Satej Soman

CAPP30254: Machine Learning for Public Policy

Spring 2019

MIDTERM EXTENDED ASSIGNMENT

Contents

1 Section A: Short Answer **1**

2 Section B: Methods, Evaluation, and Communication **3**

2.1 Decision Trees . . . 3
2.2 Evaluation 1 . . . 5
2.3 Evaluation 2 . . . 6
2.4 Evaluation 3 . . . 6
2.5 Communicating Results . . . 6

3 Section C: Solving a New Problem **7**

1 Section A: Short Answer

1. **Logistic regression** is the appropriate tool in this case. The problem setup in logistic regression yields the probability that a point will fall into a given category ($P(Y = 1|X) = \Lambda(\beta^T X)$). In contrast, a classical hard-margin support-vector machine will simply be able to classify whether unemployment will rise or fall. It is possible to use distance from the calculated hyperplane as a probability measure, but the logistic regression technique is more appropriate.

2. Logistic regression requires binarizing data that take on binary labels (e.g. $\{\text{"False", "True"}\} \mapsto \{0, 1\}$), or creating dummy variables for categorical variables. Normalizing or rescaling data is not required for logistic regression, and will change the interpretation of the coefficients.

3. When $k = 1$, the error on the training set must be 0, since the prediction for each training data point will be the data point itself since no other neighbors are considered.

4. Leave-one-out cross-validation error (LOOCV)

   In the diagrams below, the circled green point is left out in training. A blue line connects the left-out point to its nearest neighbor ($k = 1$) and a set of red lines connect the validation points to its second- and third-nearest neighbors ($k = 3$).

   Assume that the distance function in training and testing is Euclidean distance and that the cross-validation error is mean square error of each validation set.

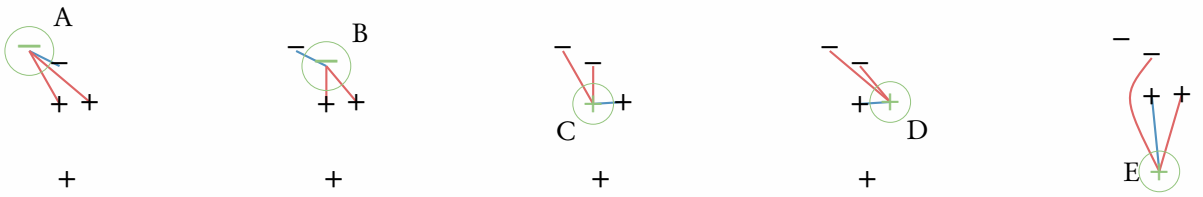


Figure 1: Nearest-neighbor calculations for all possible leave-one-out validation sets.

   A. When $k = 1$, no holdout points are misclassified, so the error is 0%.

   B. When $k = 3$, points A, B, C, and D are misclassified, so the error is 80%.

5. Either the SVM or decision tree is an appropriate choice here. A support-vector machine, with the correct kernel (likely polynomial or RBF), will be able to pick up on the horizontal pattern. Similarly a two-level decision tree would be able to divide the feature space into quadrants. The regression boundaries in this case would likely be muddled, meaning that logistic regression is not a good choice.

6. For a generic application with this dimensionality, I would prefer *k*-nearest neighbors over decision trees. Because the number of the relevant features is low, distance calculations will not be prohibitively computationally intensive, and the model does not need to be "built", since classifications are done based on the entirety of the dataset. Further, since all of the features are continuous, the decision tree approach requires careful decisions about where to discretize features. *In this example*, however, I would choose the **decision tree** approach because of how the model is likely to be deployed - instead of having doctors input patient characteristics and get back a classification based on the position of a point in 10-dimensional space, the decision tree can be turned into a simple questionnaire that lets both doctors and patients understand the diagnostic process. Since we assume we are able to discuss the model with public health and medical experts, the question of discretizing continuous variables can be tackled with domain expertise.

7. Whether boosting provides a linear classifier depends on whether the underlying classifiers are linear. Boosting creates a linear combination of the classifiers, so the boosted model is linear in the *classifiers*, but if the classifiers are non-linear, then the boosted model is non-linear in the *inputs*.

