Novice-Centric Visualizations for Machine Learning

by

Yunjia Sun

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2016

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis focuses on visualizations for machine learning tasks. More specifically, we create a taxonomy for existing machine learning visualizations, and design a system to help machine learning novices perform labelling tasks.

There are many mature visualizations to help people understand the performance of current classifiers, including scatterplots, confusion matrices and ROC curves. However, most machine learning researchers are unaware of the visualization possibilities that exist, and many published visualizations are too task-oriented or dataset-oriented to be easily applied to other tasks. This thesis defines a taxonomy for machine learning visualizations in three dimensions: the data displayed, the advanced features to add for a specific task, and the goal of the visualizations. This taxonomy seeks to help machine learning researchers select a better visualization method to analyze their data.

Previous machine learning tools focus on presenting comprehensive information to experts, treating machine learning as a black-box for end-users, or explaining the reason behind the prediction in a simple and clear way. However, to build a machine learning system, one needs to label data first, and a lot of machine learning novices want to build a classifier themselves simply by labelling data. This inspired our idea to design and implement the Label-and-Learn system, which includes five visualizations to help users better understand their data, the likelihood of the classifier's success, and to improve their user experience.

To evaluate the utility of our Label-and-Learn system, we ran user studies to compare the visualization system and traditional system in the quality of the labels, the user's mental model about the task, and the user experience. The results from the experiment show that visualizations have no negative effect on the quality of the labels, but do improve the user's mental model and the user experience. The success of the Label-and-Learn system should inspire further research in using visualizations to improve the user experience of data labelling in machine learning tasks.

Acknowledgements

This thesis cannot be completed without the help from many people. I would like to take this opportunity to thank them all.

First, I would like to thank my first and primary supervisor, Michael Terry. You introduced me to the academic research area, taught me how to come up with a research idea and how to carry it out. You also guided me through the process of visualization design, a completely new area for me. Your personality of being bold, innovative, enthusiastic and optimistic inspired me, excited me and supported me through the course of my graduate study. I still remember clearly the scene where we brainstormed dozens of ideas and finally concentrated on five, and the scene where I questioned about the value of my research and you kindly discussed with me from the perspective of industry. Thanks for introducing me the world of HCI.

I would also thank Ed Lank and Edith Law, who supervised me through the user study, data analysis and thesis writing part. Your advice and suggestions helped me efficiently go through the process. I would also thank the members of my committee: Parmit Chilana and Pascal Poupart. I appreciate your time in reading my thesis and offering suggestions to make it complete.

I would also say thanks to all my lab mates in the HCI group: Adam Fourney, Jingjie Zheng, Qifan Li, Qifeng Liu, Hemant Surale, Jeff Avery, Alix Goguey, Bahar Sarrafzadeh, Mingyu Liu, Mathiu Nancel, William Saunders, Keiko Katsuragawa, and all those people who helped in piloting my study and offered valuable suggestions on my system design. I would specially thank Adam Fourney for providing the *Online Documentation* dataset and code. The Named Entity Recognition part of my system was implemented based on this resource.

Last but not least, I want to thank all the participants who expressed interest in my research and devoted two hours to my task. I cannot complete my research without your help.

Dedication

This thesis is dedicated to my dear parents, Kuiqiang and Jianxin, whose enduring love and guiding advices supported me through my thesis writing, graduate study, and my life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Machine learning tools are quickly becoming accessible to non-experts. Given a labelled dataset, they can produce classifiers with little-to-no-effort on the part of the user. These systems include research systems such as Crayons[6] and CueFlik[7], but also include recent systems such as Google's Prediction API and Amazon's machine learning service. They provide an interactive, accessible interface for people to adjust the input or parameters and see immediate feedback so they are very easy to manipulate. They hide the implementation details such as feature engineering, model selection, parameter tuning and basic evaluation in the backend, so people can train a personalized classifier without a machine learning or even a computer science background.

However, while libraries and toolkits lower the barrier of using machine learning, creating a system that successfully solves a novel problem is still non-trivial, and can require a fair amount of expertise to solve well. Information Visualization has the advantage of effective expression of a large amount of data. Numerous visualizations have been developed to help individuals understand the performance and current state of the classifiers. Existing visualizations range from scatterplots representing a classifier's confidence scores for labelled or unlabelled data, to confusion matrices representing how the misclassified data is distributed. These visualizations provide a compact summary of a particular view of the classifier and its likely performance with new, unlabelled data. With more development in machine learning, we also have more complicated visualizations such as ROC curve to assist parameter tuning and 2d projection to assist feature selection.

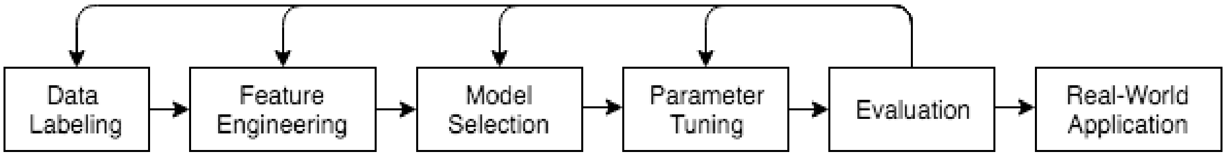


Figure 1.1: Machine Learning Pipeline

The visualizations mentioned above are expert-oriented and require some expertise in machine learning to understand well. Recently we also have a number of tools to assist non-experts in understanding and debugging classifiers. For example, Kulesza's work[17] provides natural language descriptions of why a classifier makes the prediction that it does. He shows that by explicitly explaining the feature values and each value's influence and distribution, non-experts can easily build up a mental model of how the classifier works. The mental model actually helps them personalize their classifiers. However, he primarily focuses on explaining the reasons for the classifier's performance and prediction.

Kulesza's work has done much to make machine learning accessible to non-experts, but a number of open issues remain. The machine learning pipeline contains the process of data labelling, feature engineering, model selection, parameter tuning, evaluation, and real-world application(Figure 1.1). Non-experts participate not only in the real-world application, but also in initial data labelling, which directly affects the classifier's quality. Researchers have typically focused on making the data labelling process efficient. In Settles' Active Learning[35], a system selects the instances that have the highest information gain and query their labels. The task load for the labellers is significantly reduced by applying this scheme. However, little work has been done to make the labelling process informative for the labellers.

We generalize that there are two classes of labellers: uninterested third parties hired by the classifier builders, and labellers who are directly invested in the outcomes (i.e., they are the people who will use the classifier). For the latter, the labelling period provides the first opportunity to gain an understanding of the data, and they hope to get answers for the following questions:

1. Can the classifier learn the desired concept well enough, given the data and features they have? As an example, if we identify that, for some reason, the data cannot be separated no matter how many labels we provide, meaning a concept cannot be learned, we should immediately try another model, extract more features, or even

collect more data. If all of these attempts fail, it is better to transfer the task to more expert researchers.

2. When can I stop labelling? When the classifier is unlikely to learn much more, or a lot of labelling effort would be required to gain appreciable performance, it is better to realize this situation as soon as possible to avoid wasting time on data labelling.

3. How will this data affect my classifier? Active learning keeps selecting the most informative data to query for label. It might be useful to understand the reason behind the classifier's choice, because sometimes the classifier might have bias on a certain branch of data, and sometimes there might be several data that make it hard for the classifier to make a decision. It is better to realize these situations to ensure that the most informative data is labelled, and quickly identify the uncertain data.

4. How stable is my classifier? For systems incorporating online learning, in which new labelled data is continually added to the system, it is useful to know the stability of the classifier to decide whether to accept more labelled data, and whether the current performance is acceptable.

In this thesis, we present a series of visualizations useful for labelling, and for predicting the likely performance of the classifier. More specifically, we present visualizations that 1) explain to users why this data needs to be labelled, 2) show users the change a label will bring to the classifier, 3) show users their progress in building the classifier so far, 4) show users the stability of the classifier, 5) show users the current state of the classifier, and 6) show users why the classifier makes a certain prediction.

1.2 Contributions

1.2.1 Taxonomy for Machine Learning Visualizations

Numerous visualizations have been developed to help individuals understand the performance and current state of the classifiers. They range from scatterplots representing a classifier's confidence scores for labelled or unlabelled data, to confusion matrices representing how the misclassified data is distributed. As we review previous visualizations to design our own, we realized that while these visualizations are designed for different tasks and datasets, show different aspects of data, and are drawn using different marks and channels, they still share quite a lot similarities; some of the visualizations are actually telling the same story.

Although research has been done on machine learning visualizations, machine learning researchers still stick to WEKA[41], or simply draw the most basic bar graphs, scatter plots by themselves to better understand their data. Part of the reason may be that they are unaware of the visualization possibilities in the field. Part of the reason may be that the visualizations published in research papers are too task-oriented or dataset-oriented, so it is hard to directly apply them to their new problem. Finally, part of the reason may be that it is hard to find the desired visualization from the numerous, unsorted existing work.

In this thesis, we define a taxonomy for machine learning visualizations with respect to its target users, whether the visualization shows selected data or all information, shows labelled data or unlabelled data, shows classified data or unclassified data, supports interaction or not, is feature-centric or instance-centric, is data abstracted as point or shown as it is, has dimensionality reduction or not, shows prediction's confidence or not, includes classifier's assessment or not, and highlights interesting information or not.

This taxonomy is able to classify existing visualizations with several tags, helps users understand the information it shows. We hope that with this taxonomy in mind, information visualization researchers can better understand their design target and existing work so they can design more expressive visualizations and avoid repetitive work. We also hope that this taxonomy will guide machine learning researchers to realize the possibilities of visualizations and help them choose the one that best solves their problem. They may not directly apply the existing visualizations to their data, but they can at least learn from them as a way to show certain aspects of data.

1.2.2 Label-and-Learn

Previous work on machine learning tools focus on (1) machine learning experts, (2) taking machine learning as a black-box, or (3) explaining the reason behind the prediction. We designed our Label-and-Learn system, which focuses on non-expert machine learning users who want to create a classifier themselves, and especially on their labelling experience. We hypothesize that the Label-and-Learn system will help non-expert users leverage the unavoidable process of labelling to develop insights into their data and the classifier.

The Label-and-Learn system includes six windows: five visualization windows and a labelling window. The visualizations update whenever the user labels a new piece of data. Some of the visualizations borrow ideas from existing works; the rest are designed by ourselves. All of them serve to help answer four questions: (1) Can the classifier learn the desired concept well enough, given the data and features they have? (2) When can I stop labelling? (3) How will this data affect my classifier? (4) How stable is my classifier?

To assess the effectiveness of Label-and-Learn, we designed a traditional version of Label-and-Learn which removes all the visualizations. We ran user studies on both versions to validate the usefulness of the visualizations. The results showed the success of the visualizations, which could largely improve the user's mental model about the classifier's performance. Users also reported a better experience when using the visualization interface. Meanwhile, our new design has no negative effect on the quality of the labels.

1.3 Thesis Outline and Terminology

The thesis is organized as follows:

- Chapter 2 describes previous work in human-in-the-loop machine learning, including applications of interactive machine learning that treat machine learning as a black box, interactive applications that make the machine learning process transparent to the end user, and tools for labellers.
- Chapter 3 defines the taxonomy for the visualizations on machine learning. We also review the most representative existing visualizations and classify them with our taxonomy.
- Chapter 4 describes our Label-and-Learn system in detail. It explains the machine learning task the system is designed for, design principles and mathematics behind each visualization.
- Chapter 5 describes the experiment we conducted to validate the usefulness of the Label-and-Learn system. It also presents the result from the study.
- Chapter 6 discusses the system's advantages, summarizes the contributions of the thesis, and suggests the future research to be done.

However, before continuing, it is worthwhile to precisely define some terminology that is used throughout this dissertation.

machine learning experts: Machine learning researchers who design algorithms to solve problems. They know the mathematics and rationale of the machine learning classifier such that they can program a classifier themselves.

machine learning novices: Machine learning novices are practitioners who want to build a classifier, but only know how a machine learning classifier works generally, and may not know the detailed mathematics. They should at least know that machine learning is a process that learns the patterns from the features of the data, and make predictions from the patterns.

concept: The general description of the information we want the system to learn. For example, whether an email is spam, which class this document belongs to, whether this "he" means a specific person.

instance: A piece of data in the dataset that has been labelled or is to be classified. For example, it can be an email, a document, or the word "he" in a specific context.

feature: An individual measurable property of an instance. For example, whether this email is from an address ended with "uwaterloo.ca", how many instances of "computer" this document contains, or the word that follows "he".

positive / negative: Match / Mismatch of the concept. If the concept is "Check if the email is spam", then a spam email would be a positive instance. On the contrary, a non-spam email would be a negative instance. If the concept is "Check if this document is computer-related", then a computer-related document would be a positive instance, and a non-computer-related document would be a negative instance.

confidence: The probability that an instance or a feature will be predicted by the classifier as positive. If it is predicted as 80% to be negative and 20% to be positive, then it's confidence is 20%.

stability: The likelihood that the classifier's decision will be little affected by a new label. Furthermore, we define the **stability of the feature**, which is the likelihood that an instance or a feature's confidence will be little affected by a new label with the same feature value. A classifier is more likely to be stable when it has more stable features.

For example, before we label the feature f for the first time, f's confidence was 0.5. After labelling one as positive, it suddenly changes to 0.9. Then we label another one as negative, it suddenly changes back to 0.5. At this moment, f's confidence changes significantly with a new label. We say it is unstable. However, when we have 70 f as positive, and 30 as negative, after labelling a new instance as positive, the confidence changes from 0.7 to 0.703. Then if we label another one as negative, it slightly changes back to 0.696. Because f

only changes slightly with a new label, we say it is stable. Generally, the more information, or labels, we have about a feature, the more stable it is.

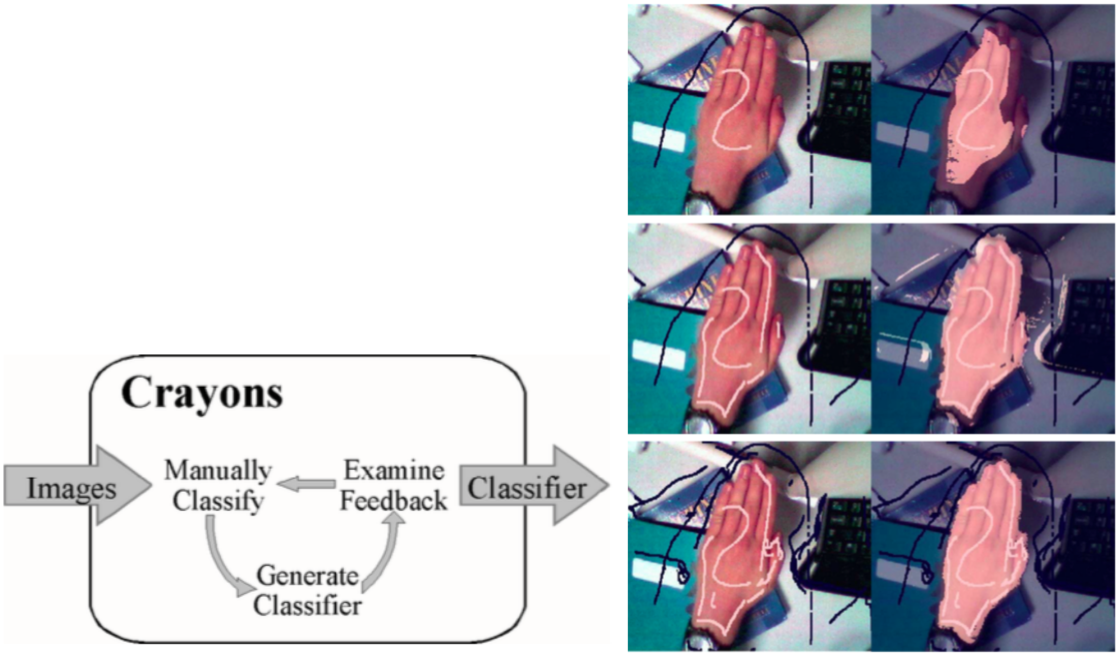
Chapter 2

Related Work

To build a machine learning system, one needs to go through the pipeline of data labelling, feature engineering, model selection, parameter tuning, evaluation, and real-world application. Although a large part of this pipeline can be automated thanks to work by statisticians and engineers, there are still parts of the process that cannot be done without the involvement of human beings, especially data labelling and real-world application. For the past decade, researchers from the area of Human-Computer Interaction, Machine Learning and Information Visualization have been looking into approaches to bridge the gap between humans' mental models and the system's statistical framework. In this chapter, we will review research from Human-Computer Interaction (HCI) and Machine Learning. Chapter 3 will define a taxonomy and review visualizations in detail.

2.1 Black-Box Interactive Machine Learning

With the hope that everyone, including experts and novices, can use machine learning to solve their problems, many HCI researchers are devoted to making it more accessible to novices, those who have little knowledge about machine learning. To help novices understand the classifier's performance without knowing the mathematics, researchers have designed systems that hide the machine learning implementation details in the backend as a black-box. Users can interact with the frontend interface to provide input, examine the output, and modify the input as the feedback loop. This section describes four applications which demonstrate this paradigm of black-box interactive machine learning: the Crayon system and three systems by Amershi: CueFlick, ReGroup, and CueT.



(a) Crayons Design Loop (b) Crayons Interaction Process

Figure 2.1: Interactive Machine Learning with Crayons

2.1.1 Crayons

Fails and Olsen's Crayon[6] is the first interactive machine learning system designed for end users. Its goal is to help UI designers who do not have detailed knowledge of image processing and machine learning to create image classifiers. A user first provides some labelled images as training data, and the system returns the current result. The user examines the result and provides more training data to correct and improve the model, forming a feedback loop(Figure 2.1a). Figure 2.1b is an example of a user interacting with the system. He first paints very little data that represents the skin, examines the result, and corrects by painting more. The system lowers the barrier of machine learning and fastens the classifier building process by hiding the implementation detail in a black box. User's responsibility is reduced to simply providing data and examining the result. The trade-off is that, users will never know the reasoning inside the black box, thus cannot decide whether this black box is suitable for this task, whether it will succeed, whether this black box can be generalized to other data or tasks, how much resource is needed, or whether this black box is stable.

