Boosted Ensemble Learning for Model Predictive Control with Reconfiguration of Modular Facilities

Yi Daia and Andrew Allmana\*

aUniversity of Michigan, 500 S. State Street, Ann Arbor, MI 48109 USA

\*Corresponding Author: allmanaa@umich.edu

Abstract

The presence of discrete decisions in mixed-integer model predictive control (MPC) renders the optimization problem more significantly difficult to solve than a traditional continuous MPC problem. A solution is the use of data-driven models which help decouple the integer decisions from optimization and enable solution of control problem online. However, when considering the dynamic reconfiguration problem of modular facilities, there is a trade-off between the prediction accuracy and the magnitude of error in the incorrect predictions. In this work, we present an approach to determine integer control decisions *a* *priori* to solving the MPC problem using the ensemble method that takes advantages of several member classifiers trained by simulation data. Results demonstrate that the ensemble method achieves a breakthrough in the trade-off between classification accuracy and the magnitude of error in the incorrect predictions. A setpoint tracking case study demonstrates the MPC with ensemble method generally chooses configurations that give quicker set point recovery than the other member classifiers.

**Keywords**: Ensemble learning, Machine learning, Network reconfiguration, Modular manufacturing, Model predictive control

* 1. Introduction

In recent years, it has become clear that worldwide issues such as pandemics, climate change, and political conflict highlight the need for sustainable infrastructure to produce critical chemicals such as fuels and fertilizers. A promising approach to further enabling sustainability is to implement modular production units which are small-scale, standardized units that can perform traditional or intensified chemical unit operations (Baldea et al., 2017; Shao and Zavala, 2020). When considering a numbered-up modular facility to achieve desired throughput, the opportunity exists to dynamically transition between modular configurations, i.e. switching from operating modules in parallel to in series, as dictated by intermittent economic conditions. Our recent work (Dai et al., 2023) demonstrates that operational conditions will often affect which configuration of the numbered-up modular system most effectively controls the system, and that switching module configurations can be an effective way of tracking large changes in set points or rejecting large disturbances.

Modern chemical process control of complex systems is often performed using model predictive control (MPC), whereby an optimization problem is repeatedly solved in a moving horizon fashion in order to determine control actions. Introducing the action of reconfiguration to the optimal control problem adds integer variables that correspond to the module connectivity, resulting in a mixed integer nonlinear program (MINLP) which is extremely challenging to solve online in a time scale relevant for control. Our previous work (Dai and Allman, 2023) built a data-driven model embedded MPC which creates mathematical models for classifying the optimal operation configuration, decoupling the integer configuration decision from optimization, and enabling solution of the mixed integer nonlinear control problem online. However, it indicates that there is a trade-off between the classification accuracy and the magnitude of error in the incorrect predictions, the latter of which can lead to severely degraded controller performance.

A promising way to address this issue is the ensemble method which aggregates multiple weighted models to obtain a combined model that captures the best behaviors of its constituent models. The success of ensembles has been explained from statistical, computational, and representational aspects (Dietterich, 2000), bias-variance decomposition (Kohavi and Wolpert, 1996), and strength correlation (Breiman, 2001). Ensembles can be built by dependent frameworks which take advantage of knowledge generated in previous iterations by base classifiers to guide the learning in the next iterations. The most well-known dependent ensemble method, boosting, is a general method for constructing a single composite strong classifier combined by weak classifiers to achieve improvement on the performance of weak learners. One of the most popular boosting algorithms is AdaBoost (Adaptive Boosting) whose main idea is to pay more attention to patterns that are harder to classify (Freund and Schapire, 1996). Researchers have applied AdaBoost in a wide range of applications, such as supervisory building control rules (May-Ostendorp, 2012), identifying key parameters and steps for semiconductor manufacturing (San et al., 2016), life prediction for LEDs (Lu et al., 2017), and recognizing the coupling faults of complex industrial process (Xu et al., 2021), to name a few examples.

This work provides an approach to pre-establish integer control decisions by implementing AdaBoost algorithms in MPC problem solving. The remainder of the paper is structured as follows. The following section presents the optimal model predictive control problem and case studied in this work. Then, we introduce the multi-class AdaBoost algorithms with additional information from the training data. Furthermore, we present the AdaBoost classifiers’ performance and analyze a set point tracking case study of the MPC with new classifiers. Finally, we conclude by summarizing this work and suggesting some potential ideas for future development.

* 1. Problem Statement and Case Study

In this work, we develop an online configuration switching approach according to the dynamic system condition in the MPC of numbered-up modular systems. In this control feedback loop, as the left part of Figure 1 shows, the current operational conditions, such as set point, measured disturbance, and state measurements, are imported into the classifiers trained offline by machine learning algorithms. After dealing with the control condition information, classifiers tell the MPC which configuration the system should use for next time step. Then the MPC takes action on the process system by manipulating process inputs (flow rates and heating rates), switching configurations if necessary, and all of which lead the system to a new state for the next loop.

