OPTIMAL CHOICE: NEW MACHINE LEARNING PROBLEM AND ITS SOLUTION

Marina Sapir

Meta Pattern, USA

ABSTRACT

We introduce the task of learning to pick a single preferred example out a finite set of examples, an "optimal choice problem", as a supervised machine learning problem with complex input. Problems of optimal choice emerge often in various practical applications. We formalize the problem, show that it does not satisfy the assumptions of statistical learning theory, yet it can be solved efficiently in some cases. We propose two approaches to solve the problem. Both of them reach good solutions on real life data from a signal processing application.

1. INTRODUCTION

Machine learning is described as 'automated detection of meaningful patterns in data' [1]. No limitations on types of data or patterns are set explicitly. Yet, classical supervised machine learning includes, mostly, three types of problems: regression, classification and ranking which all have the relational form:

1. Given is a finite set of pairs $\{\langle x_i, y_i \rangle\}$, where observations $x_i \in R_n$, and y_i are some labels,

2. The goal is to find a function which approximates dependence of labels y on vectors x.

The problems differ by types of labels and measures of success but not by their general structure. The concepts "learning models", 'PAC learnable" in statistical learning theory are developed for the relational data [1, 2].

Recently, the concept of "structural learning" is being developed to include more data types. There are several specific applied problems being considered (image and string segmentation, string labeling and alignments) and domain specific algorithms are developed. Typically, these problems are defined on data which form nested structures and/or produce outputs as some nested or easily decomposable structures. Sometimes, the labels are represented by sequences of differ ent, arbitrary lengths [3, 4]. The work [5] generalizes problems with structural output as class of machine learning problems where output can be decomposed into fixed number of substructures. The work [6] proposes generalization of machine vision problems adding relationship of inclusion of the inputs.

However, some real life problems do not fit even in these more general definitions. Consider, for example, the task of predicting an outcome of a horse race.

Given is the history of horse races, including the information about the participating horses and the winner of each race. Each competing horse is characterized by certain features. The goal is, knowing the features of the horses running in the next race, predict the winner.

This is a supervised machine learning problem, since it requires to automatically learn prediction of the winning horse. Even though the winner is the horse, the horse can not be classified as a "winner" regardless of the race. The main issue here is that victory depends not only on qualities of the winner, but on qualities of other competing horses in the same race. A horse may be sure

winner in one race and sure loser in another race, where all the horses are "better" in some important ways. As races are different, so are the winners. The data reflect relationship not between horses and the victories, but between horses and races on one hand, and the victories on another. Since a race may have arbitrary number of running horses, the description of the race can not be presented as a single vector of fixed length.

Here, we will study the general problem which requires to learn how to pick the "best" example out of finite sets of examples. We call it an Optimal Choice (OC) problem, of which horse races is one example. We will start by giving the formal definition. Then we bring examples of important and popular applications where such problems appear. We show that the main assumptions of statistical learning theory are violated for OC. Next, we explore two general approaches to solve the problem and test them on a real life data from a practical signal processing application.

1.1 OPTIMAL CHOICE PROBLEM, FORMAL STATEMENT

Let us introduce the general concepts and the definition of the problem.
A choice is vector of values of n features, $x \subset \Re n$. A lot is a finite set of choices. Denote D set of possible choices, and   set of all possible lots.

Each lot has not more than one choice identified as prime. Conveniently denote X' the prime in the lot X. If the lot X does not have a prime, it will be denoted as $X' = \emptyset$. Then, training set can be presented as a finite set of pairs

$$Z = \{\langle X_1, I(X_1', X_1)\rangle, ..., \langle X_m, I(X_m', X_m)\rangle\},$$

where all $X_i \in \Omega$ and $I(x, X)$ is identity function on X :

1. $I(x, X) \in \{0, 1\}$

2. $I(x, X) = 1 \Leftrightarrow x = X'$

3. if $X' = \emptyset, I(x, X) \equiv 0$.

