Automatic Speech Recognition
Lecture Note 4: Pattern Classification

1. **Introduction**

   A pattern classification system is used to classify data entries into classes. It often starts with a signal processing module to transform the raw data into a representation that is better suited for classification purposes. Knowledge of classes is gathered via training samples. Such knowledge is then incorporated into classification rules to classify new data. Based on how the training and classification is realized, one can distinguish between the following contrast terms.

   - **Pattern recognition vs. pattern classification**

     Classification simply refers to classifying static patterns while recognition refers to recognizing sequences of patterns, often without knowing the exact boundary between adjacent classes.

   - **Supervised vs. unsupervised trainings**

     In supervised training, each training sample is associated with a known label. In unsupervised training, such labels are not given. When such labels are needed, they are inferred from training data. This labeling may be sensitive to initial conditions and may change as training goes on. A commonly used method is clustering.

   - **Statistical vs. deterministic approaches**

     A statistical system uses statistical models for the classes and the samples. The training scheme basically estimates the model parameters from the training data. A deterministic approach does not implement any statistical models in the classification task. Though there are parameters in the system, they are not of any probabilistic nature.

2. **Feature Extraction**

   The first step in pattern classification is to decide how to represent the data of interest. The quantities and/or attributes used to represent data entries are called *features.* The goal of feature design and extraction is to reduce the variability between samples from the same class and increase the variability between samples from different classes.

Furthermore, concise representation of data saves storage and computational costs. Ideally, an ideal feature extraction renders classification trivial, since all classes are well-separated.

The number of features is an important parameter of the system. It should be large enough to discriminate training examples, and small enough to be generalized to unseen test examples (to avoid over-training). When the number of features is undesirably large, the principle component analysis (PCA) and the linear discriminant analysis (LDA) are often used to reduce the dimension of the feature space. PCA projects feature vectors into a subspace of fixed dimension with the variance in the samples accounted for maximized. LDA applies a linear transform of the input features and maximize the variance between class and minimize the variance within class.

3. **Classification Methods**

After the set of features is determined, all data can be represented as a vector in a space spanned by axes corresponding to these features. In this feature space the training and classification tasks are performed. The following are some commonly used schemes in classification systems.

- **Minimum Distance Classifiers**

  Given labeled samples, one of the simplest ways to classify a new sample is to compute the distances between this sample and the labeled samples, and classify it to be in the same class as the nearest neighbor. This method requires an appropriate distance measure. Storage issues may occur as the number of samples increases though this can be alleviated by standard methods.

- **Discriminant Functions**

  To discriminate a class from the others, one can design a function that yields large values for samples from this class and small values for samples from the other classes. Such a function is called discriminant function. Suppose there are k different classes. In the training stage, there are k discriminant functions to be determined from samples. During classification, a test data point is classified to the class whose discriminant function yields the maximum value at that point.

- **Artificial Neural Networks**

  Artificial neural networks (ANNs) is a based on the threshold logic unit (TLU). It consists of the input layer, the hidden layers and the output layer. Except for the input layer, at each node of the neural network, a non-linear function of the weighted sum of the input links is computed as the output. The most-frequently used non-linear function is the sigmoid function

  $$f(y) = \frac{1}{1 + e^{-y}},$$

  which imitates a step function (of the original TLU) but is differentiable.

  The parameters in the neural networks are the weights of the links and can be learned (updated as new samples coming in) via an error function defined at the output layer. Basically, these weights are updated to reduce the error criterion by going "downhill" in the parameter space,

  $$\mathbf{w}(t+1) = \mathbf{w}(t) - \alpha \nabla E(\mathbf{w}),$$

  where α is called the learning rate.