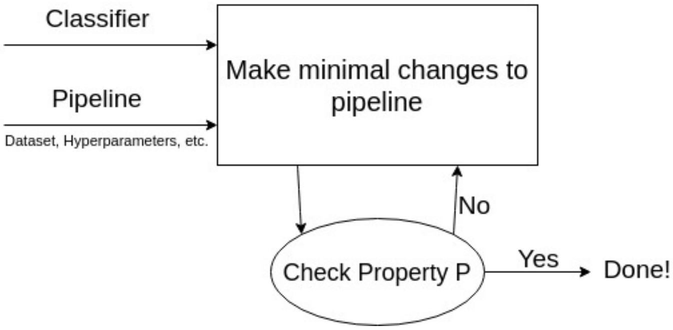
CS-799: Notes

Goutham Ramakrishnan

Guide: Prof. Aws Albarghouthi

Broad goal:

Given a machine learning pipeline with a dataset and classifier, find a minimal set of changes which need to be made to the pipeline for it to satisfy a given property P.


Classifier
Pipeline
Dataset, Hyperparameters, etc.
Make minimal changes to pipeline
No
Check Property P
Yes
Done!

Ideas:

- Use clustering (K-Means? Hierarchical? Semi-supervised?)
- Make changes to dataset iteratively
- Changes made must be interpretable
- Use training data debugging using trusted items as Property P to be starting point for exploring approaches

Discussion with Xuezhou:

- Using clustering as starting point for DUTI
- Active Debugging: Prof. Po-Ling's student working on it. Requires periodic expert feedback.
- For interpretability, find bugs first and then learn **rule list**
- Debugging the machine learning pipeline: Talk by Prof. Jerry Zhu (http://pages.cs.wisc.edu/ jerryzhu/adversarial/pub/debugML.pdf)
- Difficult to test approach due to lack of good datasets for this kind of task

Thoughts on clustering:

- Advantage of hierarchical clustering over K-means: The output hierarchy is more informative than output of K-means, number of clusters need not be pre-specified, can use dendogram to choose K. Tradeoff: Speed and efficiency
- Review on Semi-supervised clustering methods: https://arxiv.org/pdf/1307.0252.pdf
- Constrained clustering looks promising, can enforce cannot-link constraints on differently labeled trusted items
- Need to read up more on semi-supervised hierarchical clustering

Formulation:
Training set: $(X, Y) = \{(x_i, y_i)\}_{1:n}$
Trusted set: $(X, Y) = \{(\widetilde{x_i}, \widetilde{y_i}\}_{1:m}$
Number of clusters: K, Set of clusters $S = \{S_1, \ldots, S_K\}$

K-Means:

$$\underset{S}{\operatorname{argmax}} \sum_{k=1}^{K} \sum_{x \in S_i} \| x - \mu_k \|_2^2 = \underset{S}{\operatorname{argmin}} \sum_{k=1}^{K} |S_k| Var(S_i)$$

To minimize variance, add:

$$\sum_{k=1}^{K} Entropy(y_i | x_i \in C_k)$$

To ensure trusted items with different labels are not in the same cluster, add:

$$\sum_{\widetilde{x_i}, \widetilde{x_j} : \widetilde{y_i} \neq \widetilde{y_j}} Cost(\widetilde{x_i}, \widetilde{x_j}) \mathbb{1}(C(\widetilde{x_i}) = C(\widetilde{x_j}))$$

$C(\widetilde{x_i})$ is the cluster to which $\widetilde{x_i}$ belongs.

Notes: October 26

- Finding the ideal number of clusters K for a dataset is an unsolved problem, and there is no universally agreed upon metric upon which K can be decided. Methods include the elbow method, silhouette analysis, gap statistic (define a maximum number of clusters choose K with highest gap statistic), CH index etc.

- **Gap statistic:** Can be used for automatically deciding number of clusters. Compares average distance between points in a cluster with respect to a reference dataset (drawn from uniform distribution of features).
(http://www.sthda.com/english/wiki/print.php?id=239)

- Hierarchical clustering cannot be used for "predicting" the cluster of a new point (Implying that in our current procedure, trusted items need to be added to the buggy dataset before clustering). An alternative strategy is to use Agglomerative Clustering, and then train another classifier (like KNN) that can do predictions. However, clustering by itself is not a means of classification, and is hard to justify (as each new data point may potentially modify the clustering).