For example, in horse races, choices are the horses, a lot is a race, and the prime is the winner of the race.

The goal is to build a labeling function $f(x, X) \rightarrow \{0, 1\}$ such that, for every $X \in \Omega, x \in X, f(x, X) = I(x, X)$.

If the condition $f(x, X) = I(x, X)$ holds for a labeling function f and lot X, we say that a lot X is a success of a labeling function f. Success rate of a labeling function is the probability of its successes on Ω.

Let us consider a numeric function $g(x, X)$ on pairs $X \in \Omega, x \in X$. The function g correspond to a single labeling function

$$f(x, X) = 1 \Leftrightarrow \arg \max_{x \in X} g(x, X) = x$$

if g(x, X) has a single maximum on X; otherwise, f(x, X) $\equiv \emptyset$ on X.

The function g(x,X) which identifies the primes by its maxima in the lots will be called scoring function. The success rate of the corresponding labeling function will be associated with the

scoring function.

1.2 PRACTICAL EXAMPLES OF THE OPTIMAL CHOICE PROBLEM

OC problem occurs in many practical applications, even though it was not formalized yet, to the best of author's knowledge.

1. COMPETITIONS

Not only horse races, but most of competitions, including sports, beauty contests, elections, tender awards and so on, give rice to the OC type problems. Every competition with a single winner has the same form of data, essentially. Even when the same participants compete in each round of the competition, they change with time and their features change as well.

2. FINDING CYCLE IN CONTINUOUS NOISY SIGNAL

The OC type problem was discovered in real life data analysis as part of the large signal processing research. The company Predictive Fleet Technology, for which the author did consulting, analyzes signal from piezoelectric sensors, installed in vehicles. The engine's 'signature' (recorded signal) continuously reflects changes in pressure in exhaust pipe and crankcase, which occur when engine works. The cylinders in an engine fire consequently, so it should be possible to identify intervals of work of each cylinder within the signature. The goal is to evaluate the regularity of the engine and identify possible issues.

The most important part of the signature interpretation is to find the cycle: interval of time, when every cylinder works once. The problem is difficult when the signature is irregular and curves of consecutive cycles do not look the same, and when the signature is very regular and all the cylinders look identical. Some preliminary work allowed us to find the intervals of potential cycles (choices) for each signature, and each choice is characterized by four "quality criteria" (features). There is a training set, where an expert identified true cycle for each signature.

The features are obtained by aggregating several signal characteristics: two features evaluate irregularity of each of the curves (from exhaust pipe and crankcase) would have if the given interval is selected as the cycle. And two other features characterize some measures of com plexity of the interval itself. All the features correlate negatively with the likelihood of the choice being the prime. Three features are continuous, and the fourth feature is binary. They are scaled from 0 to 1.

The Goal Is To Develop The Rule Which Identifies The True Cycle (The Prime) Among The Chosen Intervals For Each Signature. The Main Property Of This Problem Is That There Is No "Second Best": Only One Cycle Is Correct, The Rest Are Equally Wrong. This Problem Lands Naturally Into The General OC Problem.

All the solutions proposed here are applied on the dataset of this problem. The data contain 2453 choices in 114 lots, on average 21 choice per lot, no lot contains less than 2 choices, and every lot has a prime.

The data are available upon request.

3. IMAGE AND SIGNAL SEGMENTATION

The problem of finding a cycle is an example from a large class of problems of image and signal segmentation. Suppose, a preprocessing algorithm can identify variants of segmentation for a particular image. If there are some criteria which can characterize each segmentation, then a lot will consist of the the variants of the segmentation for the given image with the features given by

criteria values. If an expert marks a correct segmentation for each image, we are in the situation of OC problem again.

4. MULTIPLE CLASSES CLASSIFICATION WITH BINARY CLASSIFIERS

In many cases, the classification with $k > 2$ classes is done with a binary classifier. The task is split on classifications for each one class versus all others. But what if some instances do not fit nicely in any of the classes, or found similar to more than one class? What if one uses several classification methods which point to different classes? One still needs to find the "prime" class. In these types of problems, each lot will have k choices, and each choice will be characterized by some criteria of fit between the class and the instance. One needs to find an optimal rule which will aggregate these single class criteria into a rule for all the classes together. This is an OC problem.

