CHAPTER I

INTRODUCTION

Classification or supervised learning is the process in identifying the class or group to which a data belongs, based on the basis of a training set of data whose classes are given. For example, in the task where we want to identify the type of creatures from their DNA sequence, each class is a possible type of creatures while the information to be used for prediction is the DNA sequence. The most logical way is to use our knowledge about the DNA sequence to determine to which kind of animals this DNA sequence belongs. However, the task of machine learning is not only to apply the knowledge that we have learned, but also to extract these knowledge from the data as well. The algorithm for classification will base its decision upon the dataset of DNA sequences with given classes of the creatures, interpret these information in some way and provide an output as a process of decision making called classifier.

The problem we are interested in is called customization, or in some research work is called user-adaptation. A simple description of customization is that there will be additional information about the problem in the form that we will be given an existing machine learning model that has already been trained to be suitable to the dataset in question up to some degree. For general usage, this model will be suitable to a dataset generated from a distribution with high similarity to the distribution of the dataset we are interested in. The main problem of customization is how we should use this model to boost or improve the result in the task of machine learning, and generally, the task would be inversely viewed as using the customization data to improve the model instead of using the model to improve the result from just using the customization data on the task. An obvious and practical real-world example of the problem in customization is Handwriting Recognition.

The situation in handwriting recognition is that we want to predict characters a person has written, with the information about how the writing tool has been stroked. An obvious characteristic of this problem is that every per-

son has different handwriting and there might not be any classifier that can correctly classify handwriting of everyone due to different writing habit. An obvious example would be for the characters with high similarity such as letter "O" and number "0" or letter "S" and number "5"; different people may write them with the same kind of strokes and hence given a data about the strokes used in writing a character, the result from classification of the character will depend on the writer as well. As a result, we will have to use handwriting data of each person to make a perfect handwriting classifier for himself. As in other tasks of machine learning, the higher the number of data, the higher probability that the resulted model will be good. Since handwriting of each person differs from each other, each of the input data must be generated by that specific person. Suppose that we want to have 10 samples of each character from a set of numbers 0-9 and alphabets A-Z. The total number of characters that the person in question must write will be $(10 + 26) * 10 = 360$, and that just might not be enough to create a good classifier for characters with similar writing pattern. The reason that handwriting recognition is a perfect task for customization is also because even though handwriting of everyone may be different from each other, but the same character of each person must be based on the same standard that it should be able for a man to read and recognize the character using his prior knowledge in the language. By using the language standard, we can create a classifier that fully employs additional information to perform customization.

Another advantage of customization is time efficiency. As the model does not have to be learned from scratch, performing customization is more efficient in time and it does not require the data previously used in making the model. One may view customization as incremental training which is performed by a new dataset at each time, where the model is adapted to the dataset while retaining the previous information up to some degree. Hence, the approach of customization will be suitable for the dataset of which characteristics can change overtime.

There are many algorithms for customization, but most of them have a common limitation that they usually make use of inner parameters of the model,

and thus they are model-specific (Fu et al., 2000; Lyu et al., 1995; Cao and Balakrishnan, 2005). Knowing the parameters of the model serves as additional information which would likely lead to better performance of the algorithm, but it also acts as additional constraint as well. If every algorithm for customization is model-specific, it will be obvious that we will need to have a customization algorithm for each type of models. This surely will be problematic when the type of the model cannot be identified correctly or when there is no customization algorithm for the model, which might likely be the case for any new kind of model. Therefore, we see the benefits of studying on how to perform customization in the way that can be applied to many types of models.

Though the concept of customization can be used on many tasks of machine learning, but this work will focus mainly on classification, due to the obvious application mentioned above. Also, we will base our interest only on the algorithm that can be applied to many types of models.

1.1 Objectives

1. To introduce the approach for performing customization which is not specific to the type of the model.
2. To give the guideline and some examples in how to perform customization that is not specific to the type of the model.

1.2 Scope

1. We will propose a framework and algorithms with the constraints that can be applied to many types of models.
2. We will focus on the task of classification and dimensionality reduction as a preprocess of the classification.
3. We will perform numerical experiments to evaluate the performance of the proposed method.

1.3 Procedure

1. Propose the framework for the task or subtask.
2. Propose an algorithm for the framework.
3. Conduct experiments using the proposed algorithm.

1.4 Contributions

1. We introduce the approach in performing customization which is not specific to the type of the model and also propose some frameworks and algorithms that belong to this approach.
2. Our work could lead to more development on the algorithms of the same kind.

1.5 Organization of the Thesis

We will make the settings of the problem more precise by providing some background knowledges in Chapter 2. In Chapter 3, we will introduce the concept of task-based customization and compare its advantages and drawbacks to the result from model specific algorithms. In Chapter 4 and Chapter 5, we will introduce some frameworks that are based on the concept of task-based customization on classification and dimensionality reduction and also give examples of algorithms that follows the approach, along with numerical results. Conclusion and future work are given in Chapter 6.