8. **No**, boosting can perfectly classify all the training examples only when the data are linearly separable.

9. For each of the $d = 10$ features, there are $2^d = 2^{10}$ ways to incorporate each feature into the tree, since each feature is binary. Since the output is also binary, we can label all possible points with $2^{2^d} = 2^{1024}$ possible trees.

10. **Maybe**. In order to make a decision, we first need to know the baseline performance of existing heuristics on this domain because the reported metrics may be worse than rules-of-thumb used in prenatal medicine to identify people at risk of pre-term and adverse births. Further, we would need to know whether precision at 10% is the correct metric; if the goal is to save lives, AUC-ROC may be the appropriate metric to optimize, while if the goal is to intervene in the costliest cases, we may want to focus on precision at 5% or 1%.

11. **False**. Due to bad randomization, the subsets of data used to train each tree in a random forest may cause the ensemble to perform worse than a single decision tree trained on the entire dataset.

12. Separate models are appropriate when there is a strong sociological or empirical reason to think there

13. A combined model is appropriate when the output is independent of gender (which may be hard to verify *a priori*), if the separated class has so few samples that the privacy of people comprising the data set might be violated, or if there is a legal or regulatory constraint preventing different treatment by gender.

14. Comparison:

SEPARATE MODELS

Pros:

- Models can be trained in parallel.

Cons:

- Could lead to overfitting if one of the gender classes has very few data points.
- In decision trees, forces an effective information split that cannot be used to backtrack and fix the purity of imputed classes.

COMBINED MODEL

Pros:

- Higher number of datapoints (useful when algorithm performance scales with amount of data).

Cons:

- Learning features on high-dimensional data sets may not pick up subtle but important gender-based attributes on a combined model.

15. I would need more information on the social service program being studied, and how this model would be used. Unless there are strong econometric or sociological reasons to build separate models (e.g. the social service program provides help/training to members of professions that happen to be dominated by one gender, or there is an identified mechanism causing different behavior by gender), I would proceed with a combined, gender-blind model.

2 Section B: Methods, Evaluation, and Communication

2.1 Decision Trees

A. The target variable `EnergyConsumption` takes on the value HIGH 6/10 = 60% of the time. Using a baseline random guess will have an accuracy of 60% (assuming the label HIGH is considered a positive prediction).

B. $H_0 = -\sum_i p_i \cdot \text{LOG}_2(p_i) = -(0.4) \cdot \text{LOG}_2(0.4) - (0.6) \cdot \text{LOG}_2(0.6) = 0.970951$ bits.

C. $\Delta H = H_0 - p(\text{POOR}) \cdot H(\text{POOR}) - p(\text{EXCELLENT}) \cdot H(\text{EXCELLENT})$

We know $p(\text{POOR}) = 5/10 = 0.5 = p(\text{EXCELLENT})$.

Additionally, $H(\text{POOR}) = -\sum_i p(c_i \mid \text{POOR}) \cdot \text{LOG}_2\left(p(c_i \mid \text{POOR})\right) = -0.2 \cdot \text{LOG}_2(0.2) - 0.8 \cdot \text{LOG}_2(0.8) = 0.721928$

Similarly, $H(\text{EXCELLENT}) = -0.6 \cdot \text{LOG}_2(0.6) - 0.4 \cdot \text{LOG}_2(0.4) = 0.970951$

Therefore, $\Delta H = 0.1245115$ bits.

D. Using the ID3 algorithm, we can construct the following two-stage decision tree by choosing to split on the attribute with the the maximal information gain at each level. For the parent dataset, the information gains for each attribute are listed below:

ATTRIBUTE	INFORMATION GAIN (BITS)
`Temperature`	0.20999
`HomeInsulation`	0.12451
`HomeSize`	0.00999

Table 1: Information gains for attributes considered in first-level split.

The root node should split on `Temperature`, since the maximum information gain occurs when splitting on that attribute.