5. RECOMMENDER SYSTEM

Suppose, a recommendation system presents a customer with sets of choices each time, and lets him to choose one option he likes the best. The choices may be movies, books, real estate, fashions and so on. The goal is to learn from the customer's past choices and recommend him new choices in the order of his preferences. Here, we are in situation of the OC problem again. The training set contains past lots, where each choice is characterized by several features, and the customer's choice (prime) is known. The system needs to develop a personalized scoring function on choices to present them to the customer in order of his preferences.

6. RATING SYSTEM

Let us consider a rating problem. There are two types of such problems, depending on the feedback. The feedback in the training set may be binary, or it may represent ranks. For example, the trainer marks each link as "relevant" to a query or not. Another option is to have the trainer to assign a rank (or rating) to each object in the training set. In both cases, the goal is to learn to rank the new objects.

Both approaches are hard on the trainer. If actual ratings have many values (various degrees of relevancy), it may be difficult to assign just two values correctly. Assigning multiple rating in the training set may be even more difficult. For a normal size training set, it would be too much work for a trainer to check all the comparisons his ratings imply. Besides, some of the objects he rates may not be comparable. It means, the trainer can not guaranty that all the relationships implied by his ranking are true. It leads to inevitable errors, contradictions in labels in training data.

The solution may be to ask the trainer to select groups of comparable objects (lots), and identify the best choice (prime) in each group. This will lead one to OC problem.

Finding robust solutions for any of these practical applications may be very valuable. Yet, it will require some new approaches.

1.3 OC AND STATISTICAL LEARNING THEORY

The problem has some similarity with two popular types of machine learning problems: classifi-cation and ranking.

As in binary classification, the goal in OC is to learn a rule, which can be applied on the new data to classify each choice in the lot as its prime or not.

To see similarity with the ranking problem, let us notice that selecting the prime in each lot establishes partial order on the set of choices of this lot: $X' \succ q \text{ for a } q \in X, q \neq X'.$ So, the OC problem can be considered as a problem of learning the partial order on the samples of the lot.

Despite the similarities, the problem can not be presented in relational form, and the main explicit and implicit assumptions of statistical learning theory do not hold for OC.

1. LABELS ARE NOT THE FUNCTION OF THE FEATURES

Statistical learning implicitly assumes that the values of the features determine the probability of label. Essentially, each classification method approximates a random function on $D \rightarrow R$ given on the training set.

In OC, the labels are not a function of the features alone, since they depend both on the choice and its lot. A choice may be the prime in one lot and not in another. For example, the prime horse in a small stable is expected to be a poor competitor in some famous derby. It means that close (or even identical) choices in different lots shall be assigned different labels by the learning algorithm to have satisfactory success rate.

2. THE FIT CAN NOT BE EVALUATED POINT-WISE

In classical machine learning, the fit between the true labels and assigned labels is estimated as a measure of success, accuracy. For example, in classification, the probability of the correct labels is evaluated. In ranking, some measure of the correlation (agreement in order) between the known ranks and predicted scores is estimated.

The next examples show that counting correct labels on choices or measuring correlation between the assigned and true labels in the training set can not be used to evaluate the learning success in OC.

Suppose, for example, there are 10 choices in every lot. From classification point of view, if the decision rule assigns zero to every choice, the rule is 90% correct. From optimal choice point of view, the success rate is 0, because it did not identify any of the primes.

If a scoring function scores a prime in every lot as the "second-best", many correlation measures used in supervised ranking will be rather high. In this case, on each lot, 8 out of every 9 not-prime choices are below the only prime choice, so AUC= 8/9 [7]. Yet, the scoring function fails to find the prime everywhere, and, accordingly, the success rate of this scoring function is 0.