- **Choosing K using dendrograms:** Dendrograms are a visual heuristic, and are not practical to view when the number of data points is large. Moreover, inspecting dendrograms and deciding K is also a subjective process.

- Clustering as a concept is difficult to justify when the features are categorical. The primary hurdle then becomes defining an appropriate "distance" between points. This can be highly dataset dependent.

- Two possibilities are the generalized Mahalanobis distance and the Gower distance. With these distance measures, KMeans algorithm cannot be used directly (as the "centroid" of each cluster ceases to have any meaning). Hierarchical clustering is our best bet in this regard.

- **Gower distance:** General formula for estimating "distances" between datapoints when features are a combination of categorical and numeric.
(http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/MetQ/Talk6.pdf)

- **Choices of Linkage in Agglomerative Clustering:** The three simplest linkage strategies are complete, single and average which minimize the maximum, minimum and average "distances" between points in two clusters. All three give very different clusterings in the end. Single linkage in particular suffers from chaining (too many points in one cluster), this also occurred during my experiments with the GermanLoan dataset. (Complete linkage worked acceptably)

- **Experiments:** Conducted experiments on German Loan dataset using Gower distance, agglomerative clustering with complete linkage. Dendrogram is hard to interpret. The distribution of trusted items is across clusters, flipping one cluster at a time did not lead to any conclusive results.

31st October:

DUTI Testing Methodology:

- Bugs: Mislabled datapoints, which need to be flagged by the algorithm.
- DUTI makes changes to the dataset (each change is a flagged bug) such that minimizes the number of misclassified trusted items, by optimizing their loss function (It may not be necessary that all trusted items are classified correctly even after the changes).
- The examination budget (= max number of changes) defines a natural stopping point for making changes.
- The changes are made in successive iterations (does this mean that the most significant changes are made first, as the gradient of the loss function is used?)
- The correctness of these changes are evaluated using the PR curves obtained by varying examination budget.

Our Experiments:

- So far we have tried 3 different label flipping strategies (All, Trusted, Flip-One) with different clustering algorithms (KMeans, Agglomerative with different linkages, Pairwise Constrained).
- When we change the labels of an entire cluster all at once, we hurt our precision. When we change the labels of only one cluster at a time, we hurt our recall.
- The PR curves for our clustering procedure are extremely arbitrary, defies the normal shape of PR curves. I am not even sure if its appropriate to draw PR curves in our case.
- The only benchmark we have to judge whether our algorithm is working well are whether the trusted items are being classified correctly after the changes we make. So far, experiments have not been encouraging as the clustering procedure is not able to find a relabeling that classifies all trusted items correctly.

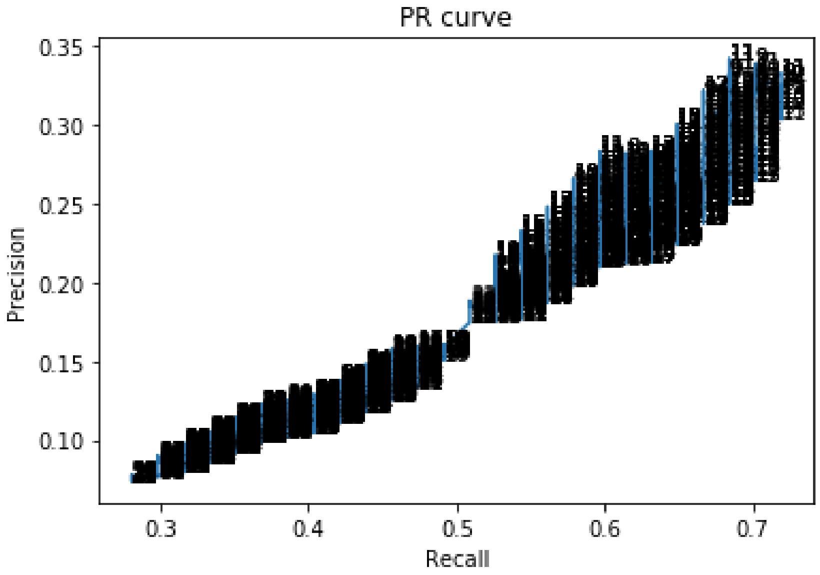
While digging around I found another interesting approach, which uses the pairwise constraints to choose from different clusterings: https://arxiv.org/pdf/1609.07272.pdf

Primary Concern:
How do we reason about using clustering, and how do we guarantee that the changes we make are in the right direction? None of the techniques has so far yielded a label modification so that all trusted items are classified correctly. Moreover, even if we hit upon such a relabeling, how do we justify that these changes are actually 'bugs'?