For the second level, we can again calculate the information gains:

TEMPERATURE VALUE	ATTRIBUTE	INFORMATION GAIN (BITS)
HOT	`HomeInsulation`	0.25163
	`HomeSize`	0.91830
MILD	`HomeInsulation`	0.0
	`HomeSize`	0.0
COOL	`HomeInsulation`	0.41997
	`HomeSize`	0.41997

Table 2: Information gains for attributes considered in second-level split.

Splitting on points where `Temperature` is MILD creates a pure class. Looking at points where `Temperature` is LOW, the information gain is the same for both splits, so we choose to split on the attribute `HomeInsulation`. Using these splits, we can create a graphical representation of the decision tree (leaf nodes with predicted labels are highlighted in blue):

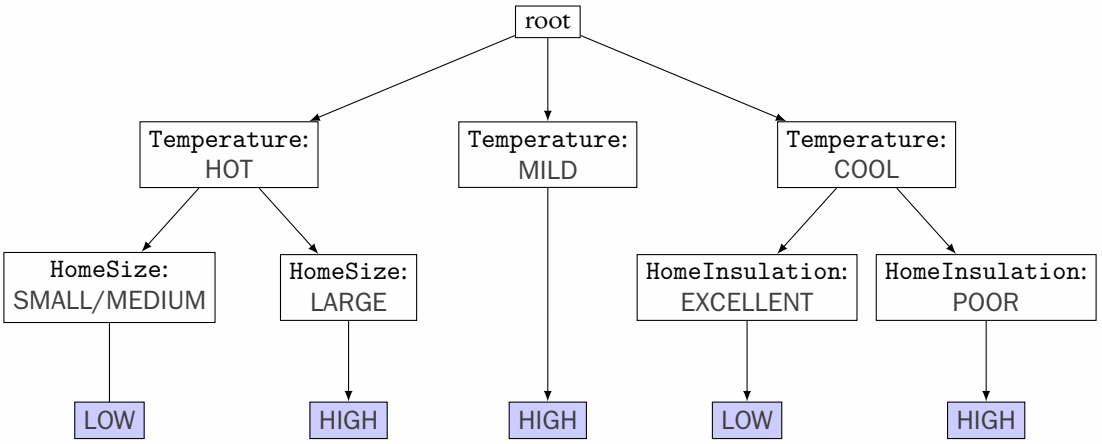


Figure 2: Visualization of decision tree generated by ID3 algorithm.

We can now evaluate the decision tree on the whole example set (generally, we would prefer to split into test and training sets, but we have very few data points in this example):

INDEX	TRUE LABEL	PREDICTION
1	LOW	LOW
2	HIGH	HIGH
3	LOW	LOW
4	HIGH	HIGH
5	LOW	LOW
6	HIGH	HIGH
7	HIGH	HIGH
8	LOW	LOW
9	HIGH	LOW
10	HIGH	HIGH

Table 3: Comparison of predictions to true labels. Misclassifications are highlighted in red.

	Actual LOW	Actual HIGH
Predicted LOW	4	1
Predicted HIGH	0	5

Table 4: Confusion matrix for decision tree.

2.2 Evaluation 1

A. Accuracy is the ratio of the number of correct predictions to the number of total predictions. Without a specified threshold, asking about overall accuracy is underdefined. We can calculate the accuracy of the SVM prediction under different thresholds and choose the threshold that maximizes accuracy. At each threshold t, we interpret the SVM output (p) as a positive prediction (1) if $p > t$.