3. INDEPENDENCE

In machine learning, both feature vectors and feedback are supposed to be taken indepen- dently from the same distribution. This is, obviously, not the case here.

As for labels, in each lot, only one label is 1, the rest are 0.

The distribution of the choice features is expected to depend on the lot. For example, more prestigious races will include better overall participants and have necessarily different from other races distribution of features.

However, we can expect that the lots appear independently, according with some probability distribution Pr(X) on $\Omega.$ Also, we can assume that there is probability distribution of a choice to be the prime in a given lot: $Q(x, X) = Pr(x = X').$ The lots and their primes in the training set are generated in accordance with these two distributions.

2. SOLVING THE PROBLEM

The statistical learning assumptions make classic machine learning problems tractable, allow some efficient solutions. We showed that the assumptions are not satisfied in OC type of problems. So, one may wonder, if OC has a decent solution.

In fact, identifying these issues helps us to find the ways the problem can be solved. We explore two paths to the solution here.

First, we consider the problem in an extended set of features, where the added features characterize lots. If the features are selected successfully, it can make the labels dependent on the choices only. Then, the point-wise fitting machine learning methods can be applied, provided that, in the end, the fit is still evaluated with the success rate.

As another possible solution, we explore general optimization methods which can optimize the OC success rate directly in original feature space.

We will show here on the example of the real life and difficult problem of finding cycle in continuous noisy signal that a good solution can be found efficiently both ways.

2.1 EXPANDING FEATURE SPACE TO APPLY POPULAR MACHINE LEARNING METHODS

As we mentioned above, the main issue with oc problem is that the labels depend both on the choice and its lot. to make the labels less dependent of lots, we add new features about lot as a whole. denote $\overline{D}$ the extended feature space. In $\overline{D}$ each choice still has its own specific features as well as new features, characterizing its lot, and common for every choice in its lot.

The goal of extending the feature space is to have similar in $\overline{D}$ choices across all lots to have identical labels with high likelihood.

Selection of the lot features, usually, requires some domain knowledge. However, there may be some empiric considerations which simplify the selection.

Let us consider a simple case, when there are features F which correlate with the likelihood of a choice to be prime and mutually correlate (if the features F are developed to predict primes, it is the case, usually). Denote m_X vector of maximal values of the features F in lot X, and suppose values of m_X are used as new features to characterize the lot. Then, a lot with higher values of m_X will, likely, have a prime with higher values of the features F. It is likely as well that lots with similar values of m_X will have similar primes. Or, at the very least, their primes will be less dissimilar, than primes of the lots with very different features m_X

In finding the cycle problem, two features, cc.d and CrankRegul, have very different distributions from one engine to another because the regularity of engines varies widely. The features have negative correlation with the likelihood of the choice to be prime. So we added two new features: min.cc.d, and min.CrankRegul, which equal minimal in the lot values of the features cc.d and CrankRegul respectively.

For good engines, the features correlate strongly. They both achieve minima on the intervals multiple of the true cycle. For bad engines, the features do not correlate. It means, for a bad engine, there may be potential cycles with low value of one feature and high value of another. Then, the bad engine has low values of both additional features, as do good engines. In this case, additional features do not help distinguishing the engines and predicting the prime. Fortunately, this does not happen often. The bad engines have, usually, higher values of these features than good engines.

With all the applied methods, the output was interpreted as a scoring function: primes were identified by maximal values of the function in each lot.

We used the R implementation of the most popular regression and classification methods: function SVM with linear kernel from e1071R package, the function neuralnet from the R package with the same name, function boosting from the R package adabag, function glm from R base to build logistic regression. In all the functions, we used predicted continuous output as a scoring function.

Testing was done with "leave one lot out" procedure: each lot, consequently, was removed from training and used for test. Percentages of the test lots, where the prime was correctly found by each method, are in the table 1.