For the case study analyzed here, we consider a benchmark three module reactor system consisting of three nonisothermal CSTR’s with models adapted from Liu et al., 2009. We assume that these modules can operate in any of the configurations specified in Figure 1 right. In particular, a parallel configuration (a), a configuration with mixing (b), a hybrid series-parallel configuration (c), and a series configuration (d) are considered. Additional details on the benchmark modular system are the same as we introduced in our recent work (Dai et al., 2023).

|  |  |
| --- | --- |
| A diagram of a device  Description automatically generated | Diagram  Description automatically generated |

*Figure 1. (Left) MPC/ML classifier feedback loop schematic; (Right) All configurations considered for the 3 module system.*

* 1. Multi-class Adaptive Boosting Algorithms

Boosting has been a very successful technique for solving the binary classification problem. However, the ways to extend AdaBoost from two-class to multi-class depend on the interpretation or view of the success of AdaBoost in binary classification, which still remains controversial. Here we choose a widely used multi-class AdaBoost algorithm (Hastie et al., 2009) which directly extends to the multi-class problem without reducing it to multiple two-class problems.

In order to better quantify the performance of a classifier prediction, we propose a the following score metric which is zero when the best configuration is chosen, and gets gradually larger if a configuration is chosen that gives worse performance.

|  |  |
| --- | --- |
|  | (1) |

Where represents the controller performance index (PI) using MPC based on the configuration chosen by the classifier, and represents the PI using the best performing configuration. When the , it indicates that the predicted configuration is the optimal configuration, and the corresponding =0. When , the classifier is inaccurate in choosing the best configuration and the . It is apparent that can be quite large in magnitude when the degradation in controller performance is large, which could happen if the controller is unable to stabilize the system or reach the desired set point in the given configuration.

To encourage the trained classifier to avoid inaccurate decisions with high values we modify the multi-class AdaBoost algorithm with the aforementioned . We call the resulting algorithm S-AdaBoost, which proceeds as shown in Algorithm 1:

**Algorithm 1.** *S-AdaBoost*

1. *Initialize the observation weights .*
2. *For*
3. *Fit a classifier to the training data using weights .*
5. *.*
6. **if****then**

*exit loop*

**end if**

1. *, for*
2. *Re-normalize*
3. *Output*

Where represents the number of all training data points, represents the number of member classifiers, represents the number of all possible classifications, represents the feature matrix of all training data points, represents the score and represents the target of the th data point, and represents the classification results of all training data. is the weight of output of weak classifiers to build *S-AdaBoost.* Note that the only difference between S-AdaBoost and multi-class AdaBoost is the existence of in 2(b), in which increases the weights of misclassified instances in terms of the magnitude of incorrectness from PI. Instead of only focusing on prediction accuracy, this modification helps the classifier avoid configurations that may cause severe degradation in controller performance and increases the tolerance for inaccurate predictions where there is little degradation in PI compared to the best configuration.

* 1. Results and Discussion

Our previous work (Dai et al., 2023) considered control of numbered-up modular systems with fixed configurations, and which demonstrated that operational conditions could affect selection of the optimal configuration of modules. Using the same benchmark modular three reactor system, in this work we collect data on the control performance of different modular configurations for various initial conditions of states, values of disturbances and set points of process outputs from a space-filling sampling of the parameter space. We run 8641 different control simulations considering 4 configurations to construct the training and test data set. The simulated ranges of all features are listed in Table 1. Following data collection, various classifier models are trained which fall within three major types: support vector machines (SVM), decision trees (DT), and k-nearest neighbors (KNN). Output of these classifiers are used to construct ensemble classifiers using both traditional AdaBoost and the S-AdaBoost algorithm proposed in Section 3. All classifiers are trained using LIBSVM v0.8.0, DecisionTree v0.12.3, and NearestNeighborModels v0.2.3 in the MLJ v0.19.5 package. Other details on packages and computing environments are the same as our recent work (Dai et al., 2023).

The performance of machine learning classifiers has been listed in Table 2. The classifier based on AdaBoost model is limited to KNN classifiers due to its highest prediction accuracy. After applying S-AdaBoost, the magnitude of error in the incorrect prediction significantly decreases without losing too much accuracy on predicting the optimal configuration, with the worst case value of an order of magnitude less than it is in any of the individual classifiers. This is due to the fact that the S-AdaBoost algorithm specifically takes the into account when choosing the configuration, thus decreasing the likelihood of severe degradation in performance from an incorrect choice in modular configuration and mitigating the risk of a poor control decision.

Table 1. Range of features that have been used for generating training data.

|  |  |
| --- | --- |
| Features | Range |
| Feed temperature (K) | 300 - 340 |
| Temperature set point in each reactor (K) | 388.7 – 438.7 |
| Initial mol fraction of B in each reactor | 0.11-0.26 |
| Output mol fraction of B set point | 0.01-0.25 |

After implementing MPC with S-AdaBoost classifier, for most cases, it performs as well as the most accurate individual classifier KNN. This is demonstrated in Figure 3, which shows a setpoint tracking case study with different MPCs, in which the classifier embedded MPC takes action of switching configuration from (a) to (b) in Figure 1 (right). The switched strategy gives the lowest number on PI and the quickest change to the new setpoint, demonstrating that MPC with KNN/S-AdaBoost classifier has the best control performance for some setpoint tracking cases. Note that the PI in Figures 3 and 4 are calculated from the real-time control simulations, while the PI used to generate are calculated by the MPC optimal solution which contains the control performance in a forward horizon at the starting point right after setpoint change.