2.1.2 Effective End-User Interaction with Machine Learning

Amershi advanced the domain of interactive machine learning in two ways. First, she proposed a design space for such systems[1]. When designing an interactive machine learning system, one should consider two design factors: (1) the interaction goals and contexts, and (2) the constraints of the data, user experience, and environment. One should also consider three design dimensions: system feedback, end-user control, and temporal factors. Second, to demonstrate that, she designed three systems:

- CueFlik[7] is a web image search application. It allows end-users to create their own rules for re-ranking images simply by dragging mouse. A user can first search images using a query, and select some images that match the rule, and when he searches images using another query, he can apply the rule to the new set of images to re-rank them. Figure 2.2 is an example of how a user creates the "scenic" rule from the search for "mountain" by dragging scenic related pictures to the designated area. The system learns the rule, and then the user can apply the rule to the search "water" and the search "car" so that the scenic related images are ranked at top.

- ReGroup[2] is an application that helps people create custom groups on demand in social networks. First, the user selects several friends he wants to add to the group. The system learns about the group and presents the user with relevant group characteristics to use as filters. This helps the user by reducing the number of people that need to be considered when creating the group.

- CueT[3] is a stream-based interactive machine learning engine for making triage recommendations. With the interface, it can assist the operator in inspecting and interpreting those recommendations and feeding operator actions back into the learning engine.

Amershi's design space is the first to summarize all the existing interactive machine applications. It could be a good guide for future researchers to design the system that best satisfies their need, as well as end users to choose the system. Our work is similar to hers in terms of summarizing previous work using a design space, but we focus on visualizations for both machine learning experts and novices. Furthermore, while her applications extended Crayons by applying interactive machine learning in more real-life systems, the reasoning

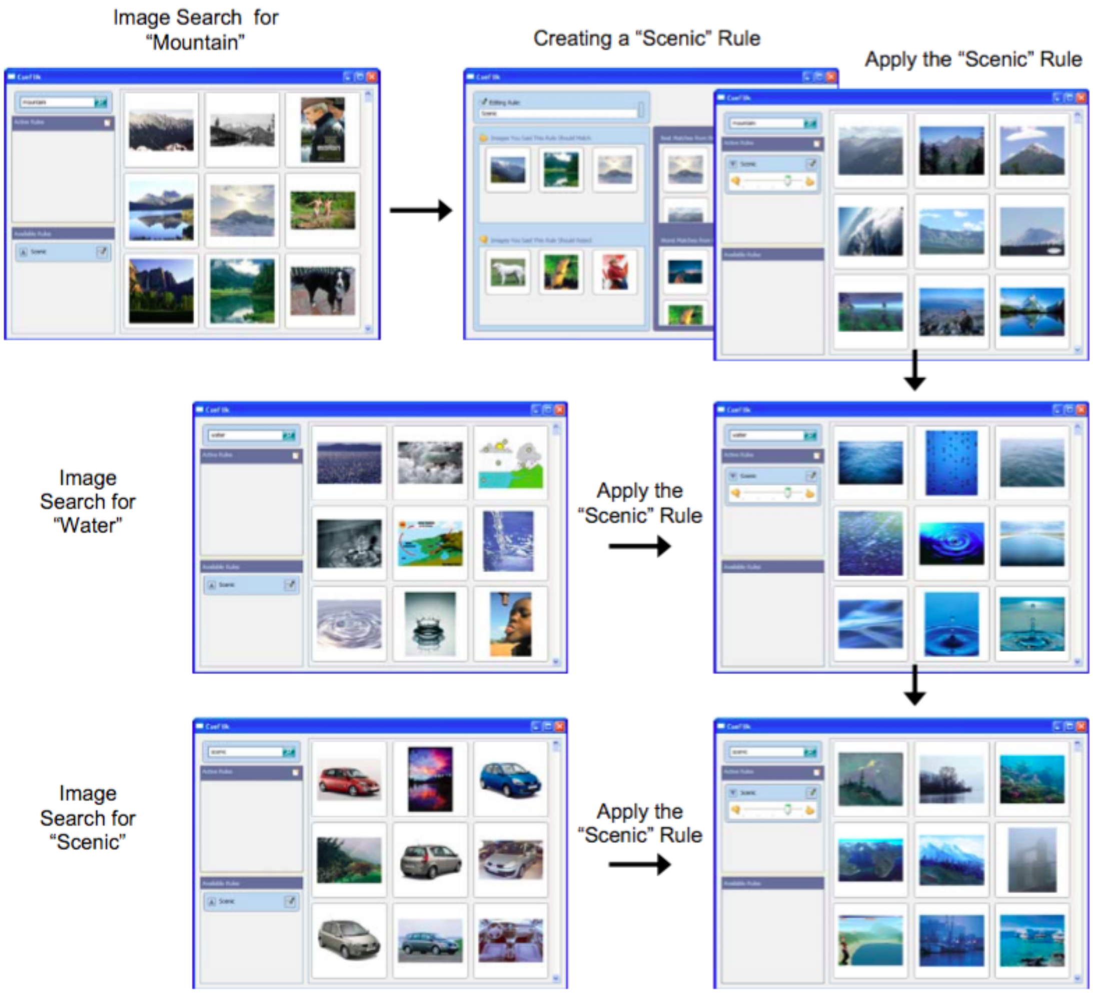


Figure 2.2: Using CueFlik to create a rule and apply that rule to image search.

of the classifier is still a black box. Users may know what features are influencing the classifier, but may be unsure of the way and the degree of the influence. It is also difficult for Machine Learning novices to determine whether the classifier is stable, where it is going to fail, and whether it can be generalized. Without this knowledge, users must trust that the system works as promised.

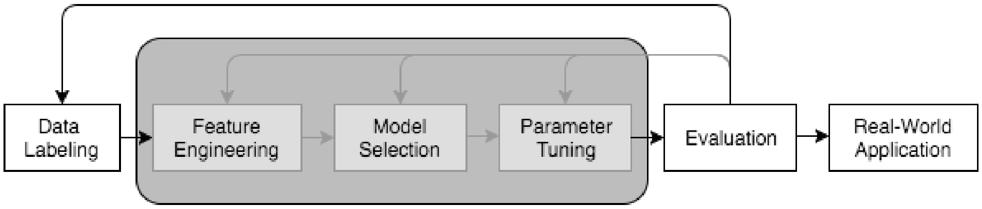


Figure 2.3: Black-Box Interactive Machine Learning Pipeline

2.1.3 Summary

Crayons is the first interactive machine learning system that applies human-in-the-loop to the classifier building process. It has the advantage of being accessible to everyone regardless of their background, has good performance on the designated task, and exhibits simplicity of interaction. Crayons has inspired researchers to develop similar systems for other tasks. These systems include Amershi's ReGroup[2], where users can input training samples as well as the features to classify friends. They also include Patel's Hindsight[26], where users can use the input from a dataset or even create the data themselves directly with the application, then examine the result on these inputs.

The challenge with the interaction machine learning systems described in this section is that, such systems encapsulate most part of the machine learning pipeline in a black-box, leaving only data labelling, evaluation, and real-world application to the user (Figure 2.3). They only show the output of the classifier's decisions and do not include visualizations that show the classifier's performance. Users know nothing about the implementation of the system so they do not know whether it is applicable for all of the dataset, just as we do not know what kind of email will be misclassified as spam. Moreover, such systems are task and dataset oriented, since the features and models are built-in. Crayons only works for image processing. CueFlik only works for image reranking. ReGroup only works for filtering, and CueT only works for alarm triaging. Users can only use the products, without the ability to build a classifier that works on a new problem. In brief, these systems are built for users who want to use an already-built classifier on a problem for which it has been a priori designed.

2.2 Transparent Interactive Machine Learning

Traditionally in the field of machine learning, we would tend to think of developers as those few talents who create magical classifiers by writing code and inspecting command line output, and think of end users as customers who happily use the magical classifier to solve their problems. With machine learning being used everywhere in our life, more classifiers are needed than being built, and the builders are no longer limited to a small number of highly talented developers. A system that integrates feature engineering, model selection, parameter tuning, and result inspection in one graphical user interface can improve the efficiency and lower the experience required to build a classifier. At the same time, with different applications that produce different outputs, the end users start to question the reasoning behind machine learning inferencing: Why does the classifier consider this email as spam? Why does it recommend this song to me? Users want the one and only classifier that can best solve their problem. To satisfy their needs, researchers have begun to build interactive machine learning systems that show the reasoning of the classifier, and let users adjust the classifier by manipulating the features, their weights, the input, or even the model from the classifier's interface.

In this section, we first focus on Patel's work simplifying the development of machine learning systems. We also describe work in enhancing classifier's transparency.

2.2.1 IDES for Machine Learning

Patel focused his research on improving the developer's experience. He has examined difficulties developers encounter in the adoption of statistical machine learning[27]: 1) difficulty following an iterative and exploratory process, 2) difficulty understanding the relationships between data and models, 3) difficulty evaluating the performance of models in the context of an application. He also implemented three systems to help developers overcome these difficulties:

1. Gestalt[24] integrates the implementation and analysis of the classification pipeline into one system. By separating the machine learning pipeline into several steps with input and output, Gestalt provides general support for all machine learning problems. The developers only need to follow the steps to read in the data, edit the code, and inspect the visualized result. It also provides connected visualizations to help users understand relationships between data, features and results.

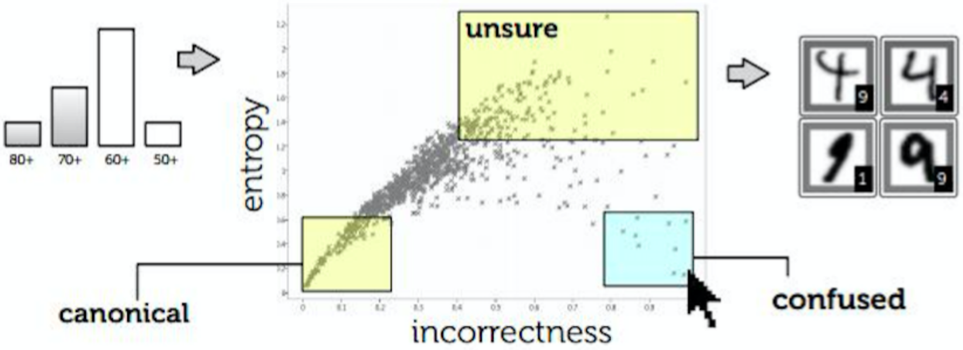


Figure 2.4: Prospect: Scatterplot of different configurations' results

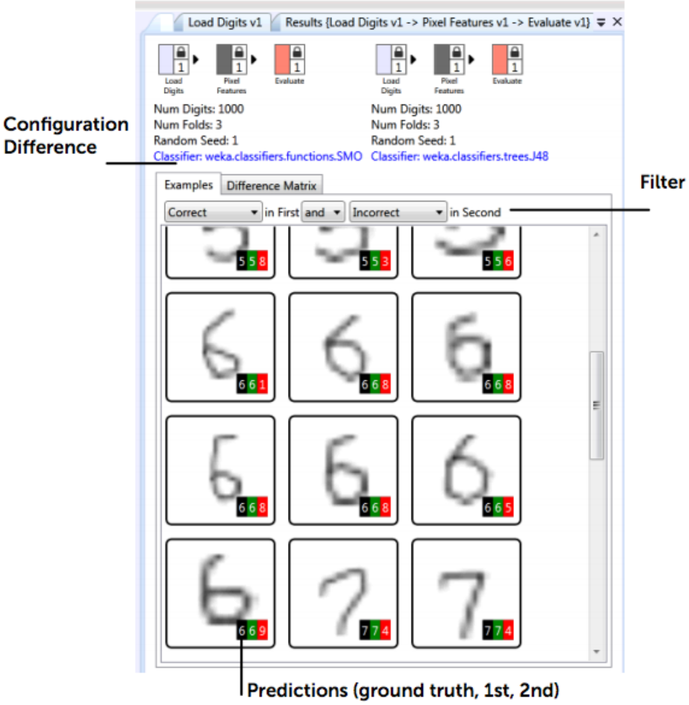
2. Prospect[25] is a system that uses multiple models to detect label noise and aid in generating new features. For each task to be solved, it generates several configurations and applies them to the data. This is to avoid the bias of a single configuration. The user can then inspect the result through visualizations(Figure 2.4) such as scatter plots and confusion matrices to detect outliers. These outliers may result from label noise, or from areas of uncertainty that can help developers generate new features.

3. Hindsight[26] is a system that focuses on comparison between configurations. A user can choose between loading data from files or drawing data directly on the system. To build a satisfactory classifier, users can try different configurations through code or through the interactive components of the system. Hindsight keeps track of all the configurations, and provides comparison on the single instance level and the whole dataset level(Figure 2.5).

2.2.2 Classifier Personalization by Building Mental Model

While Patel's research focused on developers, other research has focused on transparency for users. For example, Lim[19] and Kulesza[17] focused their research on helping users build a mental model of the classifier so that they can better personalize their application.

Lim and Dey implemented a toolkit[19] which helps users better personalize their application by answering the following questions such as: Why did it do X? Why did it not do Y? What if I did W, What will it do? How can I get the application to do Y? Their



Load Digits v1 | Results {Load Digits v1 -> Pixel Features v1 -> Evaluate v1}

Load Digits → Pixel Features → Evaluate | Load Digits → Pixel Features → Evaluate

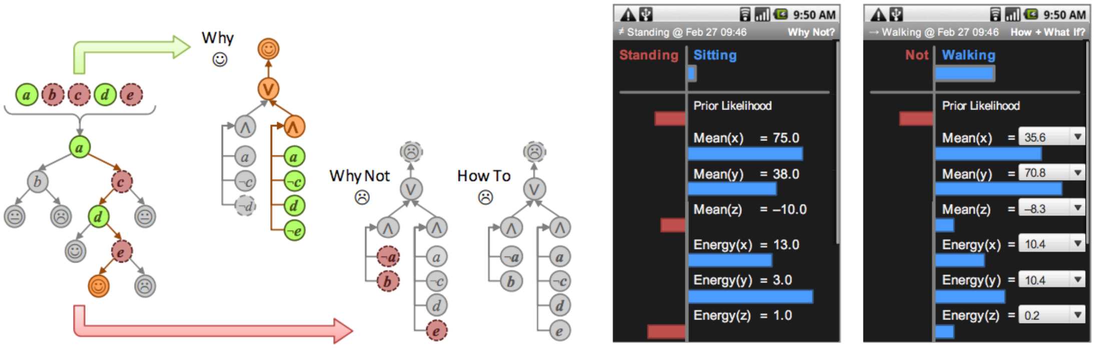
Num Digits: 1000
Num Folds: 3
Random Seed: 1
Classifier: weka.classifiers.functions.SMO

Num Digits: 1000
Num Folds: 3
Random Seed: 1
Classifier: weka.classifiers.trees.J48

Examples | Difference Matrix

Correct ▾ in First and ▾ Incorrect ▾ in Second

Figure 2.5: Hindsight: Comparing the result of different visualizations



9:50 AM
≠ Standing @ Feb 27 09:46 Why Not?
Standing | Sitting
Prior Likelihood
Mean(x) = 75.0
Mean(y) = 38.0
Mean(z) = –10.0
Energy(x) = 13.0
Energy(y) = 3.0
Energy(z) = 1.0

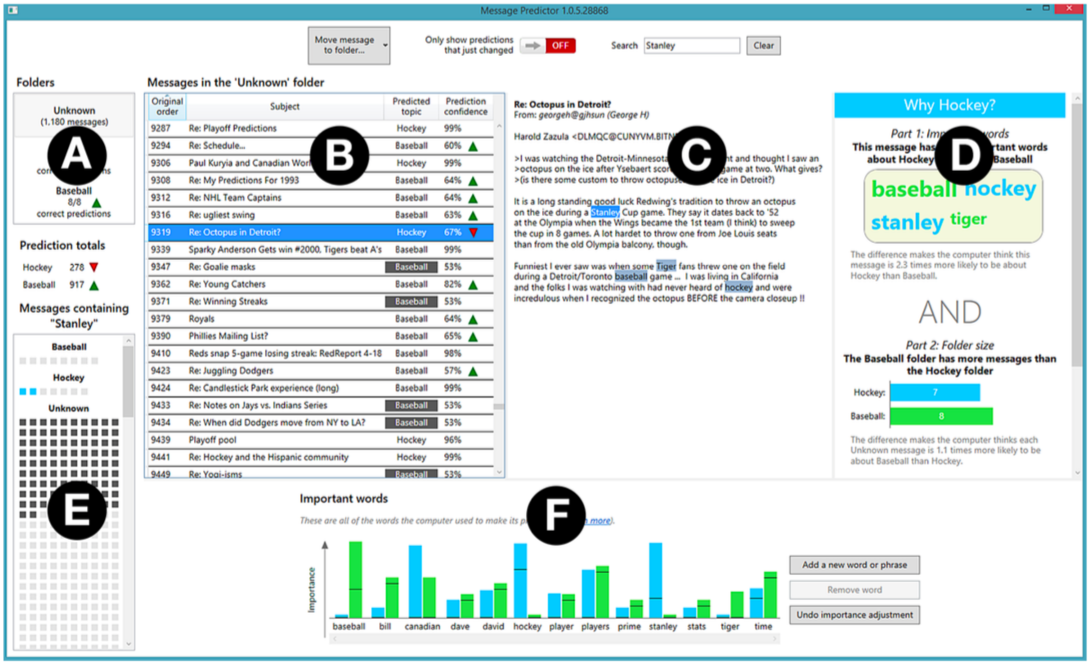
9:50 AM
→ Walking @ Feb 27 09:46 How + What If?
Not | Walking
Prior Likelihood
Mean(x) = 35.6 ▾
Mean(y) = 70.8 ▾
Mean(z) = –8.3 ▾
Energy(x) = 10.4 ▾
Energy(y) = 10.4 ▾
Energy(z) = 0.2 ▾

(a) Decision Tree

(b) Naive Bayes

Figure 2.6: Toolkit that Generate Explanations for End-Users

toolkit supports the visual explanation of rules, decision tree, naive Bayes, and HMM classifiers(Figure 2.6). They used their toolkit on three intelligible applications: IM (Instant Messaging) Autostatus, Mobile Physical Activity Recognizer and Home Activity Recognizer. They also ran user study on an intelligible mobile context-aware application. The result suggested a need for streamlining explanations while maintaining access to the rich explanation capabilities, and for integrating domain knowledge in explanations.



