COS 424: Interacting with Data

Lecturer: Léon Bottou — Lecture #4
Scribe: Huy Nguyen — 2/23/2010

1 Machine learning mix and match

Building a machine learning system involves mixing and matching many different components. Slide 2 presents a comprehensive list of different components depending on different aspects of the problems. We will revisit this slides many times in the future. Firstly, there are many possible goals we might want to achieve. Last time, our goal was regression: fitting the data with some curves. The second aspect is representation. We looked at parametric models last time. There are also nonparametric models when we do not want to tune the parameters. The models we looked at were also nonprobabilistic as opposed to probabilistic ones because the models were just functions of the data without any probabilistic interpretation. The models can also be linear or nonlinear, with deep or shallow representation. The next aspect is capacity control. As we have talked about in the past, in order to use better models, we have to pay with either more data or better data. Regularization is another thing we already talked about when we add a quadratic term to the objective function in least square fitting. There are also operational considerations associated with the task we want to solve. So far we have only looked at offline problems where we have all the training data available at the beginning and the model can train on all of them at once. In online problems, the data come one at a time and the model has to adjust to the data over time. Lastly, we have computational considerations. For small datasets, we can use exact algorithms. For big datasets, we want to use stochastic algorithms. For even bigger datasets, we use parallel algorithms. Today, we will look at classification tasks with both parametric and nonparametric models. We will also talk about various loss functions.

2 Classification

Classification, a.k.a. pattern recognition, clustering, is the task of associating a pattern $x \in \mathcal{X}$ and a class $y \in \mathcal{Y}$. Slide 5 gives a list of examples of classification tasks. The tasks can be binary classification, multiclass classification, multilabel classification, or sequence recognition. We will talk about binary, multiclass, and multilabel classification today. We will talk more about sequence recognition later when we discuss hidden Markov models.

3 Probabilistic model

Let X and Y the the random variables representing the patterns and the classes. There are two approaches to model the generation of a pattern of a particular class. Slide 6 presents two diagrams of these two approaches. The first approach is that there is a pattern generator and a class labeler. The pattern generator first generates a pattern, such as a picture of a person. Then the labeler generates a class, such as the gender of the person in the picture, based on the information from the picture. This approach is summarized as $P(X,Y) = P(X) \cdot P(Y|X)$. The second approach is as follows. First, a class generator generates a class, e.g. gender of a person. We also have many pattern generators, one per class. Once a class is generated, we select the output of the corresponding generator and output the pattern. This approach is summarized as $P(X, Y) = P(Y) \cdot P(X|Y)$.

4 Bayes decision theory

Next we look at Bayes decision theory. Slide 7 gives a derivation of Bayes optimal decision rule and error rate. Note that in the derivation, we use $\mathbb{I}(E)$, the indicator function of an event E: it is 1 when E happens and 0 otherwise. Bayes decision rule is the best possible classifier since for each x, we choose the best possible class for it that minimize the error probability. A visualization of this fact is described in slide 8, where we look at the diagram of the joint density of patterns and classes.

However, the theory assumes we know the probability distribution while in practice, we only have a finite amount of data. What we can do is either approximating the optimal Bayes classifer, picking a function from a parameterized family that minimizes the empirical error, or determining the class of x based on the classes of nearest neighbors of x.

5 Nearest neighbors

Let $d(x, x')$ be a distance on patterns. For a pattern x, we can assign to x the class of the closest training example (1NN). Alternatively, we can look at the classes of the k nearest neighbors and let them vote (kNN). There are also variants with weighted votes smoothed by the distance.

When the patterns are on a plane, we can draw the bisectors between pairs of points to divide the space into cells so that all patterns within each cell are assigned the same class as the example in that cell. A similar process can be done in any space other than the plane. This is called Voronoi tesselation.

Interestingly there is a theorem by Cover and Hart that says the error of 1NN is bounded by twice the Bayes optimal error rate. Slide 12 has the statement and the proof of this theorem.

Let $\eta(x) = \mathbb{P}\{Y = +1|X = x\}$. As we increases the number of neighbors, the error rate approaches the optimal curve, provided that $\eta(x_{knn(x)}) \approx \eta(x)$. If there is not enough data, the kth nearest neighbor of x can be far from x and this condition cannot hold. Therefore, in order to increase k, we need more data. Thus, k is a capacity parameter.

There are various ways to improve the implementations of kNN, with data structures and pruning techniques for finding nearest neighbors. Slide 14 describes some of them.

The choice of distance measure also affects the algorithm. When using Euclidean distance, it is often unnecessary to compute the square root. However, when using the triangle inequality to prune the search space, the square root is required. A variant of Euclidean distance is Mahalanobis distance, which is scaled by the inverse covariance matrix:

$$d(x, x') = (x - x')^T A(x - x')$$

where $A = \Sigma^{-1}$ or $A = (\Sigma + \epsilon I)^{-1}$ since Σ is not necessarily invertible.