Figure 4 shows the set point tracking case that is the worst prediction of KNN classifier, where KNN embedded MPC uses the series configuration (d), and S-AdaBoost one

Table 2. Machine learning classifiers performance comparison

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier Model | % cases selecting the best configuration | % cases when | Worst case value of |
| SVM | 81.7 | 1.48 | 12.10 |
| DT | 85.0 | 1.03 | 38.59 |
| KNN | 94.4 | 3.87 | 39.09 |
| AdaBoost | 94.4 | 2.78 | 39.09 |
| S-AdaBoost | 91.0 | 1.71 | 3.86 |

|  |  |
| --- | --- |
|  |  |
| *Figure 3. Mole fraction profile (left) and performance index (right) of the desired product B in the final output stream. “Switched” represents the profile generated by MPC with machine learning classifiers, including KNN and S-AdaBoost which have the same performance.* | |
|  |  |
| *Figure 4. Mole fraction profile (left) and performance index (right) of B in the final output stream for MPCs embedded with KNN and S-AdaBoost respectively.* | |

chooses the parallel configuration (a). Neither of them switches configuration during the simulated control process. Compared to KNN, MPC with S-AdaBoost guides the system to the new setpoint quicker with less oscillations because it is able to identify the best configuration. Although the classifier is trained on single iteration of an MPC optimization solve, the results demonstrate superior performance in a moving horizon, closed loop simulation as well. Overall, our results demonstrate that while S-AdaBoost sacrifices some accuracy, it successfully avoids choosing configurations that severely degrade controller performance.

* 1. Conclusion

In this work, an approach to determine integer control decisions *a priori* to solving the MPC problem using improved data-driven machine learning classifier algorithms was presented. A metric was proposed to demonstrate how similar the performance of the predicted configuration is to that of the best configuration. Based on that, a new AdaBoost algorithm was proposed to help guide decision-making on optimal configuration away from those that severely degrade performance. Finally, a set point tracking case study was analyzed which demonstrates the MPC with AdaBoost classifier has the best performance on decreasing the magnitude of error in the incorrect prediction and generally chooses configurations that give quicker set point recovery than the other classifiers with fewer cases of severely degraded performance. As future work, we aim to extend this framework to systems with larger numbers of modules, and modules of different functionality.

**References**

M. Baldea, T. F. Edgar, B. L. Stanley, A. A. Kiss (2017), Modular manufacturing processes: Status, challenges, and opportunities. AIChE Journal 63, 4262–4272.

L. Breiman (2001), Random forests, Mach. Learn., vol. 45, no. 1, pp. 5–32.

Y. Dai and A. Allman (2023), Foundations of process/product analytics and machine learning fopam 2023, university of california, davis ca, july 30-aug 3, 2023 learning the best configuration for controlling modular dynamic systems, Ann Arbor, vol. 1001, p. 48109.

Y. Dai, S. Fay, and A. Allman (2023), Analysis of model predictive control in numbered-up modular facilities, Digital Chemical Engineering, vol. 7, p. 100088.

T. G. Dietterich (2000), Ensemble methods in machine learning, Multiple Classifier Systems. Berlin, Germany: Springer-Verlag, pp. 1–15.

S. K. S. Fan, S. C. Lin, and P. F. Tsai, (2016), Wafer fault detection and key step identification for semiconductor manufacturing using principal component analysis, AdaBoost and decision tree. Journal of Industrial and Production Engineering, 33(3), 151-168.

Yoav Freund and Robert E. Schapire (1996), Experiments with a new boosting algorithm. icml. Vol. 96.

T. Hastie, S. Rosset, J. Zhu, and H. Zou (2009), Multi-class AdaBoost, Statistics and its Interface, vol. 2, no. 3, pp. 349–360.

R. Kohavi and D. H. Wolpert (1996), Bias plus variance decomposition for zero-one loss functions, in Proc. Int. Conf. Machine Learning, pp. 275–283.

J. Liu, D. Munoz de la Pena, P. D. Christofides (2009), Distributed model predictive control of nonlinear process systems. AIChE Journal 55, 1171–1184.

P. T. May-Ostendorp, G. P. Henze, B. Rajagopalan, and C. D. Corbin, (2013). Extraction of supervisory building control rules from model predictive control of windows in a mixed mode building. Journal of Building Performance Simulation, 6(3), 199-219

Y. Shao, V. M. Zavala (2020), Modularity measures: Concepts, computation, and applications to manufacturing systems. AIChE Journal 66, e16965.

Y. Xu, K. Cong, Q. Zhu, and Y. He, (2021), A novel AdaBoost ensemble model based on the reconstruction of local tangent space alignment and its application to multiple faults recognition. Journal of Process Control, 104, 158-167.