Messages in the 'Unknown' folder

Original order	Subject	Predicted topic	Prediction confidence
9287	Re: Playoff Predictions	Hockey	99%
9294	Re: Schedule...	Baseball	60% ▲
9306	Paul Kuryia and Canadian Worl	Hockey	99%
9308	Re: My Predictions For 1993	Baseball	64% ▲
9312	Re: NHL Team Captains	Baseball	64% ▲
9316	Re: ugliest swing	Baseball	63% ▲
9319	Re: Octopus in Detroit?	Hockey	67% ▼
9339	Sparky Anderson Gets win #2000, Tigers beat A's	Baseball	99%
9347	Re: Goalie masks	Baseball	53%
9362	Re: Young Catchers	Baseball	82% ▲
9371	Re: Winning Streaks	Baseball	53%
9379	Royals	Baseball	64% ▲
9390	Phillies Mailing List?	Baseball	65% ▲
9410	Reds snap 5-game losing streak: RedReport 4-18	Baseball	98%
9423	Re: Juggling Dodgers	Baseball	57% ▲
9424	Re: Candlestick Park experience (long)	Baseball	99%
9433	Re: Notes on Jays vs. Indians Series	Baseball	53%
9434	Re: When did Dodgers move from NY to LA?	Baseball	53%
9439	Playoff pool	Hockey	96%
9441	Re: Hockey and the Hispanic community	Hockey	99%
9449	Re: Yogi-isms	Baseball	53%

Re: Octopus in Detroit?
From: *georgeh@gjhsun (George H)*

Harold Zazula <DLMQC@CUNYVM.BITN

>I was watching the Detroit-Minnesota [illegible] ht and thought I saw an
>octopus on the ice after Ysebaert scor [illegible] game at two. What gives?
>(is there some custom to throw octopus [illegible] ice in Detroit?)

It is a long standing good luck Redwing's tradition to throw an octopus on the ice during a Stanley Cup game. They say it dates back to '52 at the Olympia when the Wings became the 1st team (I think) to sweep the cup in 8 games. A lot hardet to throw one from Joe Louis seats than from the old Olympia balcony, though.

Funniest I ever saw was when some Tiger fans threw one on the field during a Detroit/Toronto baseball game ... I was living in California and the folks I was watching with had never heard of hockey and were incredulous when I recognized the octopus BEFORE the camera closeup !!

Why Hockey?

Part 1: Imp[illegible]ords
This message has [illegible]tant words about Hockey [illegible] Baseball

baseball hockey stanley tiger

The difference makes the computer think this message is 2.3 times more likely to be about Hockey than Baseball.

AND

Part 2: Folder size
The Baseball folder has more messages than the Hockey folder

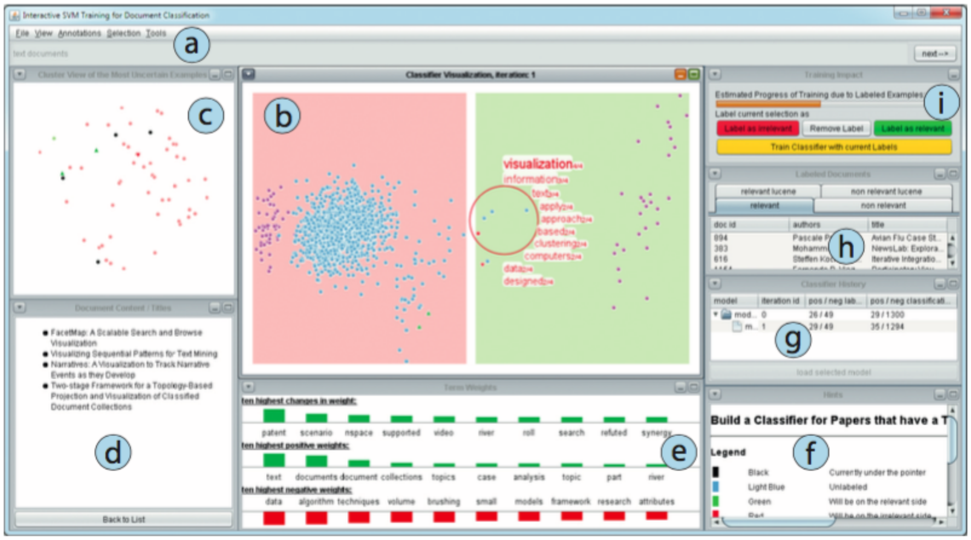
Hockey:	7
Baseball:	8

The difference makes the computer thinks each Unknown message is 1.1 times more likely to be about Baseball than Hockey.

These are all of the words the computer used to make its p[illegible] more).

Figure 2.7: EluciDebug: Using Multiple Views to Explain

Kulesza's EluciDebug[17] uses different views(Figure 2.7) to explain to novices how the classifier makes each prediction. It helps them build a mental model of the classifier by highlighting the features in each instance, showing them the overview of features and prediction, and showing them how their modification of the system changes the result. The system supports the interaction of labelling instances, picking up features and adjusting their weights. Their experiments show that users can build a better mental model and create better classifiers using the system, and users feel the system is accessible and are confident using it. However, in the user study, Kulesza only measures the mental model of how the classifier makes a prediction and how to make the classifier perform better, which are under the assumption that a workable classifier can be built. He does not examine whether users can realize the weakness of the classifier and the dataset, so that they can switch to a better strategy to solve their problem at an early stage.



Estimated Progress of Training due to Labeled Examples:

Label current selection as

doc id	authors	title
894	Pascale P…	Avian Flu Case St…
383	Mohamm…	NewsLab: Explora…
616	Steffen Ko…	Iterative Integratio…

model	iteration id	pos / neg lab…	pos / neg classificati…
mod…	0	26 / 49	29 / 1300
m…	1	29 / 49	35 / 1294

- FacetMap: A Scalable Search and Browse Visualization
- Visualizing Sequential Patterns for Text Mining
- Narratives: A Visualization to Track Narrative Events as they Develop
- Two-stage Framework for a Topology-Based Projection and Visualization of Classified Document Collections

Build a Classifier for Papers that have a T

Legend

■	Black	Currently under the pointer
🟦	Light Blue	Unlabeled
🟩	Green	Will be on the relevant side
🟥	Red	Will be on the irrelevant side

Figure 2.8: An Interactive Desktop for Visual Classifier Training

2.2.3 Interactive Desktop for Visual Classifier Training

Heimerl et al.[11] implemented a system(Figure 2.8) to improve the efficiency of active learning. The system includes a 2D scatterplot showing how each instance is distributed according to the features and the classifier's prediction. The horizontal axis of the 2D plane is the confidence that the instance is classified as a positive example, and the vertical axis represents the diversity of the documents closest to the decision boundary. Whenever the user labels an instance, the view updates to reflect the current state of the classifier. The users can also choose the instances they think might be useful. Users can select an instance or an area of instances to view their feature distributions and then label them. Hovering in the interface over input documents, it displays at most ten terms sorted according to their document frequency. The system also has a Cluster View, which puts special emphasis on representing document similarity. A progress bar is displayed at the top right corner informing analysts about the impact of their labelling. However, according to experiments, their system did not improve active learning efficiency. Users need more time to understand how to use the system, what each view means, and some views were not used by some participants. Heimerl et al. also did not measure users' understanding of the dataset and classifier. Our aim is to show that a similar system can help people identify classifier's performance at early stage. We also hope to see that such a system would improve the user experience in labelling.

2.2.4 Summary

Transparent machine learning systems improve the efficiency and experience for developers to build a classifier. They also lower the barrier for the end users to understand and personalize their system. These systems give us some experience on how to tackle visualization problems: We could show the multi-dimensional data in limited space, support effective interaction, help end users build mental model by explaining the rationale of the classifier, and avoid overwhelming the users with too much information by only visualizing the important features, among others.

However, these systems are designed for developers who want to quickly build a classifier using existing models, and for novices who want to understand the reasoning behind an already-built classifier so that they can better adjust it to their own problem. Our research aim is to support novices who want to quickly prototype a new classifier for a new task. We hope to help novice machine learning users quickly understand the weakness of the classifier and the dataset so that they can select the best strategy to approach their problems.

2.3 Tools for Labellers

To aid novice users in applying machine learning, we focus on the labelling task. Today, machine learning has matured to a point where we have many theories and algorithms to solve different kinds of problems. The bottleneck is the quality and quantity of labelled training data, a time-consuming and costly process in the pipeline. Researchers have done work to improve the quality of labels, or to reduce the number of labels needed by the classifier. This includes work in structured labelling (to improve quality) and active learning (to reduce labels needed).

2.3.1 Structured Labelling

When labellers label a large amount data for a long time, *concept evolution* may happen. They may label the same data in a different way in a later stage. This phenomenon will affect the quality of their labels and introduce noise to the classifier. Kulesza et al.[16] tried to solve this problem with *structured labelling*, where users can group their data based on their concept. This is extremely helpful for the data they are unsure about, so they could revisit them later and follow the same rule to label them. Result from user study shows that structuring was preferred by participants and helped them label more consistently.

2.3.2 Active Learning

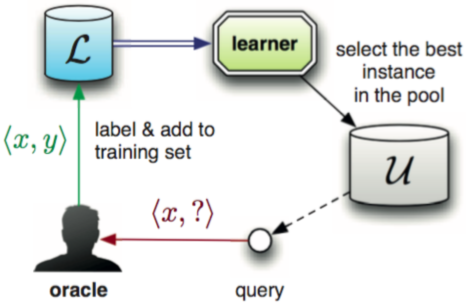


Figure 2.9: Active Learning selects the best instance to query for label

In many modern machine learning problems, data may be abundant but labels are scarce or expensive to obtain. Such problems include speech recognition[43], information extraction[36], and computational biology[39]. Active learning attempts to solve this problem by selecting the unlabelled instances for the labellers to label. The instances are selected according to their informativeness calculated by the backend algorithm. In this way, the system can still achieve high accuracy with few labelled instances, thereby minimizing the cost of obtaining labelled data, as is shown in Figure 2.9.

Active Learning is a field that has been well studied (see [34]). Depending on the dataset, we have different frameworks to query the label, such as uncertainty sampling[18], query-by-committee[37][8], expected error reduction[30] and density-weighted methods[35], among others. In this work, we use a density-weighted method to select the instances for the labellers. Chapter 4 will go into the detail of our framework.

2.3.3 Summary

While active learning can reduce the labeller's workload, and structured labelling can improve the quality of the labels, it cannot help improve the labeller's understanding or user experience. During the labelling process, the labellers, who have to constantly respond to the queries from the system by providing labels, are still in a passive mode. Their motivation may be to finish the task as soon as possible so that they get paid, or to start building the classifier. Boredom, distraction, and irritation with the task can set in, which will affect their efficiency as well as the quality of the labels.

Rather than reducing the quantity of labels required, our work focuses on improving the quality of labeller's user experience. We hope a well-designed visualization system can help users gain insight into their data and classifier, see the effect of each new label, get motivated by their inner curiosity about the data, and experience a sense of achievement as classifier performance improves. In this way, not only the system is doing active learning, but the labellers are also actively labelling and learning. Understanding the complicated math is not required; users can understand much about the classifier just by peeking at the visualizations while doing the unavoidable labelling needed to create their machine learning system.

Chapter 3

Taxonomy for the Machine Learning Visualizations

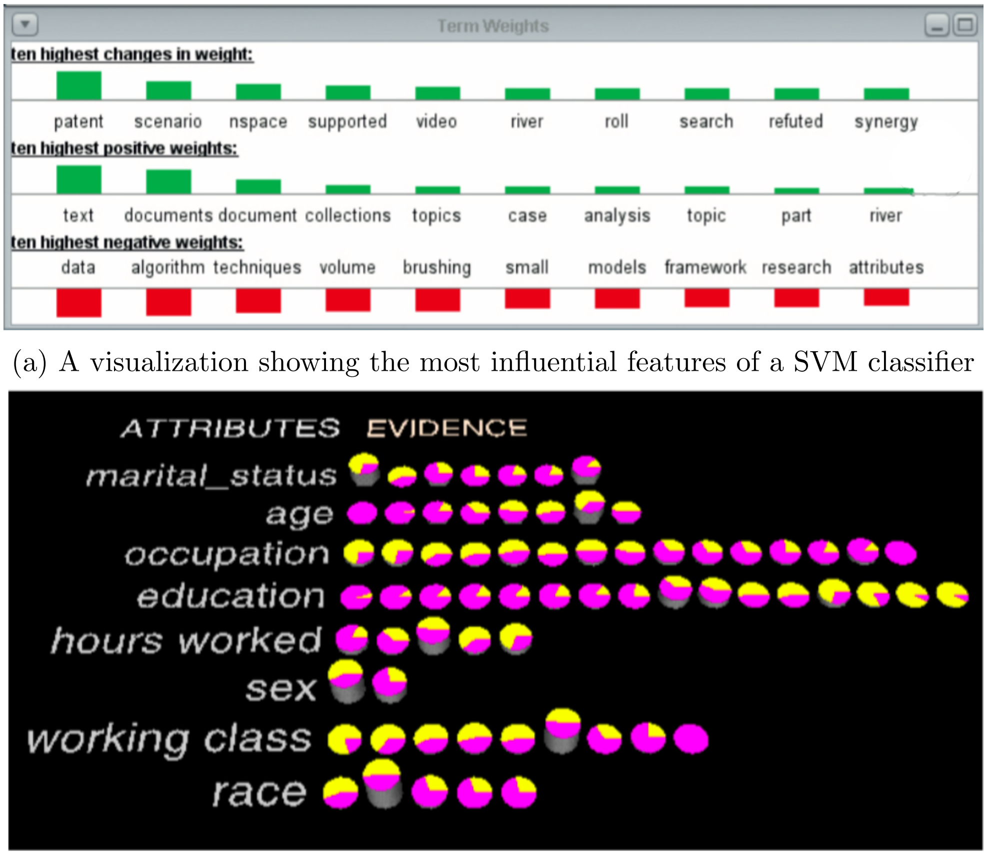
In this chapter, we define a taxonomy for existing machine learning visualizations. The taxonomy is further generalized in two dimensions: data displayed and advanced features. This taxonomy seeks to guide classifier builders or visualization experts to design better visualizations that meet their requirements. In the final section, we provide suggestions on how to choose the right visualization based on target users and design goals.

3.1 Data Displayed

3.1.1 Selected Information vs All Information

The dataset for a machine learning problem can be as huge as terabytes, or as small as kilobytes. But even with the large screen interaction, we still cannot show all the data in a single visualization: We have the choice of showing only selected data with detail, or the whole dataset when we have enough space.

Figure 3.1a from [11] shows a visualization for an SVM document classifier's features. The features for this problem are the dataset's whole vocabulary, which cannot fit into the graph. The system shows the top ten most interesting data in three sets: most changed, most positive, and most negative. This strategy usually shows the most informative data, helping users quickly locate the influencing factors, or skeleton of the classifier. Another strategy is to show the sampled data, so we can preserve the overall distribution of the



(a) A visualization showing the most influential features of a SVM classifier

(b) A visualization showing all features of a naive bayes classifier

Figure 3.1: Selected Information vs All Information

data. When the size of the data we want to show is small, we can put all data in the visualization. As in Figure 3.1b from Becker et al.'s work[4], the naive Bayes classifier has limited features and feature values, which can be visualized in one graph. The influencing data still stand out since they have the larger area or brighter color. Users can identify the less influencing features and consider removing them from the classifier.

This branch allows users to go beyond the dataset size limitation when designing visualizations. When the dataset size is too large to fit in the screen, we can always show the most representative data, whether by random sampling or by frequency ranking.

