text{old}}$

Want to find weight w for $x^{(t)}$ and distribution d^{new} to give weak learner

Write y^{old} for the margin vector $X^{(t)} w^{\text{old}}$

Then

$$w^{\text{new}} \leftarrow (w^{\text{old}}, w)$$

$$y^{\text{new}} \leftarrow y^{\text{old}} + w x^{(t)}$$

Iteration, cont'd

w is solution of

$$\arg\min_w \ln\left(\sum_i \exp(-y_i^{\text{old}} - wx_i^{(t)})\right)$$

d^{new} is solution of dual program

$$\arg\min_d y^{\text{old}} \cdot d + \sum_i d_i \ln d_i - 1$$

$$\text{subject to } \sum_i d_i = 1, d^\top x^{(t)} = 0$$

Note: $\ln\left(\sum_i \exp(x_i)\right)$ is dual to $\sum_i x_i \ln x_i + \delta(x| \sum_i x_i = 1) - 1$

Iteration, cont'd

KKT conditions are:

$$\begin{aligned}\sum_i d_i &= 1 \\ d^\top x^{(t)} &= 0 \\ y^{\text{old}} + wx^{(t)} + \lambda + \ln d &= 0\end{aligned}$$

Last equation gives

$$\begin{aligned}d_i^{\text{new}} &\propto \exp(-y_i^{\text{old}} - wx_i^{(t)}) \\ &\propto d_i^{\text{old}} \exp(-wx_i^{(t)})\end{aligned}$$

Iteration, cont'd

Partition $d^\top x^{(t)}$ according to whether weak learner was right or wrong

Now $d^\top x^{(t)} = 0$ gives

$$\exp(-w) \sum_{i | x_i^{(t)} = +1} d_i^{\mathrm{old}} = \exp(w) \sum_{i | x_i^{(t)} = -1} d_i^{\mathrm{old}}$$

Sums are probabilities under d^{old} that weak learner was right or wrong

So $w = \frac{1}{2} \ln \frac{1+\gamma^{(t)}}{1-\gamma^{(t)}}$

Is weak learning reasonable?

If weak learner can always succeed, then $\exists$ weighted vote that gets every training example right

If weak learning is wrong, AdaBoost seems like a good algorithm for finding a counterexample

In practice, this appears not to be a problem