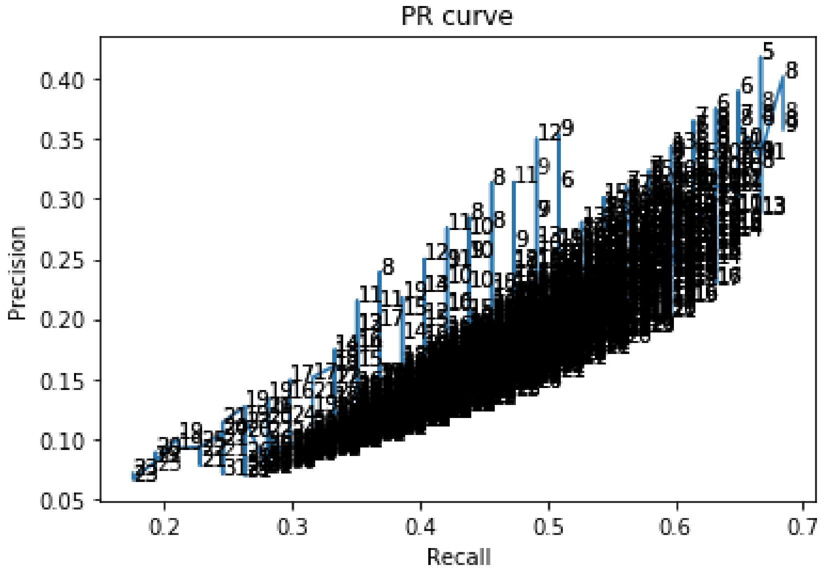
Some graphs of results:
Labels of points correspond to the number of mislabled trusted items