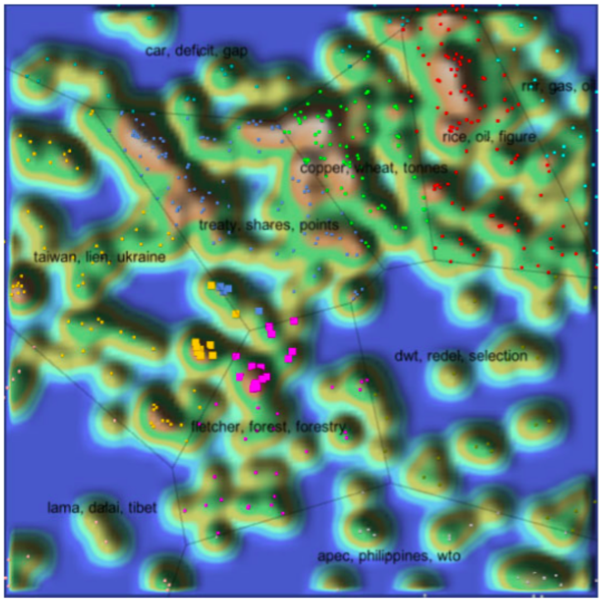
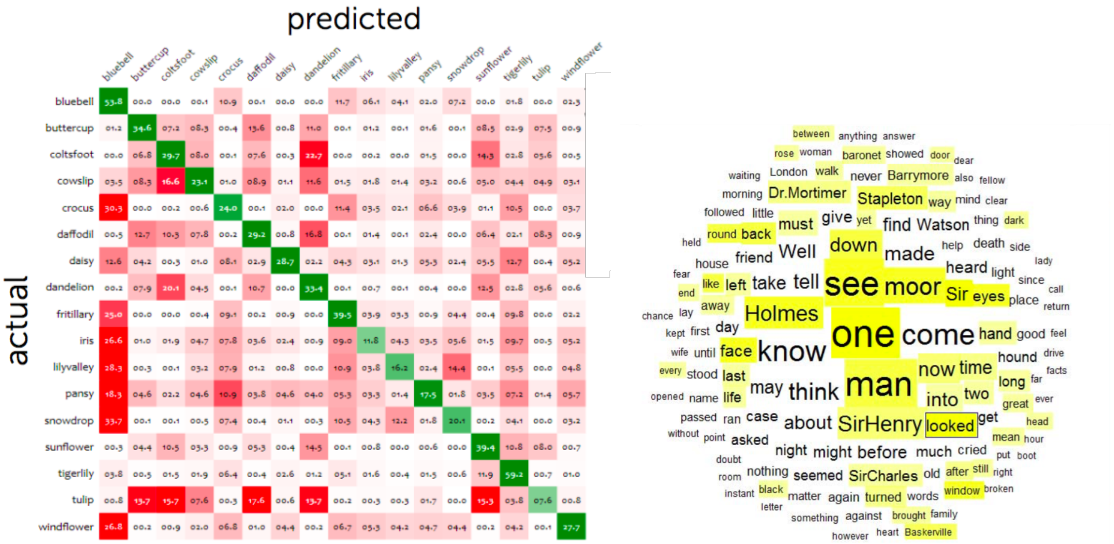


Figure 3.2: Document Clustering

3.1.2 Labelled Data vs Unlabelled Data

In most cases, classifiers are built on labelled data by supervised machine learning, and most visualizations serve to inspect the classifier's performance. However, when the cost of labelling is high, we may want to have a rough understanding of the data before spending effort on labelling. It is useful to know whether there are certain patterns in the data or whether they can be clustered. Thanks to unsupervised learning, we have techniques for clustering such as k-means and hierarchical clustering[29]. Figure 3.2 from Seifert et al.'s work[33] shows a visualization for documents clustering. Documents are represented as points on a 2D plane. They are clustered according to their topic, and the polygonal areas are generated by the Voronoi area subdivision. Regions populated by a large number of topically related documents are represented as hills to stand out on the graph.

This branch gives users the opportunity to visualize the data even before they begin labelling. Users can determine whether the data has certain patterns or clusters by visually exploring the dataset.



	bluebell	buttercup	coltsfoot	cowslip	crocus	daffodil	daisy	dandelion	fritillary	iris	lilyvalley	pansy	snowdrop	sunflower	tigerlily	tulip	windflower
bluebell	53.8	00.0	00.0	00.1	10.9	00.1	00.0	00.0	11.7	06.1	04.1	02.0	07.2	00.0	01.8	00.0	02.3
buttercup	01.2	34.6	07.2	08.3	00.4	13.6	00.8	11.0	00.1	01.2	00.1	01.6	00.1	08.5	02.9	07.5	00.9
coltsfoot	00.0	06.8	29.7	08.0	00.1	07.6	00.3	22.7	00.0	00.2	00.0	01.5	00.0	14.3	02.8	05.6	00.5
cowslip	03.5	08.3	16.6	23.1	01.0	08.9	01.1	11.6	01.5	01.8	01.4	03.2	00.6	05.0	04.4	04.9	03.1
crocus	30.3	00.0	00.2	00.6	24.0	00.1	02.0	00.0	11.4	03.5	02.1	06.6	03.9	01.1	10.5	00.0	03.7
daffodil	00.5	12.7	10.3	07.8	00.2	29.2	00.8	16.8	00.1	01.4	00.1	02.4	00.0	06.4	02.1	08.3	00.9
daisy	12.6	04.2	00.3	01.0	08.1	02.9	28.7	02.2	04.3	03.1	01.3	05.3	02.4	05.5	12.7	00.4	05.2
dandelion	00.2	07.9	20.1	04.5	00.1	10.7	00.0	33.4	00.1	00.7	00.1	00.4	00.0	12.5	02.8	05.6	00.6
fritillary	25.0	00.0	00.0	00.4	09.1	00.2	00.9	00.0	39.5	03.9	03.3	00.9	04.4	00.4	09.8	00.0	02.2
iris	26.6	01.0	01.9	04.7	07.8	03.6	02.4	00.9	09.0	11.8	04.3	03.5	05.6	01.5	09.7	00.5	05.2
lilyvalley	28.3	00.3	00.1	03.2	07.9	01.2	00.8	00.0	10.9	03.8	16.2	02.4	14.4	00.1	05.5	00.0	04.8
pansy	18.3	04.6	02.2	04.6	10.9	03.8	04.6	04.0	05.3	03.3	01.4	17.5	01.8	03.5	07.2	01.4	05.7
snowdrop	33.7	00.1	00.1	00.5	07.4	00.4	01.1	00.3	10.5	04.3	12.2	01.8	20.1	00.2	04.1	00.0	03.2
sunflower	00.3	04.4	10.5	03.3	00.9	05.3	00.4	14.5	00.1	00.8	00.0	00.6	00.0	39.4	10.8	08.0	00.7
tigerlily	03.8	00.5	01.5	01.9	06.4	00.4	02.6	01.2	05.1	01.6	00.4	01.5	00.6	11.9	59.2	00.7	01.0
tulip	00.8	13.7	15.7	07.6	00.3	17.6	00.6	13.7	00.2	00.3	00.3	01.7	00.0	15.3	03.8	07.6	00.8
windflower	26.8	00.2	00.9	02.0	06.8	01.0	04.4	00.2	06.7	05.3	04.2	04.7	04.4	00.2	04.2	00.1	27.7

(a) A Confusion Matrix from Prospect[25]

(b) Word Cloud Explorer[12]

Figure 3.3: Classified vs Unclassified Data

3.1.3 Classified Data vs Unclassified Data

Another dimension of the data is whether the data has been classified or not. Classified instances are always shown with their original labels for users to compare and locate the mistakes the classifier makes. A confusion matrix(Figure 3.3a) is a representative example. One of the two axes (horizontal and vertical) represents the original class and the other represents the predicted class. All the instances not on the diagonal are misclassified. Users can quickly find out classes and the instances that the classifier cannot correctly distinguish, and seek ways to solve these classifier errors. Visualizations for unclassified but labelled data also aims at showing the feature distribution in a same class, and the comparison among different classes, helping users further understand the dataset and quickly generate useful features before actually implementing the classifier. Figure 3.3b shows a word cloud that can be applied to both labelled and unlabelled data. Users can easily identify the most influential words in the dataset.

By studying this branch of taxonomy, users could realize the opportunities to find features with visualizations on both classified and unclassified data. They could find the first several features using the unclassified data, and discover more features once they could examine the result of the classified data.

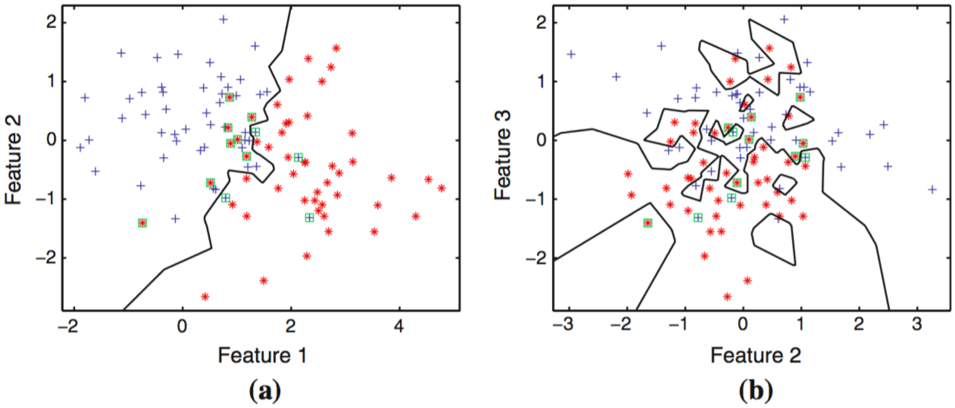


Figure 3.4: Different ways to project a 10-dimensional dataset onto 2-dimensional space

3.1.4 Feature-Centric vs Instance-Centric

Visualizations on instances help us better understand the dataset, while visualizations on features help us better understand the classifier. For example, Figure 3.2 and Figure 3.3a are instance-centric, as the finest element on the graphs represent an instance. Figure 3.1a and Figure 3.1b are feature-centric; they inform us what the classifier relies on to make predictions. Viewing these features can tell us whether the classifier is overfitted or does not yet have enough knowledge about the data.

Traditionally we consider data as each instance in the dataset, but data can also be the features extracted from the dataset. These feature-centric visualizations highlight the importance of features. They provide additional information on the internal reasoning of the classifier.

3.1.5 Dimensionality Reduction

The bottleneck of the visualization is not only the size of the dataset, but can also be the length of the feature vector, which is the number of features. When we want to visualize the instances distribution across the feature values and we have more than three features, there is no way to display the multi-dimensional data in a single visualization. To tackle this problem, we have to select the features that are important, or perform dimensionality reduction on the data to combine several features into one. Research exists in feature engineering on this topic, such as Principle Component Analysis[14]. Figure 3.4 from

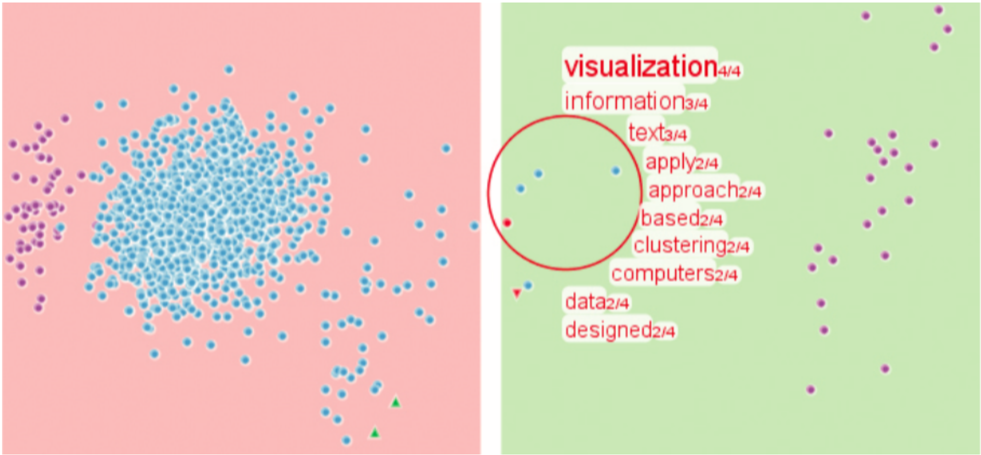


Figure 3.5: An interactive visualization where user can circle an interesting area and view the summary of that area

Migut et al.'s work [20] shows a 10-dimensional dataset of two Gaussian distributed classes for the SVM projected to 2D space. The visualization shows that the decision boundary separates the data well.

With dimensionality reduction, users can design instance-centric visualizations even when the data has many dimensions. Dimensionality reduction places the data on a 2d plane, and gives users a perspective on how the data is distributed and how the features are correlated.

3.2 Advanced Features

In the previous section, we discussed ways to show different kinds of data most effectively. In this section, we will introduce the advanced features added to the traditional visualizations to enhance understanding, make them interactive, and highlight useful information.

3.2.1 Supporting Interaction vs Static View

Interactivity is crucial for building visualization tools that handle complexity [22]. Visualizations are not only static graphs or plots; they are also visualization systems. For a specific task, we may design a visualization system that allows users to perform more specific actions to understand data in more detail. This is the overview first, zoom and filter, then detail-on-demand strategy[38]: view the data in general, find an interesting area, and explore the interesting area. A real-life example is Google Map. Unlike traditional paper maps that either show the overview of an area without detailed street, or shows only a small area with every road, Google Map allows users to see the overview of the city first, then enlarge the area in which they are interested. They can also click on the shops or restaurants to view detailed information. Figure 3.5 in Heimerl's system[11] is an example of an interactive visualization. Users can click on a point to view the data it represents, or circle an area to see the summary of the points inside. However, if all the information is equally important, and there is enough space to show them all, we could simply use the static view so that users can easily get all the information without the interaction method. Whether to include interaction in the system depends on the amount and the granularity of the data we want to show.

Using interaction techniques, we can visualize large datasets without losing information. Users can explore the dataset by viewing both the overall distribution and detailed information. However, there is a significant cost to the implementation of interactivity.

3.2.2 Data Abstracted as Shape vs Data Shown as it is

In visualizations, data can be shown as abstract points or as it is. When we have large amount of data and the goal is to see how the data are distributed, we can use scatter plots, bar graphs or line charts that abstract the data as shapes, as shown in Figure 3.5 and 3.2. In this condition, distributions and statistical numbers are the important information we care about and are used to choose the best data processing strategy. When we have dozens of data, or all the data can be represented as dozens of data which need to be studied thoroughly, it is more important to show the original data and let the users explore them as much as they can. For a machine learning problem, the scale of the dataset is usually thousands, millions, or billions, but the feature values may be limited to dozens. A thorough study of the features can give us a quick and general understanding of the properties of the real data. The nomogram in Figure 3.6, the pie chart in Figure 3.1b, and the word cloud in Figure 3.3b are all examples of visualizing the concrete features.

In visualizations, data are not necessarily abstracted as shapes. We can directly display the original data on the screen if we want to quickly know what the representative data look like, whether they are images or pieces of text.

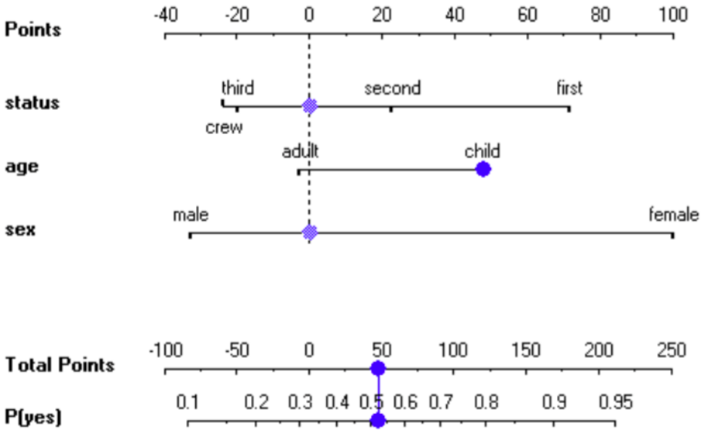


Figure 3.6: A nomogram for prediction of survival probability of a passenger on HMS Titanic [21]

3.2.3 Prediction Confidence

The prediction confidence is the probability calculated by the system of the data being classified as one class. When there are only two classes, the confidence can be easily visualized as a point on a line, e.g., 70% positive and 30% negative. When there are multiple classes, we can project it to a 2D radar chart, and the closer the points are to the center, the less confident is the system's prediction. Figure 2.8 is an example of 2-class confidence visualization, and Figure 3.7 from Seifert and Granitzer's work [32] is an example of multi-class confidence visualization. The information about confidence can tell users how much they can trust the system. A 99% accurate classifier where most data have confidence of 60% is less trustful than a classifier with the same accuracy, but most of the data have confidence of 90%.

Confidence can tell more about the classifier's performance than simple accuracy. The data that the classifier is uncertain about can stand out in the visualization. Users can also know whether there is still room for improvement. However, adding this information may result in information overload. If the visualization keeps updating, or users have difficulty understanding the concept of confidence, this information can only distract users from their task. Designers should take user's expertise and time into account when adding this information.