Another technique is dimensionality reduction. We project the patterns to a subspace that retains most of the variance. The details are described in slide 15.

6 Discriminant function

Next we discuss the discriminant functions, which is used in binary classification. Given a function f_w, pattern x is assigned the class $sign(f_w(x))$. The function works as follows. First from x, we compute $\Phi(x)$, the basis expansion of x, or a set of features of x. Then some weight w is applied to it. For example, a linear discriminant function has the form $f_w(x) = w^T\Phi(x)$. The perceptron algorithm is an example of linear discriminant functions. The algorithm is described in details in slide 18. It can also be understood as a stochastic gradient applied to a particular loss function (the perceptron loss, $\ell(z) = \max(0, -z)$). This interpretation is described in slide 18. The gain parameter γ is used to speed up convergence.

7 Minimizing the empirical error rate

Next we discuss the optimization problem of choosing w so that the empirical error rate is minimized.

$$\min_w \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{y_i f(x_i, w) \le 0\}$$

The trouble with using the misclassification loss function is that it is noncontinuous, nondifferentiable, and nonconvex, which makes optimization very hard. Thus, we try to approximate this function with some nicer surrogate loss function ℓ. For example, consider the quadratic loss function $\ell(z) = (z-1)^2$. This function approximates well the loss function around the threshold but it has a drawback that it also assigns big losses to large correct assignments. Some other functions considered are exponential loss $\ell(z) = exp(-z)$, log loss $\ell(z) = \log(1 + exp(-z))$, perceptron loss $\ell(z) = max(0, -z)$ and hinge $\ell(z) = max(0, 1-z)$. The exponential loss is mentioned mostly for historical reason since the log loss function approximates the misclassification loss function better. Other loss functions people use include quadratic+sigmoid loss, and ramp loss. The choice of the surrogate loss function is dependent on the constraints from the optimization algorithm. Slide 24 summarizes some of the constraints. We also want loss function to work in the case of ambiguity i.e. the minimum of the loss function should correspond to the class with higher probability. For example, with hinge loss, the minimum always corresponds to the correct class. However, with perceptron loss, whenever there is ambiguity, the minimum is always 0, which does not give any information about which class a pattern belongs to.

8 Asymmetric cost

We continue the discussion on binary classification on two classes ± 1, now with asymmetric cost. For example. a heart patient does not want false negative in pacemaker, and is willing to tolerate for false positive. However, the producer of the pacemaker would have a different compromise between these two rates. Suppose that we have a threshold function classifier. In this case, the approach is to look at the Receiver Operating Curve (ROC), and choose the appropriate threshold. Again, we use decision theory to see what the optimal decision rule looks like. Slide 27 gives a derivation of the optimal asymmetric decision rule. The nice thing about the optimal decision rule, $f(x)$, is that it is a threshold function. Also notice that $\eta(x)$ is the optimal ROC curve. In practice, we only observe the empirical ROC curve, and we want to choose the weight so that it is as close to the optimal curve as possible. One possible way is to maximize area under curve (AUC). We want $min \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \mathbb{I}\{f(x_i, w) \leq f(x_j, w)\}$, which is noncontinuous, nondifferentiable, and nonconvex. Therefore, we use a surrogate loss function $min \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \ell(f(x_i, w) - f(x_j, w))$. In practice, AUC might optimize the area in some direction we do not care about. There are many other algorithms for addressing this problem.

9 Multiclass

When there are more than two classes, there are two common approaches: one versus all, and one versus others. Slide 30 gives a summary of these two approaches. It is also possible to perform classification into multiple classes directly instead of using many binary classifiers. We can formulate the optimization problem with cost function similar to the binary case. Slide 31 summarizes this approach. Similar to the binary case, if there are two classes with the same score, the perceptron loss has the problem with 0 again. The fix is to use hinge loss, or log loss. This approach is more expensive than one versus all but usually not better than one versus all in practice.

10 Multilabel

We can also have classification problem where each pattern can be assigned multiple labels. We can use one binary classifier for each label but this ignores the dependency between labels. When we classify documents by topics, if a document belongs to a subtopic, it must also belong to the topic containing that subtopic. However, when they are decided by different classifiers, it is not always the case. A more complex way is as follows. Give a score $f_k(x)$ to a document x and a topic k. $R_w(y)$ measures the compatibility of the topic set y. The set of topics assigned to x is $argmax_{y_1 \ldots y_k} R_w(\{y_1 \ldots y_k\}) + \sum_k f_k(x)$. We can again use the same loss functions as in the case of multiclass classification.