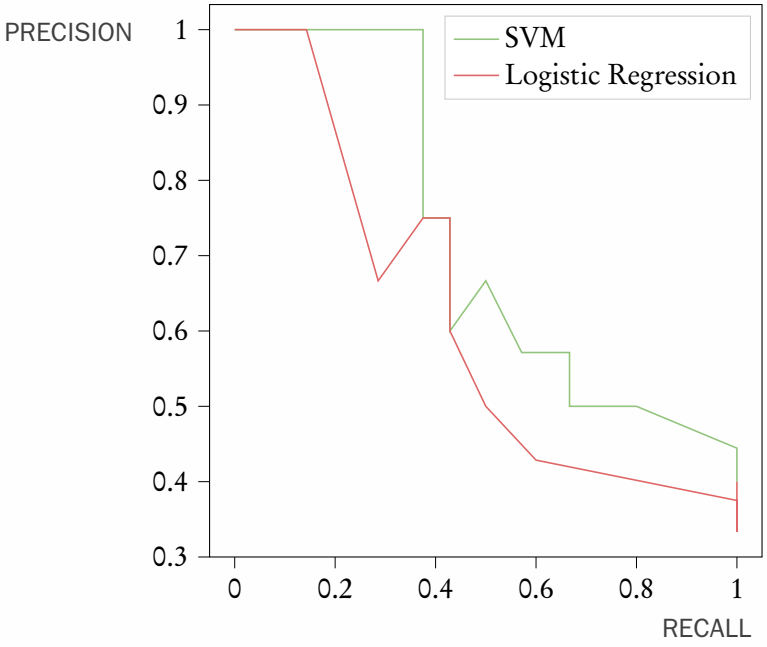
Letting accuracy vary from 0 to 1 in increments of 0.1 yields the accuracy curve in Table 5.

At a threshold of 0.7, we maximize accuracy at 0.9. Choosing this value as our threshold, we can say the SVM here has an accuracy of 90%.

THRESHOLD	ACCURACY
0.0	0.4
0.1	0.6
0.2	0.7
0.3	0.7
0.4	0.7
0.5	0.7
0.6	0.8
0.7	**0.9**
0.8	0.8
0.9	0.8
1.0	0.6

Table 5: Accuracy of SVM at various probability thresholds.

B. From the definitions of precision and recall, we can generate precision-recall curves for each of the classifiers:



$$\text{PRECISION} = \frac{TP}{TP + FP}$$

$$\text{RECALL} = \frac{TP}{TP + FN}$$

Figure 3: Precision-recall curves for the SVM and Logistic Regression classifiers.

C. Assuming precision and recall are the metrics we care about, the SVM dominates the logistic regression in PR-space; i.e. for a given precision the SVM has a higher (or equal) recall, and for a given recall, the SVM has a higher (or equal) precision. We also must assume 10 examples are sufficient to test the classifiers.

2.3 Evaluation 2

A. During the process of tuning the model, we let the predictions range from overly pessimistic to overly optimistic. When predictions are overly pessimistic, the model is unable to correctly classify many points. Once the model uses a more reasonable range for prediction, its precision improves.

B. The dataset may be skewed with very few negative examples, meaning the points learned as negative have a low, but non-zero, model score. Letting the threshold stay near zero will misclassify all of these points.

C. Boosting with a penalty for misclassification/misprediction for data points falling the top 5% will force the performance at the top 5% to increase, at the cost of performance at the bottom 95%.

2.4 Evaluation 3

A. In general, $L2$ penalty outperforms the $L1$ penalty on this dataset, measured across all available metrics. No single value of the cost parameter dominates the others across all metrics that we have access to.

B. 1) If the goal is to prioritize interventions with the 5% highest-risk population, then precision at 5% is the guiding criterion. The classifier performing best under this metric is **logistic regression** with a cost of $C = 0.1$ (regardless of whether the penalty is $L1$ or $L2$). Precision at 5% for this classifier is 0.82.

   2) If the resources available for interventions have not been determined, I would choose AUC-ROC as the metric on which to focus because it captures performance across the entire population. Under this metric, **logistic regression** with the penalty set to $L2$ and the cost set to 0.1 maximizes AUC-ROC at 63%.

2.5 Communicating Results

A. There are a number of things to keep in mind:

- A risk score is not necessarily a prediction and the model is not saying that Jenny has a 50% chance of not graduating. Depending on the model, it may be a ranking (i.e. Jenny is only medium risk), or may fall below the threshold for prediction.
- Jenny may have some characteristics that are not evident at present but that the model has picked up on. For example, Jenny, like some bright students, may be at risk of burnout but has not burned out yet.

B. The clearest way to confirm the accuracy of the model is to find other students like Jenny and compare the model output to their actual outcomes. Additionally, determining the model's goal (e.g. precision across the population or precision at a certain X%) will influence whether it even matters that Jenny's score is 50/100.