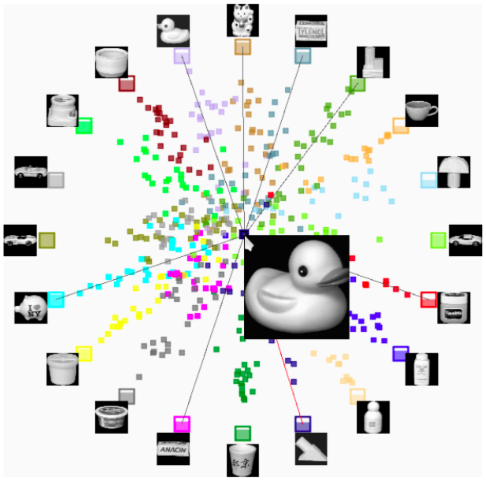
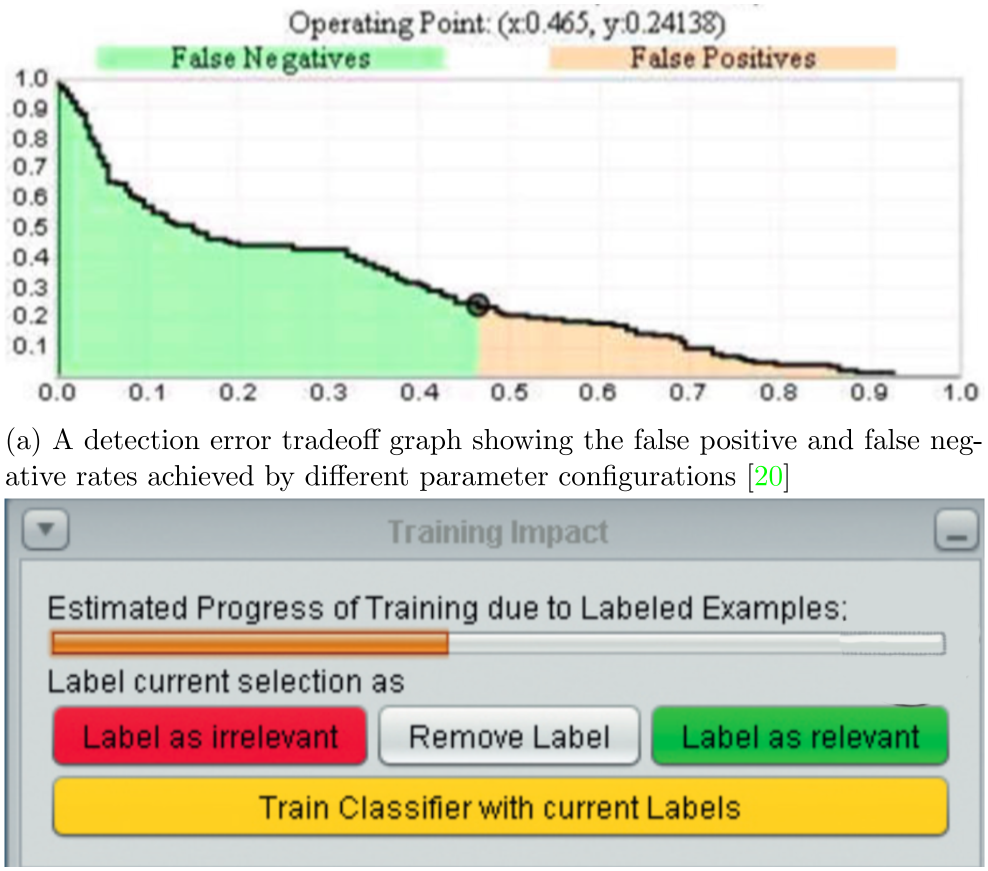


Figure 3.7: A multi-class visualization for image classification

3.2.4 Classifier's Assessment

All visualizations are tools for users to assess the current classifier. A visualization that shows the actual score of the classifier's performance can make this process even more convenient by helping users focus their attention only on the results and numbers. Those visualizations showing confidence only provide an indirect way to assess the classifier's performance. Figure 3.8a shows a detection error tradeoff (DET) graph that indicates how the parameters could affect the false positive and false negative rate, and users can directly select the desired score pair and tune the parameter to achieve that. The receiver operating characteristic (ROC) curve is a more popular alternative that plots true positive against false positive. Figure 3.8b is a progress bar that shows the margin size of the SVM model so that users can understand how their classifier performs with just one peek at the visualization.

If users have more time to explore the visualization, simply showing the confidence is enough to describe the classifier's performance. If users just want one number representing the performance, it is better to directly show the score. When considering this feature, designers should look at the goal of the visualization users.



(a) A detection error tradeoff graph showing the false positive and false negative rates achieved by different parameter configurations [20]

Estimated Progress of Training due to Labeled Examples:

Label current selection as

(b) A progress bar for an SVM classifier [11]

Figure 3.8: Classifier's Assessment

3.2.5 Highlighting Interesting Information

When the task is to identify interesting information, highlighting in the visualization can be a useful feature. Figure 3.5 highlights the data whose predicted class has changed in red, so that the user understands the influence of his label, and can examine the reason why classification changed. Figure 3.9 highlights the points closest to the decision boundary so that users can quickly focus on the instances that the classifier is unsure about. Whether to highlight data, and what data to highlight depends on the actual task. If the task is to explore, the visualization should allow the users see as much data as possible. If the task is to identify, the visualization should help users quickly focus on salient data.

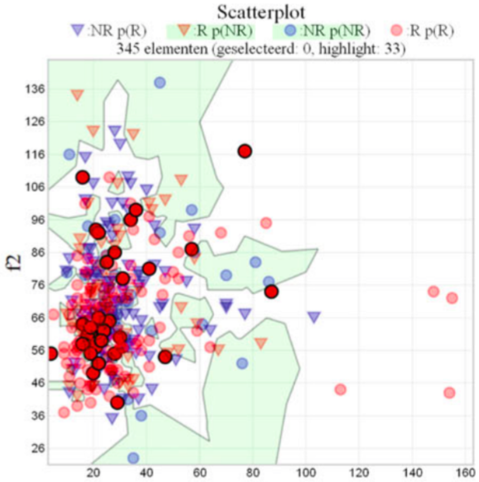


Figure 3.9: Highlighting the points closest to the decision boundary in 2D space[20]

When considering this visualization feature, designers must be clear about the goal of the visualization: to help explore or to help identify. Adding this feature can also create a more user friendly interface, reduce the effort of finding the target in the visualization, and reduce the chance of missing a target.

3.3 Goal

The first two questions one must answer before designing a visualization are: who will use it, and why do they want to use it. Understanding these two questions is fundamental to choose what data to show and how to show them.

3.3.1 Target Users

The first question to answer is who will use the visualization. We posit that there are two kinds of users that may be interested in machine learning visualizations: machine learning novices and experts.

Novices

As we note in 1.3, novices are practitioners who want to build a classifier, but only know how a machine learning classifier works generally, and may not know the detailed mathematics. We need to consider novices' limited technical background when designing visualizations for them, so it is better to avoid using too many academic terms and processed data. Natural language and examples of the actual data or features are usually used to make the classifier more approachable to people. The channels to show the data should also be simple and generally easy to understand, such as the traditional scatter plot, bar graph, and pie chart to lower their effort to understand them. In Kulesza's system [17], we can see several demonstrations of the principle. Figure 2.7-D uses natural language and the real features to explain a the classifier's prediction. And Figure 2.7-F shows the current classifier's feature values using a bar graph and mark each feature on the axis.

Experts

Alongside novices, we also note machine learning experts in 1.3. They are users who know the mathematics and rationale of the machine learning classifier such that they can program a classifier themselves. Since experts who know all the terminologies and can understand the abstract data, we have more options to avoid using our limited space for explanation and displaying real data, but help them gain insight about the whole dataset in the shortest time. So our goal would be showing them as much as information, and allowing them to explore their interest in detail. These can be achieved by abstracting the data with channels such as points or even pixels to save space, using dimensionality reduction or other statistical values to show only the interesting data points and aggregated results, and use the overview first, zoom and filter, then detail-on-demand strategy to let them explore their interest. For example, ROC curve uses true positive rate and false positive rate to illustrate the performance of the classifier with different threshold. Figure 3.5 abstracts the documents as points and focus-plus-context strategy to save the space and let developers explore the data in detail.

3.3.2 Task Goal

The second question (beyond users) involves task. Munzner[22] summarizes three actions that define user goals for the general use of visualizations: analyze, search and query. In this thesis, however, the goals for machine learning tasks are more specific, and we generalize them according to the process of machine learning pipeline. They are data labelling, feature engineering, model selection and parameter tuning.

Data Labelling

Traditionally, the data labelling process is considered as time and effort that must be invested before actually building the classifier. It can be performed by anyone and is not related to getting insight into the data. The only research to improve the labelling experience is Heimerl's system. Here he presents two visualizations for the labellers: a scatterplot(Figure 3.5) for them to choose the next data to label, and a progress bar(Figure 3.8b) indicating how much they have done. In this thesis, we emphasize utilizing the labelling process and Chapter 4 introduces our work on improving the labelling experience.

Feature Engineering

Feature engineering can be generalized as four parts [5]: brainstorm features, devise features, select features, and evaluate models. The evaluation of models can motivate us to brainstorm more features, forming an iterative loop.

The first set of features comes from an initial brainstorm, which is our general perception of the data. This general perception can be the expert knowledge about the problem when it has been well studied, or a summary based on the data's distribution. As we have no classifier built yet, we can simply use visualizations for unclassified data to find the clusters and view its distribution to discover the features.

Once we have possible features brainstormed, programmers can be employed to devise and extract features. Although many algorithms [9] can be employed to select features, using visualizations to show the feature's performance can tell the users which ones are influencing the classifier's decisions and which ones are confusing the classifier's decisions. For this task, we could use feature-centric visualizations to focus on a feature's performance. Dimensionality reduction may also be needed to see the data distribution across different features when the feature vector is long.

Finally, we need to evaluate the features and identify deficiencies. To support this, the visualization system should help users discover uncertain areas where the classifier is likely to make mistakes, as well as provide interaction for users to inspect the data in detail. To achieve this task, it is better to use visualizations that have labelled data, support interaction and show the prediction confidence. Figure 3.10 is an example of helping users

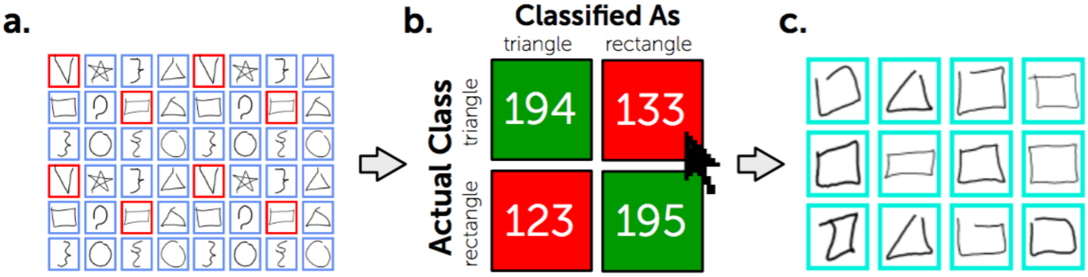


Figure 3.10: Gestalt [24]: Finding features from confusing instances

discover features. Users can click on the mistakes in the confusion matrix to identify the features that could help distinguish the two classes.

Model Selection by Evaluation

Suppose we already have a set of valuable features for the classifier. The next step is to select the best model that can make the best use of the features. Today we have a number of toolkits and libraries that contain different machine learning models and algorithms to choose from. To analyze the efficacy of these existing models, all we need is to compare their performance and select the best. To best evaluate the classifier's performance, we need labelled and classified data, prediction result and the confidence, and the whole dataset if resources allow. This information can help us see which model best aligns with the data's underlying structure by looking at which model can classify the data with the highest confidence.

Parameter Tuning by Evaluation

At this time, we may have already focused on one model and a set of features according to the data or feature's properties. The last thing is to tune the parameters. Similar to model selection, we need to compare the performance of different configurations so we need labelled and classified data, prediction result and the confidence, and ideally the whole dataset. Different from model selection, we have unlimited configurations of the parameters, so it is impossible to compare them all at the same time. One solution is to simply show the trend line of the accuracy across different configurations as in the DET curve in Figure 3.8a so that users can choose the best configuration by the score. Another solution is to allow users to explore different configurations by supporting interactions.

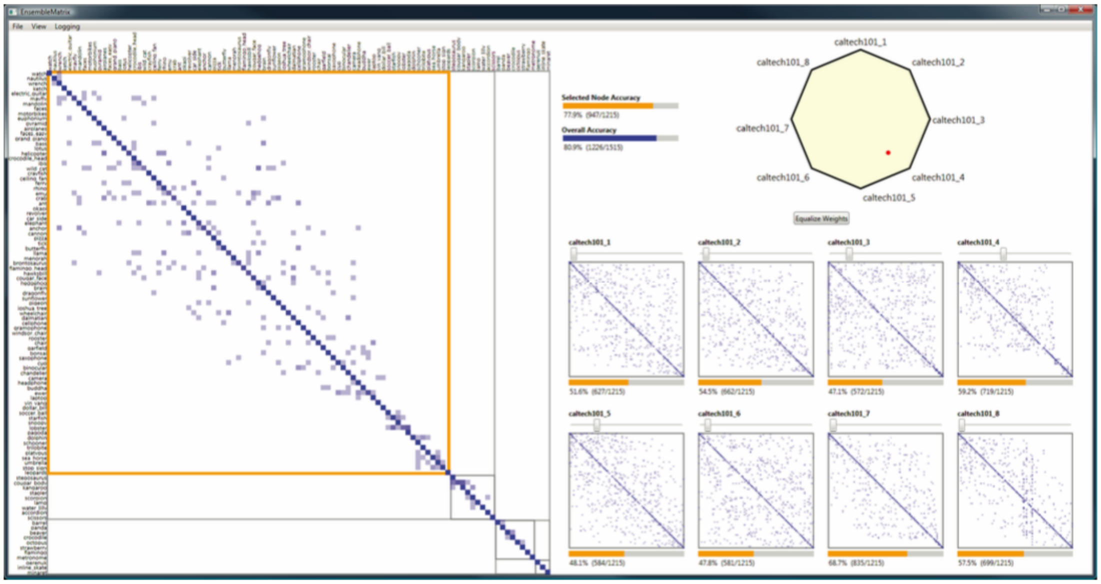


Figure 3.11: Ensemble Matrix allows users to build an ensemble classifier(left) by adjusting the weight of each component classifier(right).

This helps users see how each possible configuration affects the classifier's decision. It also helps users feel control over the classifier's performance. Talbot et al.'s Ensemble Matrix [40] (Figure 3.11) allows users to try different configurations to best assemble different classifiers.

3.4 Summary

This chapter summarizes previous machine learning visualizations by defining a taxonomy and tagging previous visualizations with the taxonomy. There are two dimensions in the taxonomy: the data displayed, focusing on the visualization's content; and the advanced features, focusing on the techniques to make the visualization usable. In the last section, we also provide a guide to design visualizations based on target users and design goals. We leverage this taxonomy to design our Label-and-Learn system, which we describe in Chapter 4.

Chapter 4

Label-and-Learn

One task necessary to develop any machine learning system is the provision of labelled data, i.e. data that has been pre-classified typically by a person who examines and labels data to generate a sufficient dataset for the classifier to learn from. Data labelling is tedious, yet it is an unavoidable task in building a machine learning classifier. As well, it can be expensive, with a cost either in designer/user's time (if the developer labels data) or in money (if paying someone else to label data). On the other hand, we believe that labelling can serve as the first opportunity for developers to get insight into their dataset.

Label-and-Learn is a system designed for machine learning data labellers who want to train a machine learning algorithm to solve a specific problem without involvement of a machine learning expert. It can users get a better understanding of the data while labelling. In this chapter, we will introduce the system by describing the specific kind of labelling tasks addressed, the machine learning model used, the traditional interface for labellers and our innovative visualization interface.

4.1 Task

The machine learning task our system helps to solve is Named Entity Recognition [23]. More specifically, given a string that matches one of the named entities, the classifier should identify whether it actually refers to the named entity. For example, the classifier should identify "US" refers to "United States" in the context "I'm studying computer science in the US", but not in the context "He teaches us English". The classifier uses the features from the context of the matched string: the two words before the matched string, the matched string itself, and the two words after the matched string. The features in the second context are “he”, “teaches”, “us”, “english”, and empty.

There are several reasons that we choose Named Entity Recognition as our task: (1) It is used in many real-world systems (identifying date, time and location in an email, or identifying celebrities, companies, events in the news), (2) It usually requires a large amount of human labelling work that is boring and error-prone, (3) The features are easy to identify and understand (unlike features in audio or image data that are numbers or vectors that are only understandable by a computer).

4.2 The Backend Classifier

The idea of a Label-and-Learn system is to help users understand the data, as well as help them predict whether a satisfactory classifier can be built on the current dataset and features. To achieve that, we implemented a classifier in our system's backend to give our users some information on the performance of the baseline method: where it is good, and where it will fail.

4.2.1 Naive Bayes classifier with Bag-of-Words Model

We choose naive Bayes as our model because it has competitive performance in text categorization, and it is easy to implement, understand and visualize compared to other methods such as SVM. Using the traditional naive Bayes and bag-of-words model to classify a document, the order of the words in the text does not affect the decision. As our problem is Named Entity Recognition, we add a small modification: In our model, we have the prior probability, which is the ratio of positive instances in the training data, and three sets of dictionaries:

- pevious dictionary, containing the tokens before the matched string,
- current dictionary, containing the strings of named entities, and
- next dictionary, containing the tokens after the matched string.

To identify whether the string t_3 is actually the named entity in the context $[t_1][t_2][t_3][t_4][t_5]$, the system uses the following formula to calculate its positive probability:

$$p(+|[t_1][t_2][t_3][t_4][t_5]) = \frac{p(positive)}{p(positive) + p(negative)} \tag{4.1}$$

where:

$$p(positive) = p(+) \prod_{1\le i\le 5} p(t_i|+) = p(+) \prod_{1\le i\le 5} \frac{count(t_{i,+}|set_i)}{count(pos|trainingset)} \tag{4.2}$$

$$p(negative) = p(-) \prod_{1\le i\le 5} p(t_i|-) = p(-) \prod_{1\le i\le 5} \frac{count(t_{i,-}|set_i)}{count(neg|trainingset)} \tag{4.3}$$

Here, $p(+)$ is the ratio of positive instances in the training set, $count(pos|trainingset)$ is the number of positive instances in the training set, $count(t_{i,+}|set_i)$ is the number of t_i's positive occurrences in set_i. Negative probabilities use a similar representation. Specifically, $set_1 = set_2 =$previous dictionary, $set_3 =$ current dictionary, $set_4 = set_5 =$next dictionary. We use three different sets to emphasize the order of the words, as the terms that are more likely to appear before the named entity and the terms that are more likely to appear after the named entity might be different.

4.2.2 Active Learning

As mentioned in section 2.3.2, Active Learning is an effective framework to reduce a labeller's workload. It requests the label of the datum that has the highest information gain, quickly reducing the system's uncertainty. Our Label-and-Learn system also applies this framework to improve the labelling experience.