1. Clustering mode: all (brute force) K=2-12

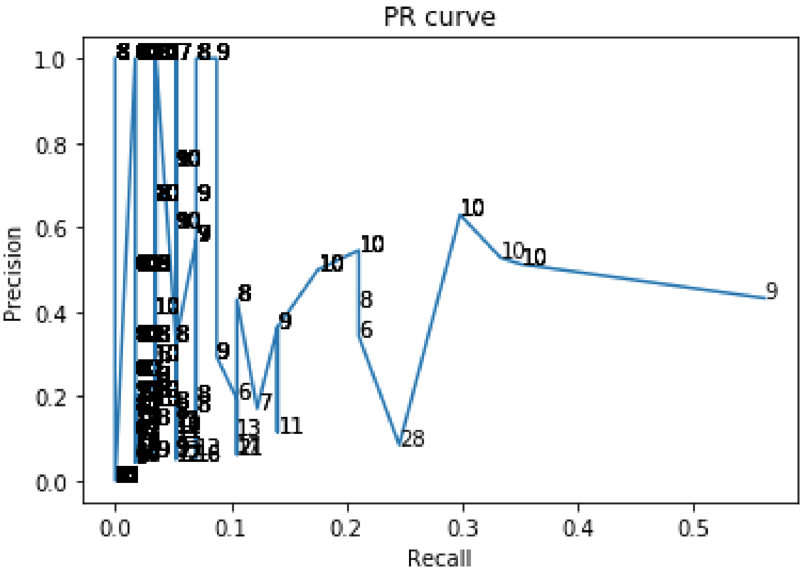
Agglomerative



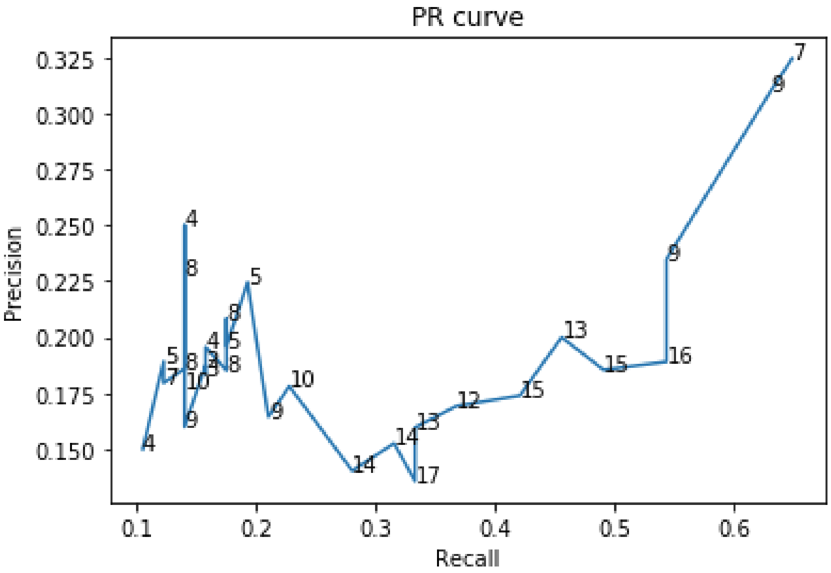
Pairwise Constrained



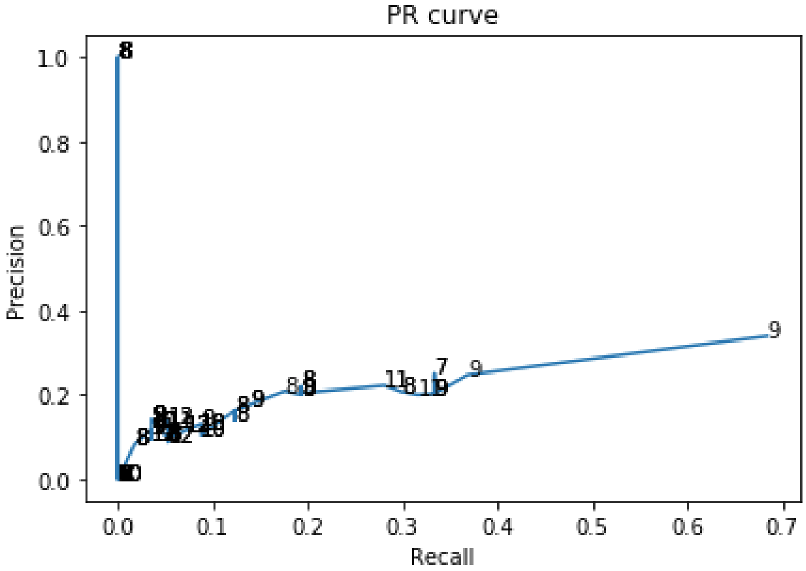
2. Clustering mode: Flip One
Agglomerative K=2-200



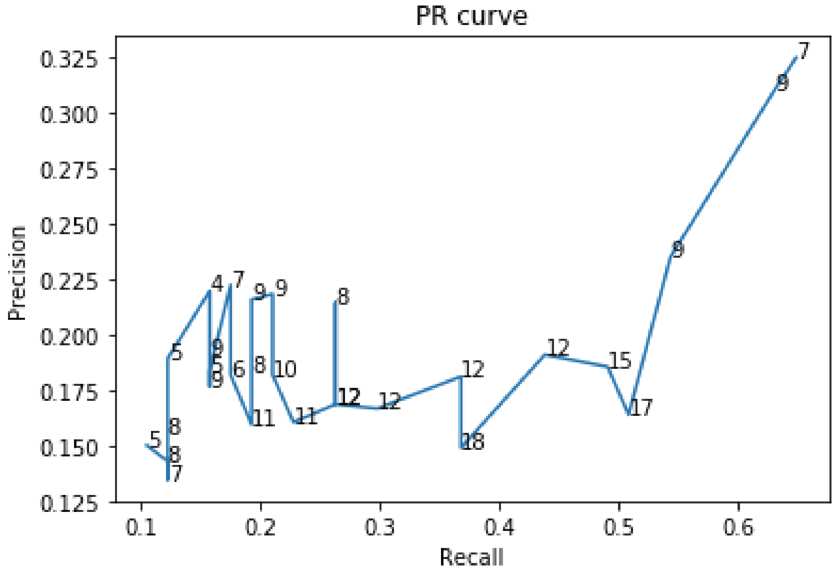
Pairwise Constrained K=2-30



3. Clustering mode: trusted
Agglomerative K=2-380



Pairwise Constrained K=2-30



Observations:

- At no point in any of the experiments did all the trusted items get correctly classified.
- PR curves are arbitrarily shaped, with very poor precision in general
- There are some points using the Flip-1 clustering obtained at a very large value of K which have good precision and poor recall.
- No general pattern between the points on the PR curve and number of misclassified trusted items