3 Section C: Solving a New Problem

A. I would formulate the record-linking problem as a supervised machine learning problem by forming a vector from record from each source of data (Corrections, EMS, etc) with the following structure:

$$\mathbf{v}_n = (\ \text{FNAME}, \text{LNAME}, \text{DOB}, \text{ADDRESS}, \text{GENDER}, \text{RACE}, \text{SOURCE}\ )$$

To turn these vectors into training data, we can apply a function f to each pair of rows in the records:

$$f\left(\mathbf{v}_i, \mathbf{v}_j\right) = \left(S(\text{FNAME}_i, \text{FNAME}_j),\ S(\text{LNAME}_i, \text{LNAME}_j),\ S(\text{DOB}_i, \text{DOB}_j),\ \cdots,\ \text{SOURCE}_i,\ \text{SOURCE}_j,\ \text{LABEL}_{ij}\right)$$

Here, $S(x, y)$ is a text similarity metric, such as Levenshtein edit distance or Jaccard coefficient. For fields such as gender or race, we may want an exact match. The component LABEL_{ij} (the column to predict) is a binary variable equal to 1 if vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ should be linked, and 0 otherwise. This label can be found from either a known group of linked records, or by having domain experts manually annotate a suitable number of example record pairs.

From this seed dataset of known linked vectors, we can generate many training rows by the following heuristic. If we have vectors $\{\mathbf{v}_1,\ \mathbf{v}_2,\ \mathbf{v}_3,\ \mathbf{v}_4\}$, and we know $\mathbf{v}_1$ and $\mathbf{v}_2$ should be linked, we can generate 12 rows of training data where the similarity components are found from applying our text similarity metric, and our labels would be 1 for the rows comparing $\mathbf{v}_1$ to $\mathbf{v}_2$ (and *vice versa*), and 0 for all other pairwise combinations of the vectors.

As an example using just full names, dates of birth, and record source where we know the first two vectors should be linked:

$$\mathbf{v}_1 = (\texttt{Tony Stark}, \texttt{11-05-1958}, \texttt{`CORRECTIONS'})$$
$$\mathbf{v}_2 = (\texttt{T. Stark}, \texttt{11-05-1958}, \texttt{`EMS'})$$
$$\mathbf{v}_3 = (\texttt{Carol Danvers}, \texttt{07-11-1967}, \texttt{`EMS'})$$

Using notional values for the similarity metrics, the training data would look like:

VECTOR INDICES	NAME SIMILARITY	DOB SIMILARITY	SOURCE 1	SOURCE 2	LABEL
(1, 2)	0.86	1.00	`CORRECTIONS'	`EMS'	1
(1, 3)	0.14	0.00	`CORRECTIONS'	`EMS'	0
(2, 1)	0.86	1.00	`EMS'	`CORRECTIONS'	1
(2, 3)	0.06	0.19	`EMS'	`EMS'	0
(3, 1)	0.14	0.00	`EMS'	`CORRECTIONS'	0
(3, 2)	0.06	0.19	`EMS'	`EMS'	0

B. As discussed above, the features would be:

- first name similarity
- last name similarity
- address similarity
- gender match
- race match
- source 1
- source 2

C. Logistic regression or decision trees seem appropriate in this case. Logistic regression has the advantage of returning risk scores that can be thresholded by application - instances requiring high match confidence can use a higher threshold, while quick matches can use a lower one.

D. Accuracy and AUC-ROC curve are important for this example. Given the data sources and applications, both false positives and false negatives could have disastrous consequences. For example, being unable to match a homeless person to their record in the Department of Mental Health may prevent them from receiving the appropriate psychiatric care. Conversely, matching to the wrong record from Emergency Medical Services may cause the administration of the incorrect drugs. Both accuracy and AUC-ROC penalize false positives and false negatives and should therefore be optimized.

E. Given the necessary paucity of positive labels in the generated training/testing data, it is unclear whether the chosen machine learning technique will outperform rule-based approaches.