Uncertainty Sampling is the simplest sampling method in active learning. It selects the data that is closest to the decision boundary since that is the most uncertain one. However, due to the text data's long-tail property, many feature values appear only once or twice in the training set; labelling the data containing them may not bring as much influence as labelling data containing a high-frequency feature value.

Instead, we choose a density-weighted method for the system, which uses the intuition that informative instances should be those with informative content as well as those that are representative of the underlying distribution, as described in Chapter 5 of [35]. Each time the system selects the data x^*_{ID} according to the criterion below:

$$x^*_{ID} = \underset{x}{\operatorname{argmax}}\, \phi(x) \times \left(\frac{1}{U} \sum_{x' \in U} \operatorname{sim}(x, x') \right)^{\beta}$$

Specific to our task, U is the size of unlabelled data, we set β as 1,

$$\phi(x) = H(x) = -p(+|x)\log(p(+|x)) - p(-|x)\log(p(-|x))$$

sim(x, x') is the number of same features that appear in the same place in the two instances.

$$\text{sim}(x, x') = \sum_{1 \leq i \leq 5} \delta_{t_{i,x}, t_{i,x'}}$$

$$\delta_{t_{i,x}, t_x i, x'} = \begin{cases} 1 & \text{if } t_{i,x} = t_{i,x'} \\ 0 & \text{if } t_{i,x} \neq t_{i,x'} \end{cases}$$

Then we have

$$\sum_{x' \in U} \text{sim}(x, x') = \sum_{1 \leq i \leq 5} count(t_{i,x} | set_i)$$

which is the sum of the number of occurrences for each term in the set of its position.

We implemented the dictionaries as a HashMap, so both $\phi(x)$ and $\sum_{x' \in U} \text{sim}(x, x')$ take $O(1)$ to compute, it takes $O(n)$ to select the maximum datum for labelling, which is an acceptable time complexity for datasets as large as 1 billion.

4.2.3 Evaluation

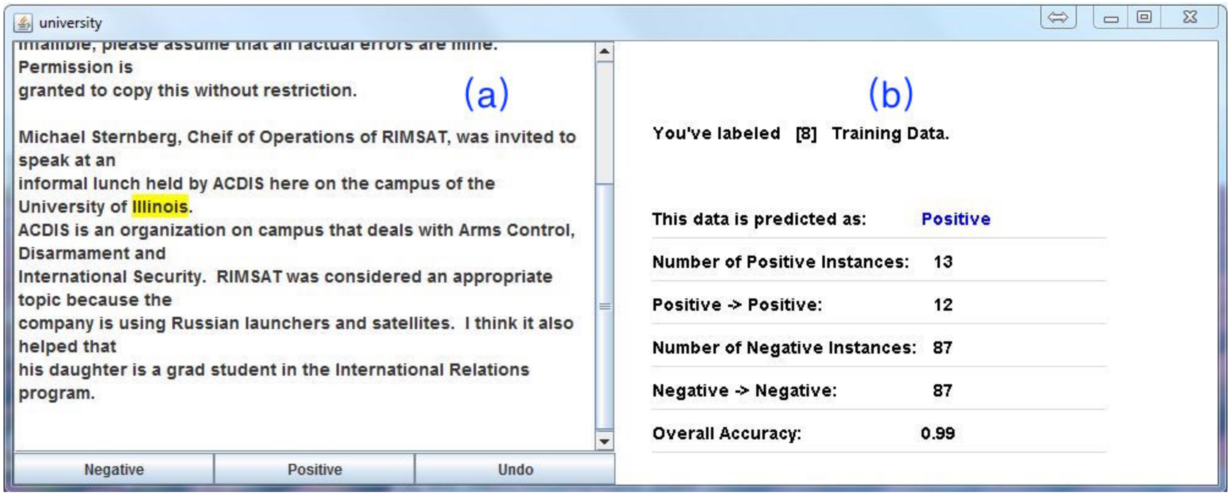
Whenever the user adds a label to the system, the classifier updates its model. To help users understand how well the current model performs, we provide users with two kinds of information to evaluate the system.

(1) Before labelling the training dataset, users will be asked to label 100 test data. When the system updates because of a new label, all the test data will be reclassified. Users can evaluate the classifier based on its correctness on the test data.

(2) Before labelling each training datum, users can see the current classifier's prediction on this data. If the classifier's prediction always matches the user's decision, then we can hypothesize the classifier has learned the concept well. Otherwise, if the classifier's prediction always contradicts the user's decision, then we can hypothesize that the classifier is still learning the concept, or the classifier is incapable of learning the concept.

4.3 Traditional Interface

The goal of this thesis is to explore how best to enhance data labelling such that users of machine learning can perceive the effect of their data labelling, assess classifier performance, and remain motivated. Given the above information, it may be the case that revealing the



infallible, please assume that all factual errors are mine.
Permission is
granted to copy this without restriction.

Michael Sternberg, Cheif of Operations of RIMSAT, was invited to speak at an
informal lunch held by ACDIS here on the campus of the
University of Illinois.
ACDIS is an organization on campus that deals with Arms Control, Disarmament and
International Security. RIMSAT was considered an appropriate topic because the
company is using Russian launchers and satellites. I think it also helped that
his daughter is a grad student in the International Relations program.

You've labeled [8] Training Data.

This data is predicted as:	Positive
Number of Positive Instances:	13
Positive -> Positive:	12
Number of Negative Instances:	87
Negative -> Negative:	87
Overall Accuracy:	0.99

Figure 4.1: Traditional Interface: (a) Labelling Window (b) Statistics Window

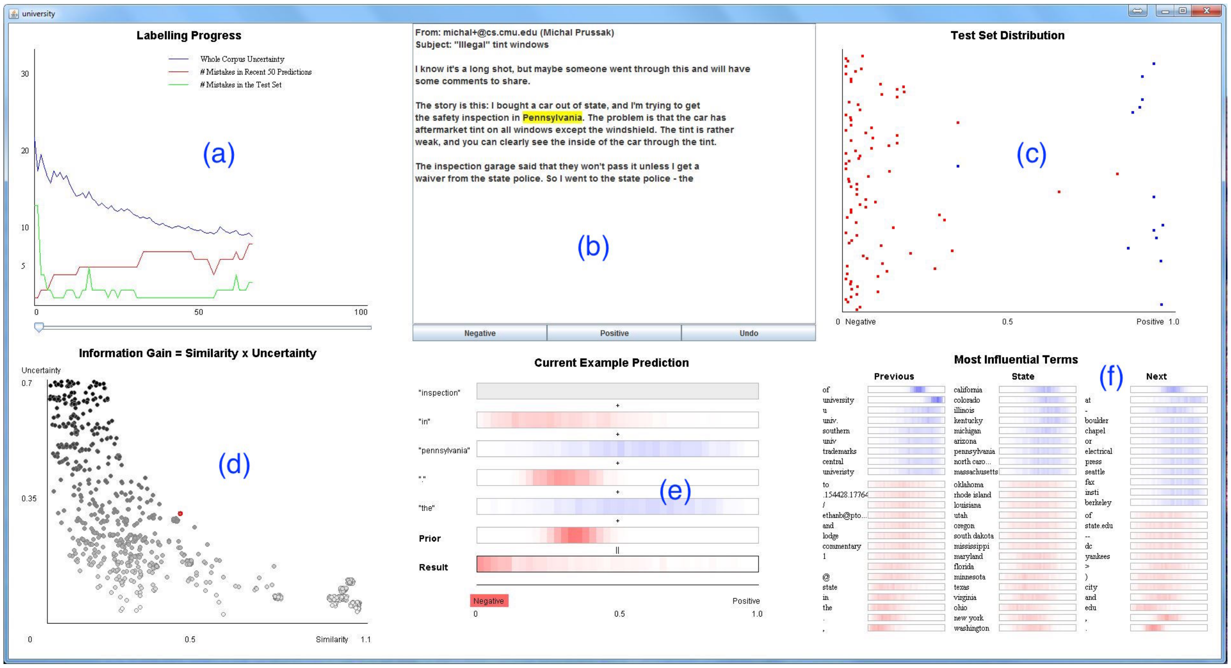
above information in text form to the user is sufficient. With this in mind, we created a traditional machine learning labelling interface (Figure 4.1) that reveals this information to the labellers. It consists of a labelling window(a) and a statistics window(b). The labelling window contains the current labelling task, and the statistics window contains information about the classifier's performance.

4.3.1 Labelling Window

The Labelling Window(Figure 4.1(a)) is the interface where users perform the labelling task. It highlights the string that matches a named entity and shows the surrounding context. Users can click the "Negative" or "Positive" button to label an instance. They can also click the "Undo" button if they realize they labelled something incorrectly. In most of the labelling tasks today, labellers only work on this window, which is sufficient if they are uninterested in more information about the classifier.

4.3.2 Statistics Window

The Statistics Window(Figure 4.1(b)) shows the information about the current classifier's performance. It shows the classifier's prediction on the current data, with "Positive" in blue color or "Negative" in red color. The font color helps users monitor the classifier's



From: michal+@cs.cmu.edu (Michal Prussak)
Subject: "Illegal" tint windows

I know it's a long shot, but maybe someone went through this and will have some comments to share.

The story is this: I bought a car out of state, and I'm trying to get the safety inspection in Pennsylvania. The problem is that the car has aftermarket tint on all windows except the windshield. The tint is rather weak, and you can clearly see the inside of the car through the tint.

The inspection garage said that they won't pass it unless I get a waiver from the state police. So I went to the state police - the

Previous	State	Next
of	california	
university	colorado	at
u	illinois	-
univ.	kentucky	boulder
southern	michigan	chapel
univ	arizona	or
trademarks	pennsylvania	electrical
central	north caro...	press
univeristy	massachusetts	seattle
to	oklahoma	fax
.154428.17764	rhode island	insti
/	louisiana	berkeley
ethanb@pto...	utah	of
and	oregon	state.edu
lodge	south dakota	--
commentary	mississippi	dc
1	maryland	yankees
	florida	>
@	minnesota	)
state	texas	city
in	virginia	and
the	ohio	edu
.	new york	,
,	washington	.

Figure 4.2: Visualization Interface: (a) Labelling Progress (b) Labelling Window (c) Test Set Distribution (d) Information Gain (e) Current Prediction (f) Influential Terms

prediction without moving their eyes from the labelling window. The following lines show the classifier's performance on the test set, including the number of positive instances in the test set and the number of positive instances being predicted correctly as positive, the number of negative instances in the test set and the number of negative instances being predicted correctly as negative. It also helps users keep track of the labels they have already provided. Without visualizations, developers are likely to output such data to the terminal if they want to see the updates of the classifier's performance. We put this information in our interface to simulate that environment.

4.4 Visualization Interface

As mentioned in Munzner's book[22], the visual systems provide a very high-bandwidth channel to our brains. One question we had, given our ability to monitor and analyze classifier's behavior during labelling, was whether textual information was sufficient to monitor classifier's performance for labellers, or whether enhanced visual presentations, i.e. visualizations, of classifier's performance might better communicate information to labellers. To test this, we designed a visualization version of Label-and-Learn to facilitate users' understanding of the data. As is shown in Figure 4.2, this interface consists of six windows: Labelling Progress(a), Labelling Window(b), Test Set Distribution(c), Information Gain(d), Current Prediction(e), and Influential Terms(f). The Labelling Window(b) is the same as the one in traditional interface. In this section, we will go through the five novice-centric visualizations and tag them according to our taxonomy from chapter 3.

4.4.1 Current Prediction

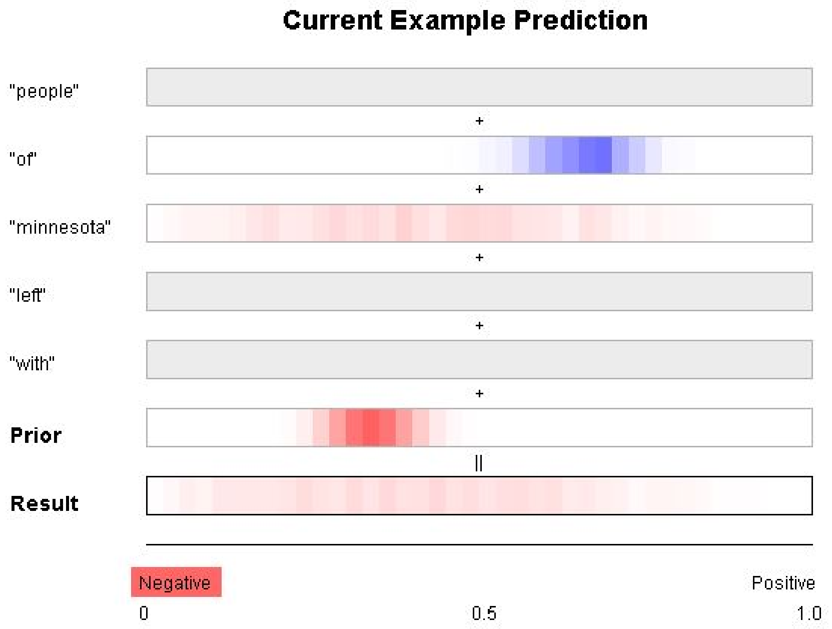


Figure 4.3: Current Prediction. This figure is a close-up of Figure 4.2(e)

For the current example that the user is labelling, this current prediction visualization(Figure 4.3) shows the *confidence* and *stability* of the current instance's features, prior probability, and final result, so that users understand why the system makes the current prediction. Blue and the right hand side means positive, while red and the left hand side means negative. The more condensed and saturated the region is, the less deeply the confidence will be affected by new labels, i.e. it is more stable. Conversely, the more area it covers, and the lighter the shading is, the more deeply that the confidence will be affected by new labels, i.e. it is less stable. If the color is grey everywhere, it means there is no information about the term at this moment, and the term has no influence on the classifier's decision. Its confidence will be entirely determined by the first label so it is at its most unstable state.

As the classifier uses the formula 4.1 to classify an instance, the confidence of a feature shown in the visualization is calculated as follows:

$$\text{confidence}(f) = \frac{p(f|+)}{p(f|+) + p(f|-)} \tag{4.4}$$

$$p(f|+) = \frac{count(f_+|set_f)}{count(pos|trainingset)}, p(f|-) = \frac{count(f_-|set_f)}{count(neg|trainingset)}$$

We are using $p(f|+)$ and $p(f|-)$ instead of $count(f_+|set_f)$ and $count(f_-|set_f)$ because $p(f|+)$ and $p(f|-)$ are the terms used in the naive Bayes calculations as in 4.2 and 4.3, and comparing them would give users better understanding of the model's decision.

The prior probability is calculated as follows:

$$p(+) = \frac{count(pos|trainingset)}{size(trainingset)}$$

The shading of the color is actually the posterior distribution of the above probabilities. We assume $p(f|+)$, $p(f|-)$, $p(+)$ all follow beta distribution. We take 1000 samples from each beta distribution($p(f|+)$, $p(f|-)$, and $p(+)$), calculate and combine the result using 4.4 for each sample, and use the distribution of the samples as the feature's posterior distribution to indicate stability. The more condensed the distribution is, the more information we have, and the less likely the probability will be affected by a new label.

For the last bar, we highlight the final decision by using red shading for a negative label and blue for a positive label, helping users quickly see the result.

This visualization is designed to help users understand why the classifier makes the current prediction, and understand which features are positively affecting the classifier in making a correct decision.

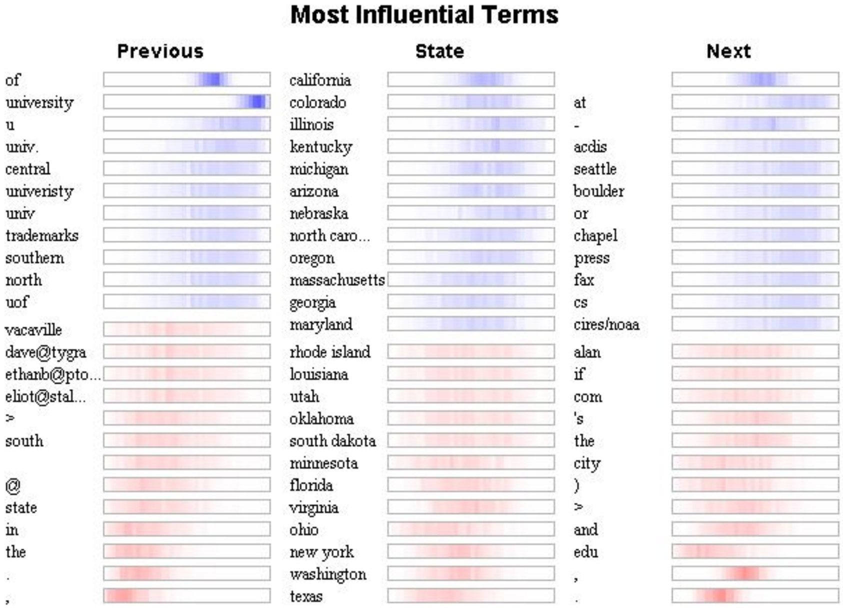


Figure 4.4: Influential Terms. This figure is a close-up of Figure 4.2(f)

4.4.2 Influential Terms

Shown in Figure 4.4, the Influential Terms visualization shows the 24 most influential terms in the previous dictionary, the current dictionary (State), and the next dictionary. The bar for each feature shows its confidence and stability, using same effects as in Current Prediction: blue and the right hand side means positive, while red and the left hand side means negative. The more condensed and saturated the region is, the more stable the feature is.

A feature's influence is calculated as:

$$\text{Influence}(f) = count(f|set_f) \cdot \max(\text{confidence}(f), 1 - \text{confidence}(f))$$

We select the 24 most influential terms according to this scheme to avoid highly positive/negative terms that appear only once, and those high-frequency terms that do not show strong signs of positive/negative. In each dictionary, the terms are first separated by overall polarity: positive indicators are above the negative indicators. In the positive part, the terms are sorted according to influence, with the highest influential term at the top. In the negative part, the terms are reversely sorted according to influence, with the highest influential term at the bottom. This helps users quickly locate the most influential terms.

