Statistical Techniques in Robotics (16-831, F08) Lecture #24 (11/13/2008)

Gradient Boosting

Lecturer: Drew Bagnell *Scribe: Sergio Valcarcel*

1 Review Online Kernel Machine

1. Start
$$f = 0 \tag{1}$$
2. Receive
$$x_t \tag{2}$$
3. Make prediction using
$$f_t(x_t) \tag{3}$$
4. Receive
$$C_t(f_t(x_t)) \tag{4}$$
5. Update
$$f(x_{t+1}) \Leftarrow f_t - \eta_t \; i(f(x_t)k(x_{tj}) \tag{5}$$
6. Optionally shrink all weights

$$Regret = \sum_t (C_t(f_t(x_t)) - C_t(f^*(x_t))) \;\; | \; f^* \in H_k \tag{6}$$

Bound of the functional gradient time:

$$Regret \leq \| \; f^* \; \| \cdot \| \; \nabla C_t(f) \; \|_k \; \sqrt{T} \tag{7}$$

$\| \; f^* \; \|$ is the size of the function. How big can $\| \; \nabla C_t(f) \; \|_k$ be related to step 5? The answer is: $\alpha^T K \alpha$ which reminds to Support Vector Machines: $i^T()K(x,x)i()$;

About the size of the kernel if we take a Gaussian kernel we must normalize so the maximum equals one. So:

$$Regret \leq \| \; f^* \sqrt{T} \tag{8}$$

And that is cool!

We compute with any function in the Hilbert space with the Gaussian kernel. Any function you can write like this:

$$f = \sum \alpha_i K(i,j) \tag{9}$$

Being K the Kernel Matrix of $\alpha^T K \alpha$.

There is a trade off:

- the bigger the kernel width, the smoother but also the larger computations
- the smaller the kernel width, the more regret we will pay because all points will be constrained.

`picture`

2 Gradient Boosting

What happens if we hold the area but we do it narrower and narrower ($l \to 0$)?

`picture`

$$\int \zeta(x)\zeta(x-x')h(x) = h(x') \tag{10}$$

How to control the complexity? Compute this functional gradient and them picking some experts which lead to higher inner product.

`picture`

We are going to take something highly correlated with our functional gradient, but simple!

2.1 Algorithm

The algorithm is:

$$f_{t+1} \leftarrow f_t + \eta_t A[\nabla C(f_t)] \tag{11}$$

1. Try to find functional gradient
2. Try to find something simple and highly correlated
3. Apply

Note: If we think this as a cost, we should use $f_t - \eta_t A$.

How to do that? What is the key?

$$A[\nabla C_t] = \arg\max_{h \in H} < h(x), \nabla(C_t) > \tag{12}$$

What does the inner product of our hypothesis look like?

$$\int \gamma x h(x) \nabla C_t(x) = \sum_i h(x_i)\alpha_i \tag{13}$$

Where α_i is the multiplier on the delta function.

The simplest case is when the functional gradient has only values $h \in (-1, 1)$, in this case $(-\delta, +\delta)$.

Note: If we call $\alpha_1 \Leftarrow w_i$ then we are dealing with a Classification problem. Also, if $\alpha_i \in [0, 1]$ it is a weighted classification.

How many times does h agree with the sign?
If we think now in h as the simplest case, then the Linear regression values will be huge!

Linear SVM leads to threshold output which is not linear by combining K branch $h(x_i)\alpha$ on original parameters.

2.2 Gradient Boost Functional Descent

Applying **decision trees**...

2.2.1 Algorithm

1. Start:
$$f = 0 \tag{14}$$
2. Compute $f_t(x_i)$ for all data points i and create a training set which looks like:
$$x_i, sign(i(f_t(x_i)), |i(f_t(x_i))| \tag{15}$$
3. Train the algorithm with the training set and get:
$$h_{t+1} \tag{16}$$
4. Update:
$$f_{t+1} = f_t - \eta_t h_{t+1} \tag{17}$$

   Notes:

   - f_t has an output real value
   - η_t is some scalar
   - h_{t+1} is the output of that value

2.3 Special cases

If $C_t(f) = e^{-y_t \cdot f(x_t)}$ then

1. $A\gamma a$ Boost, very popular vision. η_t controls the step in the direction of the sign.
2. $\epsilon - A\gamma a$ Boost. η_t is a small constant. Problem of overfitting.
3. Rank Boost:
   $\max(0, f(x_i ncorrect) - f(x_c orrect))$
   The incorrect thing is going higher than the correct thing.
   How does the functional gradient look like for this functions? It should look like:

   `picture`

   The result of the max function is:

   - 0 when correct
   - And for the things we score incorrectly we have a decision tree code: $x_{incorrect}, +1, -1$ / $x_{correct}, +1, -1$

We can do regression with SVM classifiers applying the same rule.