Selection of parameters of the neural net is a challenge. We used 1 hidden layer with 3 and 4 neurons. Adding more layers or neurons does not help in our experiments. ADAboost strongly depends on the selected method's parameters as well. In our experiments, ADAboost builds 30

Table 1. Success Rate, Machine Learning Methods

Method	Orig. data	Extended data
SVM, linear kernel	0.80	0.86
ADAboost	0.74	0.83
Logistic regression	0.82	0.87
Neural net, 3 neurons	0.79	0.79
Neural net, 4 neurons	0.80	0.81

decision trees with the depth not more than 5. Changing these parameters did not improve the solution.

2.2 OPTIMIZATION OF THE SUCCESS RATE CRITERION

We were looking for a linear function of the original features which maximizes the success rate. The dataset at hand is relatively small, so we could use the most general (slow) optimization methods, which do not use derivatives.

2.2.1 USING STANDARD OPTIMIZATION METHODS

We applied the standard Nelder - Mead derivative-free optimization method (implemented in the function optim of the R package stats) to find the a linear function of the features optimizing the success rate criterion. Depending on the starting point, the success rate of the found functions on the whole dataset was from 0.82 to 0.86. Other optimization methods implemented in the same R function produced worse results. It is interesting to notice that the the method works amazingly fast, much faster than neural network or ADAboost on the same data.

2.2.2 BRUTE FORCE OPTIMIZATION

The data for the cycle problem are rather small, so we applied the exhaustive search to find true optimal hypothesis in a narrow class of hypotheses defined by the next rules:

1. The hypotheses are linear functions of the features $f(x) = a_1 \cdot x_1 + \ldots + a_4 \cdot x_4.$

2. All coefficients a1, . . . , a4 have integer values from 0 to n,

where n is a parameter of the algorithm. Out of all the possible rules with the same threshold n and identical (1% tolerance) performance, the algorithms picks the rule with minimal sum of coefficients.

For n = 15, the optimal linear function has the maximal value of coefficients equal 5. The rule has success rate 88%. The algorithm with the parameter n = 5 was applied in "leave one lot out" experiment with the same success rate, 88%.

3. CONCLUSION

Here we formulated the new machine learning problem, Optimal Choice, and proposed two paths to solve it. The problem emerges in various practical applications quite often, but it has undesirable properties from statistical learning theory point of view. Formalization of the problem and close look at its statistical properties helped to find ways to solve it.

The solutions were applied to a real life problem of signal processing with rather satisfactory results.

The proposed solutions have some shortcomings. The first approach is based on extending feature space, which would require some domain knowledge in each case. The direct optimization of the success rate criterion worked very well on relatively small data, but it may not be scalable.

The goal of this article will be achieved, if the OC problem attracts some attention and further research.

REFERENCES

[1] Shalev-Shwartz S, Ben-David S (2014) Understanding Machine Learning. Cambridge Uni- versity Press, NY.

[2] Hastie T, Tibshirani R, Friedman J (2009) Elements of statistical learning. Springer.

[3] Ricci E, Bie TD, Cristianini N (2008) Magic moments for structural output prediction. Jour- nal of Machine Learning Research 9: 2803 - 2846.

[4] Sohn K, Lee H, Yan X (2015) Learning structured output representation using deep condi- tional generative models. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R, editors, Advances in Neural Information Processing Systems 28, Curran Associates, Inc. pp. 3483–3491.

[5] Cortes C, Kuznetsov V, Mohri M, Yang S (2016) Structured prediction theory based on factor graph complexity. In: Advances in Neural Information Processing Systems 29: Annual Con- ference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. pp. 2514–2522.

[6] Nowozin S (2009) Learning with Structured Data: Applications to Computer Vision. Ph.D. thesis, Elektrotechnik und Informatik der Technischen Universitat Berlin. URL nowozin.net/sebastian/thesis/nowozin2009phdthesis.pdf.

[7]   Ertekin S, Rudin C (2011) On equivalence relationships between classification and ranking algorithms. Journal of Machine Learning Research 12: 2905 - 2929.