This visualization helps users quickly locate the most influential features, and see whether these features match his knowledge about the task, what features are missing, and whether the features are stable.

4.4.3 Test Set Distribution

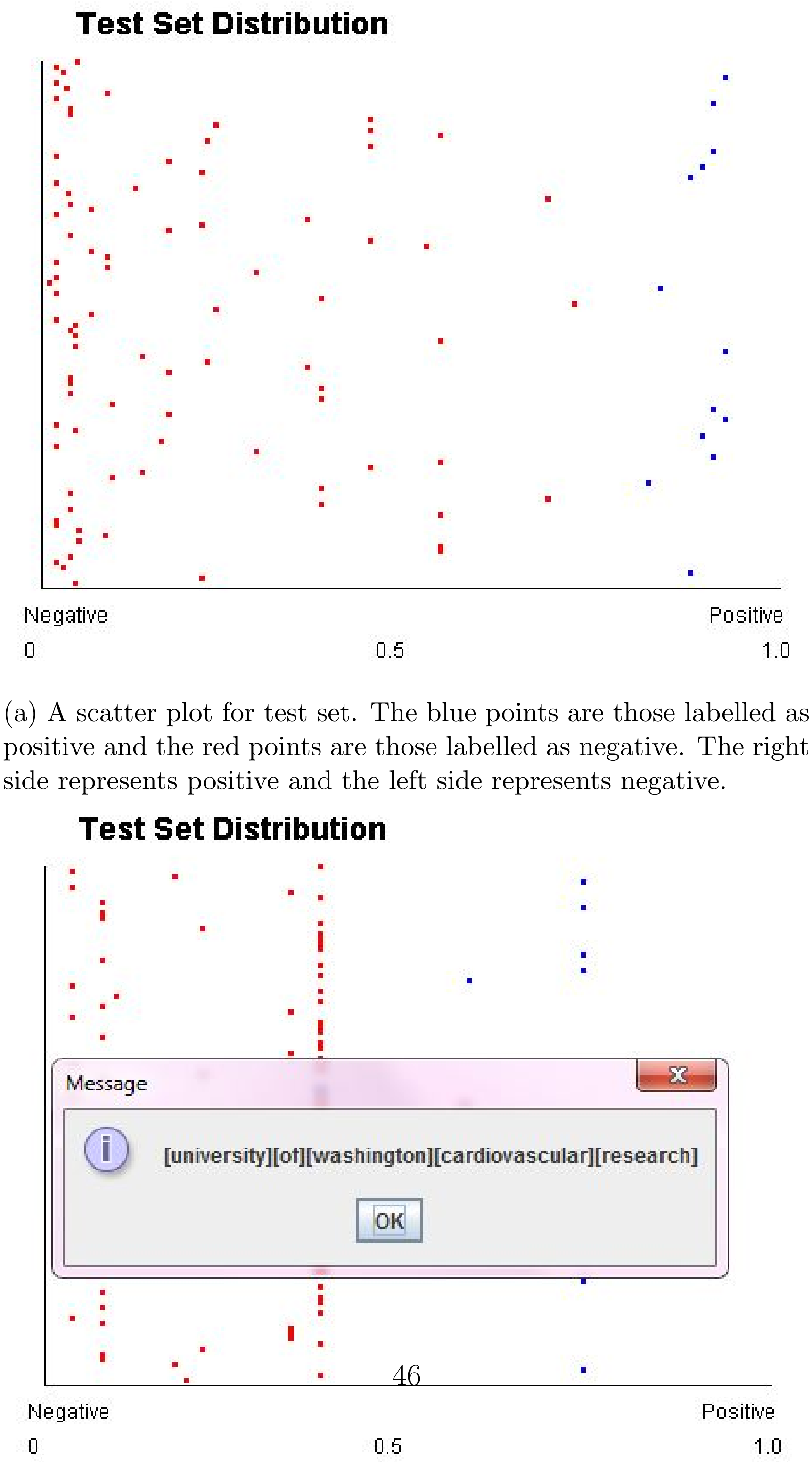
The Test Set Distribution, shown in Figure 4.5a, shows the predictions for the 100 test data the user first labelled. The blue points are those labelled as positive and the red points are those labelled as negative. The data points are spreaded out on y-axis simply by their order in the dataset. The x-axis represents the system's certainty in its prediction, with the right side representing positive and the left side representing negative. The blue points on the right side are true positives, those on the left side are false negatives. The red points on the left side are true negatives, and those on the right side are false positives. The points around the center are those instances the system is uncertain about. This visualization helps users examine the classifier's performance on the test set. When a user wants to know why an instance is wrongly predicted, or why the system is uncertain about an instance, the user can click on the interested point and examine the data the point represents, as shown in Figure 4.5b.

4.4.4 Labelling Progress

The Labelling Progress visualization provides a record of the labeller's progress with three trend lines(Figure 4.6).

As the user labels the data, the system keeps track of the following information:

- The **red line** indicates the number of mismatches in the last 50 predictions and user's labels. If the classifier is accurate in predicting, there should be no mismatch in its last 50 predictions and user's labels, and the line should fall to zero.
- The **green line** indicates the number of mistakes the current classifier makes in predicting the test set. If the classifier is accurate in predicting, it should make no mistake in the test set, and the line should fall to zero.



[university][of][washington][cardiovascular][research]

(a) A scatter plot for test set. The blue points are those labelled as positive and the red points are those labelled as negative. The right side represents positive and the left side represents negative.

(b) Users can always click on a point to examine the data.

Figure 4.5: Test Set Distribution. This figure is a close-up of Figure 4.2(c)

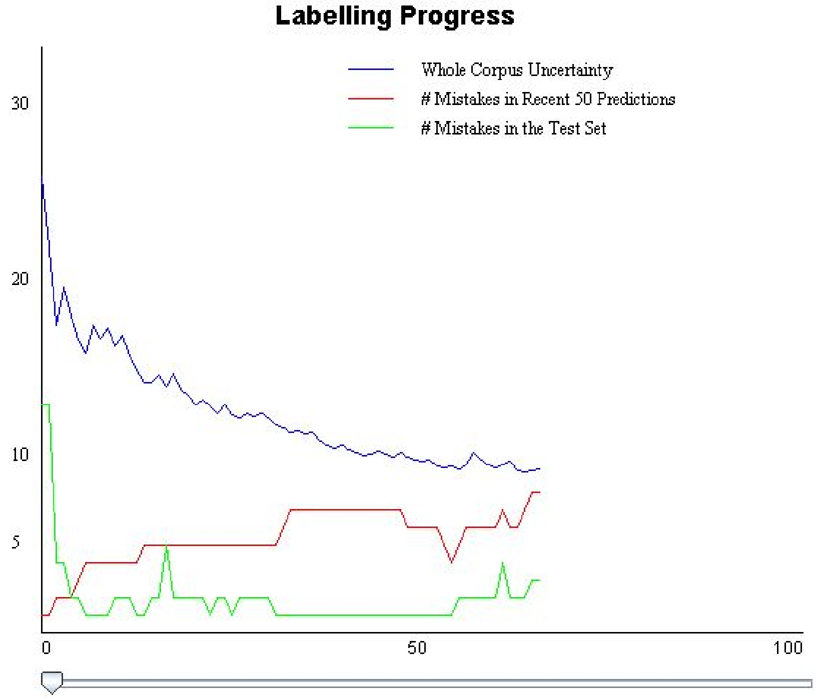


Figure 4.6: Labelling Progress. This figure is a close-up of Figure 4.2(a)

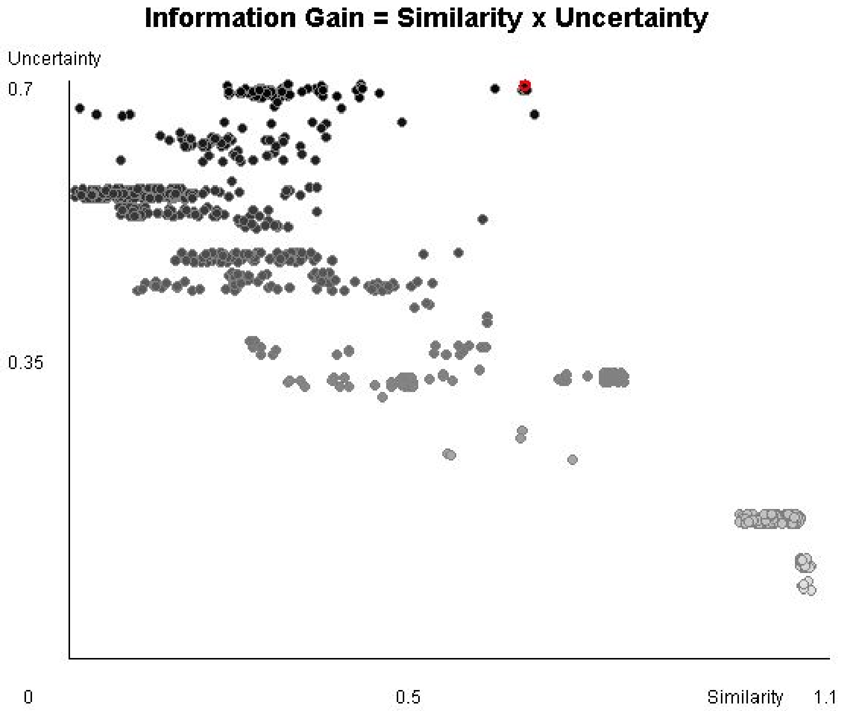


Figure 4.7: Information Gain. This figure is a close-up of Figure 4.2(d)

- The **blue line** indicates the overall uncertainty the system has across all data, which is calculated as $\sum_{x \in U} H(x)$, where $H(x)$ is data x's entropy. If the classifier has enough information to predict, there should be little uncertainty in the dataset, and the line should also fall to zero.

Users can drag the slider to see previous progress. This visualization helps users see how the classifier is improving with more labels added. It can help users answer the following questions: Is the classifier good enough? (Observe if the three lines are close to zero) Is the classifier still learning? (Observe if the red line and the blue line are going up but the green line is going down) Is it going to be better with more labels? (Observe if the three lines have the tendency to go down) This visualization can also help users see what they have achieved so far and motivate them to continue labelling. The use of three trend lines largely reduces the chance that users would prematurely give up.

4.4.5 Information Gain

As described in section 4.2.2, the system applies the Active Learning scheme to reduce the labeller's workload. It selects the instance with the highest information gain and requests its label. For each data, the information gain is calculated as:

$$IG(x) = H(x) \cdot Sim(x) = H(x) \cdot \frac{1}{U} \sum_{x' \in U} \mathrm{sim}(x, x')$$

Figure 4.7 visualizes all the data in the unlabelled pool with their information gain and how it is calculated. The y-axis represents the entropy of each data point, and the upper points are data with high uncertainty. The x-axis represents the similarity score of each data point, and the points towards the right are the most representative data. The shades of the points also reflects their entropy. The system always chooses the top-right data point for the user to label next, as it has the largest product of similarity and uncertainty. This point is highlighted as red. This visualization shows the user why this data is selected to label, how the unlabelled data is distributed according to the similarity, and the classifier's performance on the unlabelled data. It is easy to see that if all the representative data points have low entropy, and only a few low-similarity data points have high entropy, then we can say the classifier is quite certain about most of the data and already has enough information to make predictions.

4.5 Summary

This chapter described Label-and-Learn, the visualization system we implemented to help machine learning novices quickly build a classifier for named-entity-recognition tasks. The system uses a naive Bayes model to learn and classify data. The traditional interface contains a labelling window where users perform the labelling task, and a statistics window where users can see statistics about the classifier's current performance. The visualization interface contains a labelling window and five visualization windows: Current Prediction, Influential Terms, Test Set Distribution, Labelling Progress, and Information Gain. The viusalizations are designed to help users develop more insight into the data. We design this system to help users understand their classifier. Based on our taxonomy, we tag our visualizations as in Table 4.1. We did not include visualizations using dimensionality reduction as they need some expertise to interpret and interact. In the next chapter, we will evaluate our two Label-and-Learn interfaces in a laboratory-style experiment.

Visualization	Selected / All Information	Labelled / Unlabelled Data	Classified / Unclassified Data	Feature / Instance-Centric	Interaction	Original Data	Prediction Confidence	Classifier's Assessment	Highlighting
Current Prediction	Selected	Unlabelled	Classified	Feature	-	✓	✓	-	✓
Influential Terms	Selected	Labelled	Unclassified	Feature	-	✓	✓	-	-
Test Set Distribution	All	Labelled	Classified	Instance	✓	-	✓	-	-
Labelling Progress	All	Labelled	Classified	Instance	✓	-	-	✓	-
Information Gain	All	Unlabelled	Classified	Instance	-	-	✓	-	✓

Table 4.1: Visualizations under the taxonomy

Chapter 5

Evaluation

To investigate the visualizations' effectiveness with novice labellers, we evaluate our approach-as instantiated in Label-and-Learn-via the following research questions:

- In comparison to the traditional interface, do the visualizations influence the quality of user's labels?
- Can the visualizations help users better understand the current classifier's performance and better predict the classifier's success than the traditional interface?
- Do the visualizations give users better labelling experience?
- What kind of information about the classifier do the users want to know? Which visualization is the most helpful towards understanding the dataset?

5.1 Experiment Design

In the experiment, participants will use both the traditional version and the visualization version of the Label-and-Learn system to build two classifiers by labelling data. In addition to simply providing labels, we ask the participants to imagine themselves as people who want a successful classifier built, so that they will take a closer look at the data while labelling, have a rough understanding of how the data looks, and decide whether a workable classifier can be built.

5.1.1 Measures

We evaluate the utility of our system by collecting the following data during the experiment:

Quality of the Labels

For each system participants used, we collect the labels provided and compare the number of mistakes they made to analyze the quality. Note that whenever a participant makes a mistake, the system will alert him and ask him to relabel. This ensures that the classifiers built by all the participants are the same so that we can use the same classifier's result to assess participants' mental model.

Mental Model

After labelling 50, 100, 200, 400 instances, the participant completes a short quiz(Appendix A) with the following questions to assess his mental model:

- **Prediction of the future performance.** Predict how many mismatches there will be in the next 50 labels and the system's decision, as well as how many mistakes there will be in the test set after 50 more labels.
- **Prediction of the success.** Predict if the dataset and the algorithm is good enough such that the classifier will be successful with 1000 training data, and decide whether it is time to stop labelling because the performance is unlikely to improve.
- **Stability of the features.** Select the most unstable feature from the given choices. This question tests if participants can identify weaknesses of the classifier. The features in the choices include: extreme positive/negative features, uncertain features, features that appeared many times, features that appeared less frequently, and features that never appeared.
- **Classifier's decisions and reasons.** Predict the classifier's decisions on three instances and provide reasons by selecting the influencing factors: features or prior probability. The questions include instances that are always predicted as negative, instances that are always predicted as positive, instances whose predictions change after more labels, and instances that have the same number of positive factors and negative factors. If a participant finds equal number of positive indicators and negative indicators, she can choose the "Uncertain" option.

User Experience

After finishing the labelling task and the mental model quizzes, participants rate their workload for the two systems using the NASA-TLX questionnaire[10]. We would also interview participants about their preference for the two systems if given a pure labelling task and if given a labelling and learning task.

Helpful Information

Participants rate the utility for each visualization on a Likert-Scale (Appendix B), and provide feedback in a semi-structured interview (Appendix C). Questions explore features and data that participants found useful, visualizations that they did not examine, and other information they wished to see.

5.1.2 Participants

We recruited 20 participants from University of Waterloo graduate and undergraduate students. The participants were screened to ensure they understood simple charts and graphs (e.g., students with a high school math background, or people who can use Excel). As the study examines the usefulness of the visualizations, and our visualizations are non-traditional, those who cannot understand simple graphs and charts will introduce a confound to the experiment; they lack the essential knowledge to build a classifier. There are already visualization tools supporting machine learning experts, and our visualizations are designed to support machine learning novices, so we did not accept participants who had taken a machine learning course, or had written code about machine learning algorithms.

As participants were required to use both versions of the system to build classifiers for two tasks, and as they needed to label a certain amount of data, the experiment took approximately 2 hours in total. Each participant received 20 CAD as remuneration.

5.1.3 Machine Learning Tasks

Each participant used both versions of the system to build a classifier. During the classifier building process, we hypothesize that, participants will also build a mental model of the performance of each classifier. To avoid learning effects, the Named-Entity-Recognition tasks they perform on the two versions of the systems are different. The two tasks are a university task and an address task.

University Task

In this task, participants train the system to determine whether the highlighted state is part of a university's official name.

Positive examples include:

- **New York** University
- **Massachusetts** Institute of Technology (colleges and institutes count)
- U of **Michigan** (abbreviations count)

Negative examples include:

- universities of **California** (does not specify a particular university)
- The University of Portland (**Oregon**) (is not part of an official name)
- the **Florida** researchers (is not about a university)

Characteristics of the dataset:

- A workable classifier can be built.
- It has strong positive features such as "university" and "of".
- Instances with "university" after the state name start to appear around 190 labels, which means the classifier cannot identify [**State Name**] university until 190 labels have been obtained.
- The classifier starts to stabilize after 250 labels.

Address Task

In this task, participants will train the system to determine whether the highlighted state is part of a full address, and refers to the state. A full address should contain either the street number or zip code.

