

Tax Collection Optimization

Soheil

1 Main Ideas

The system optimizes the total tax collection in long-term, considering the connection between businesses and dependencies among their needs, resources and legal constraints. The data analysis and optimization is done in a constrained Markov decision processes (C-MDP) framework. The output of the the C-MDP is a customized policy in tax collecting, which was different than its predecessor that had broad-brush rules, therefore, improves both efficiency and adaptiveness of the collection. It also enhances fairness in the procedure.

2 Important Notes

- Although the sequential dependencies between various collections stages are automatically discovered, the system also accepts hard constraints, which are expressed as rules, for use in its optimization process.
- Existing systems based on manually constructed rules suffer from the shortcoming that the rules do not adapt to changes in the environment.
- An alternative to such a rigid manual rule system is a mathematical model that describes the complex collections process, applies analytics to estimate it, and then optimizes to arrive at a collections policy within the class of policies describable in the model. The MDP provides one such framework with a rich-enough vocabulary to describe the complexity involved in a collections process.
- In a previous work the states of the MDP were fixed a priori, and the transition probabilities among them were estimated prior to the optimization process. This is not enough for our set of diverse tax payer characteristics.
- The *states* of the MDP represent the stages in the collection process that respective taxpayers cases undergo. The *actions* are the collections actions taken by DTF for those cases, and the *rewards* are the dollars collected.
- We can't simply assume that we know the structure of our MDP in such a high dimensional feature space problem. Instead, we have to use RL, approximate dynamic programming, to seek for a near optimal solution,

given access only to the data generated from the MDP, not the MDP structure itself
(no idea what that means!)

- Several types of constraints are involved. The first category consists of resource constraints that are imposed because some collections actions consume physical resources. Other constraints might not be resource related; these include bounds on the number of certain types of actions that are considered reasonable in a given period. The second major category of constraints, called action constraints, are those that are imposed by business or legal considerations
- In previous works, either the data modeling has not been considered, or considered separately from the optimization process. In here, data analytics and optimization are tightly coupled and applied in an iterative fashion.
- The key point lies in the constrained version of Bellman equation. Each iteration in an update consists of two process; estimating R , maximizing over actions. The estimation is done using segmented linear regression. When this process is completed, the output (a set of action effectiveness coefficients per segment) is entered into the optimization process, which determines a policy that allocates actions to maximize R over the population while respecting various constraints.
- In addition to the modeling features, the engine uses a group of features called action constraints binary features specifying, for each action considered, whether the action is currently allowed on the case, according to the business and legal rules

3 Questions

1. How many states and actions does the MDP that they are using have? There is a drastically simplified version of the state chart which demonstrates the states that a taxpayer is at. However, in their analysis, they used segmentation based on 200 features to model the state of a taxpayer. In addition, in table 1, a list of actions alongside with their respective costs is shown.
2. What is the purpose of the constraints mentioned in the paper? In the constrained version of Bellmans equation, the maximization on the right side is obtained through constrained optimization rather than through strict maximization.