Learning with control

Balázs Kégl

LAL, CNRS/Université Paris Sud

AppStat team meeting, January 13, 2010

Setup

- Set of nodes $\mathcal{J} = \{1,\ldots,M\}$

- multi-class discriminant function $\mathbf{g}_j$ at each node

- controler at each node $a_j : \mathcal{J} \rightarrow \mathcal{J}^*$

- full discriminant function

$$\mathbf{f}_j(\cdot) = \begin{cases} \mathbf{0} & \text{if } j = j_{\text{terminal}}, \\ \mathbf{g}_j(\cdot) + \sum\limits_{j' \in a_j} \mathbf{f}_{j'}(\cdot) & \text{otherwise,} \end{cases}$$

Alternating decision trees

- node classifiers $\mathbf{h}(\mathbf{x}) = \mathbf{v}\varphi(\mathbf{x})$

- controler:

$$a_j(\mathbf{x}, \varphi_j^{(1)}, \ldots, \varphi_j^{(T_j)}) = \left\{ c_{j\mathrm{sign}\left(\varphi_j^{(1)}(\mathbf{x})\right)}, \ldots, c_{j\mathrm{sign}\left(\varphi_j^{(T_j)}(\mathbf{x})\right)} \right\}.$$

- controler tightly coupled with classifiers
  - formal boosting algorithm
  - very complex function
  - idea: decouple the controler and the classifier

Cascades

- binary node classifiers $\mathbf{g}(\mathbf{x}) = g(\mathbf{x}) : \mathbb{R}^d \to \{-1, 1\}$

- controler:

$$a_j\big(g(\mathbf{x})\big) = \begin{cases} \{j+1\} & \text{if } g(\mathbf{x}) = +1, \\ \{j_{\text{terminal}}\} & \text{otherwise.} \end{cases}$$

- no known boosting algorithm to learn this structure

Classification-controled DAGs

- Extended cascades

- Controler gets only the output of $\mathbf{g}(\mathbf{x})$ as input

Attentional boosting

- Inspired by Larochelle NIPS'10
  - Base classifiers are grouped together by a natural clustering in their parameter space
  - Each subset can look at a subset of the features
  - Each node in the decision process is assigned to one of the groups
  - Controler gets everything in the node plus the parameter of the node

Multi-instance boosting

- Bags of input are classified positively if at least one of the elements is positive, negatively otherwise

- Controler should formalize the process of “looking for” the object

- special case of attentional boosting in the sense that only at the end when the object is found should we make a classification

Sparse boosting

- Each node is one base classifier, coming from a pool
- Goal is to reach a decision by looking at a fixed (usually small) number of classifiers by navigating in the pool
- If used together with autoassociative boosting, it could be used to build a deep booster: for the next level, represent the input as a sparse binary vector, and learn it at the next level
- Already tried something like that, that's what ParasiteLearner is about in multiboost

Ideas for learning

- alternate between boosting and controling
- given the nodes, controler could be learned by an MDP
- given the controler, fix the data set in each node and boost
- e.g: sparse boosting is just one interation (learn a large pool then learn a controler)
- multi-instance: find the best element in each bag and iterate (of course we assume some kind of structure among bag elements, as in images)