Positive examples include:

- 49 Locust Avenue, Suite 104; New Canaan, **Connecticut** 06840
- 2074 Abington Road, Cleveland, **Ohio** 44106

Negative examples include:

- 5841 S. **Maryland** Ave, MC 0953 (refers to the avenue, not the state)
- 7 West 66th Street, **New York**, NY 10023 (refers to the city, not the state)
- University of **Arizona**, Tucson, Arizona (is not part of a full address)

Characteristics of the dataset:

- A workable classifier cannot be built.
- The dataset does not have strong positive features. Most positive indicators are specific city names and postcodes. Only when the training data covers all the city names and zip codes can a successful classifier be built.
- The classifier rarely classifies an instance as positive, unless it has seen both the city name and the zip codes in the training data. Only one out of five positive instances in the test set is correctly classified as positive.
- The classifier stabilizes after 200 labels.

The documents were randomly selected from the 20 NewsGroups dataset[1]. Before the experiment, we processed the data and extracted the matched strings of state names. Each matched string is counted as one instance. Participants were only required to label the highlighted string as positive or negative, without identifying all the state names in the documents.

5.1.4 Procedure

The study used a 2×2 factorial design with two factors: system version and dataset. Each participant was randomly assigned a combination of system version and task: traditional-university and visualization-address, or traditional-address and visualization-university. The combinations assigned to the participants and the orders of the two systems were counterbalanced. The procedure for each participant is listed below:

1. Participants receive a short tutorial about machine learning and naive Bayes classifiers to understand how the system learns and predicts.

2. Participants use the first interface to build a classifier for the first task by labelling 400 instances.

   (a) Participants receive a short tutorial about the interface design, the meaning for each number and figure. After that, they take a look at the mental model quiz to understand what information they should look for while labelling.

   (b) Participants label 50 instances, and finish the first mental model quiz.

   (c) Participants label 100 instances, and finish the second mental model quiz.

   (d) Participants label 200 instances, and finish the third mental model quiz.

[1] http://qwone.com/ jason/20Newsgroups/

(e) Participants label 400 instances, and finish the last mental model quiz.

3. Participants use the second interface to build a classifier for the second task by labelling 400 instances (Same procedure as in 2).
4. Participants rate the workload for each system using the NASA-TLX.
5. Participants rate the utility of each visualization on a Likert-Scale, participate in an interview and provide feedback for the system.

We ask participants to complete the quiz four times in each trial because the classifier changes and will make different decisions each time on the same data. The participant's understanding of the data also changes as he knows more about the dataset. The interval between each quiz increases because both the classifier's model and the participant's mental model change early in the experiment, and become stable and less likely to change after seeing more data.

5.2 Result

As each measure is influenced by two factors: the system and the dataset, and the experiment is a mixed design, we use 2-way analysis of variance (ANOVA) to analyze the effects of the two factors and the interaction between them. And as our data is non-parametric, we applied aligned rank transform [42] on the raw data before doing analysis.

5.2.1 Quality of the Labels

The means and standard deviations of the labelling errors in each condition are shown in Figure 5.1. The mean of errors with the visualization version is larger than that with the traditional version, and we have a large standard deviation in each condition. This is due to participant's individual understanding of the concept and the speed of labelling. The result from mixed effect model shows no significant effect from the system version, dataset, or the interaction between the two factors. Based on this data, we can draw the conclusion that the system version has no significant effect on the quality of the labels.

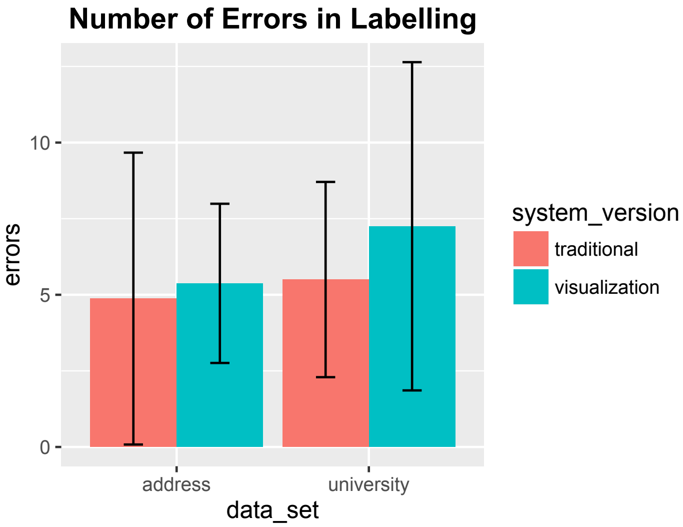


Figure 5.1: Labelling errors in each condition

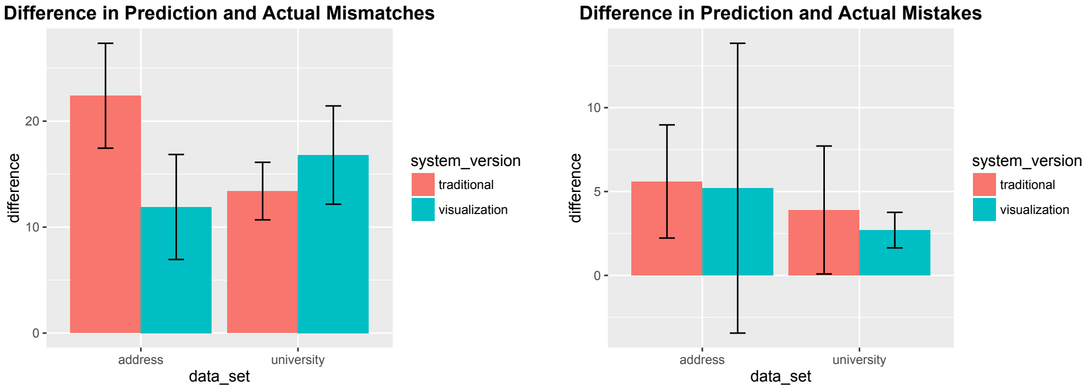
5.2.2 Mental Model

Prediction of the Future Performance

In each quiz, the participant would predict the number of mismatches in the next 50 labels, as well as the number of mistakes in evaluation of the test set after the next 50 labels. For each task a participant worked on, the score is calculated as

$$\sum_{i=50,100,200,400} |\text{predict}_i - \text{actual}_i|$$

where predict_i is the participant's prediction when he finishes labelling ith data. actual_i is the system's actual performance with $i + 50$ data. The means and standard deviations of the difference in mismatches in each condition are shown in Figure 5.2a, and the means and standard deviations of the difference in test set mistakes are shown in Figure 5.2b. The difference score should be low if a participant could correctly predict the system's future performance. Using aligned rank transform and mixed-effect model to analyze the mismatches, we found a significant effect of the system version ($F_{1,38} = 18.165, p < 0.0001$); the visualizations helped participants keep track of the number of mismatches in the last 50 predictions. Significant effect is also found in the interaction between dataset and system version($F_{1,36} = 37.8, p < 10^{-6}$). From the bar graph we can see that visualizations



(a) Mismatches in next 50 labels (b) Prediction of the Future Performance

Figure 5.2: Mistakes in Test Set after next 50 labels. Lower difference means better prediction.

particularly help with the *address* dataset. For the *address* data in the traditional version, participants could only see that the performance in the test set was not improving and mistakingly thought the mismatches in the predictions would stay the same. However, our system uses an active learning scheme to select instances, so the mismatches in predictions would actually decrease as the instances selected later are those the classifier is certain about.

The difference for the mistakes in the test set is smaller than the mismatches, due to the relative stability of the test set mistakes compared to the prediction mismatches. We find high standard deviation in the visualization system using *address* dataset, as one participant who is very doubtful about the dataset's success, provided very large numbers for this condition. Significance test does not show effects from the system version, dataset, or the interaction between them.

Prediction of the Success

In each quiz, participants decided whether the dataset and the algorithm are good enough such that the classifier will be successful with 1000 training data. They also decided whether it was time to stop labeling because the performance was unlikely to improve. Participants receive 1 score for correctly answering each question in each quiz (8 points in total). The mean and standard deviation are shown in Figure 5.3. Participants scored

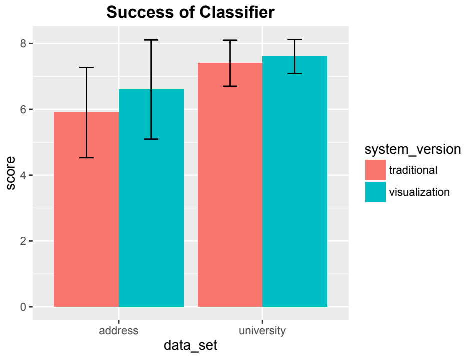
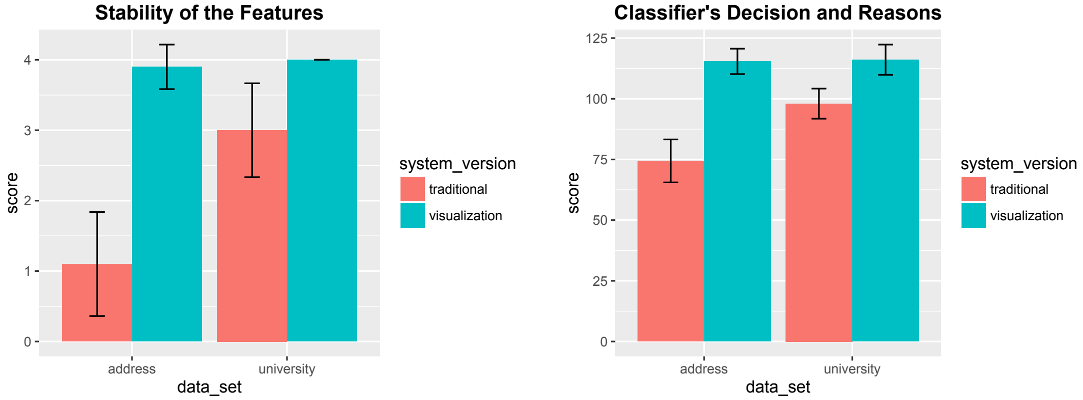


Figure 5.3: Success of the Classifier

higher when using the visualization system. They also scored higher when doing the *university* task. We found significant effect from the system version ($F_{1,38} = 4.20, p < 0.05$). We also found significant effect from the dataset ($F_{1,38} = 15.99, p < 10^{-3}$), which may be because participants were optimistic about the *address* dataset's success at the beginning, while a workable classifier cannot be built for the task. No significant effect was found from the interaction between the system version and the dataset ($F_{1,36} = 1.72, p = 0.20$).



(a) Stability of the Features (b) Classifier's Decision and Reasons

Figure 5.4: Knowledge about Current Classifier. Higher score means more correct answers.

Stability of the Features

In each quiz, participants selected the most unstable feature from the given choices. Higher score means more correct answers. The bar graph (Figure 5.4a) shows that participants score higher when using the visualizations. We found a highly significant effect from the system version ($F_{1,38} = 125.94, p < 10^{-12}$), as participants can simply look up each feature's presence and stability from *Most Influential Terms*. Highly significant effect was also found from the dataset ($F_{1,38} = 64.71, p < 10^{-9}$) and the interaction between dataset and system ($F_{1,36} = 35.38, p < 10^{-6}$). When participants were labeling for the *university* dataset, they are relating the text with the actual universities because they know of the existence of the universities, thus paying more attention to the data even without the visualizations.

Classifier's Decision and Reasons

In each quiz, participants predicted the classifier's decisions on three instances, and also provided the reasons by selecting the influencing factors from the five features and prior probability. Participants receive 4 points for each correct prediction and 1 point for judging the influence for each factor (6 points in total). There are 10 (points) $\times$ 3 (questions) $\times$ 4 (quizzes) = 120 points for each trial. The correct reasons weigh more than the correct decisions, since we want to make sure the participants know how the classifier works and why it makes a decision. When the participants have no clue of the classifier's rationale, they can still receive some points for guessing the correct decision.

Figure 5.4b shows the mean and standard deviation for each condition; the visualization system helped users predict the classifier's decision more accurately and reasonably. We found a highly significant influence from the system ($F_{1,38} = 115.41, p < 10^{-12}$). We also found a significant influence from the dataset ($F_{1,38} = 39.88, p < 10^{-6}$) and the interaction of the dataset and system ($F_{1,36} = 38.02, p < 10^{-6}$). As in the *university* dataset, features "university" and "of" are highly positive, and make it easy to make a judgement. However, in the *address* dataset, there is no such obvious feature, and positive terms are typically city names and zip codes. It is hard to keep track of all data without the help of visualizations.

5.2.3 User Experience

Figure 5.5 show the means and standard deviations of the NASA-TLX score. Low score means low workload.

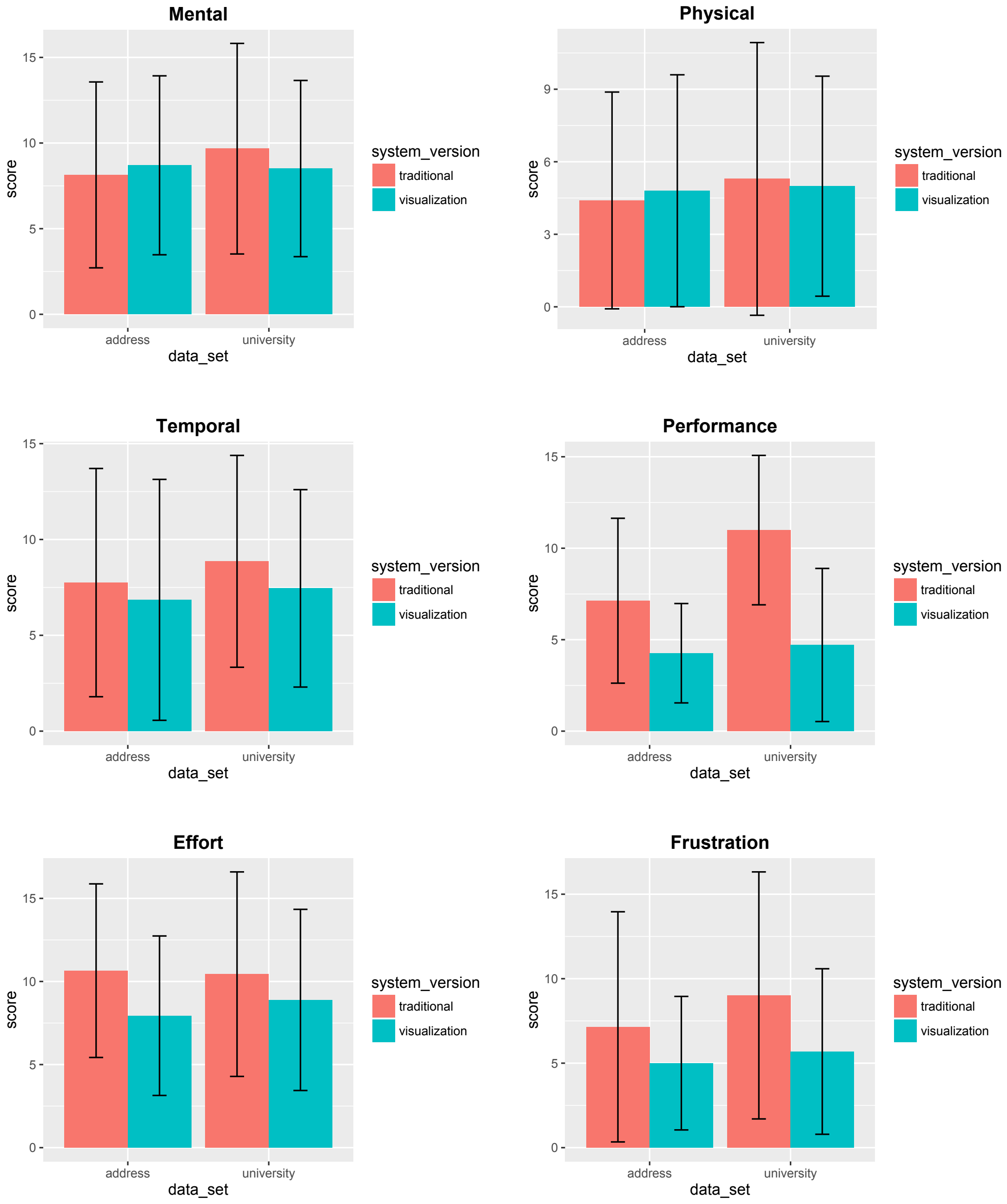


Figure 5.5: NASA-Task Load Index. Low score means low workload.

Mental Workload

As participants were not required to memorizing all the data, and they can answer the quizzes just from the visualizations or the vague impression from the data they have labelled, they focused most of their attention on labelling data, so there is no big difference in mental workload.

Physical Workload

All the experiments were conducted with participants sitting at a desk clicking mouse on "positive" or "negative" and finish some quizzes with a pen. The physical workload do not vary from situation to situation.

Temporal Workload

Temporal workload indicates how hurried participants felt when accomplishing the task. Although in the bar graph, visualization system showed lower temporal workload than traditional system, there is no significant effect from either the system version, dataset or the interaction between them. As all the participants finished the experiment in two hours, and 8 participants finished in 90 minutes, participants did not feel any need for hurry in completing the experiment.

Performance

In the NASA-TLX, better performance means lower workload, so the score is lower. From the bar graph, we can see that participants are more confident in their performance when using the visualization version of the system. Further significance test shows highly significant influence from system version ($F_{1,38} = 18.9, p < 10^{-4}$), because participants can always find the correct answers with the visualization system, but can only retrieve from the impression when using the traditional system. We also found significant influence from the interaction of system version and dataset ($F_{1,36} = 4.40, p < 0.04$). This is due to the *university* dataset has more positive instances, so the model is more complicate than the *address* dataset, and is harder to perform analysis without the visualizations.

Effort

Generally, participants rate lower effort in the visualization system. Compared to retrieving from the obscure memory when answering quizzes in the traditional system, simply looking up for information when answering quizzes in the visualization system, is less effortless.

Frustration

Figure 5.4b shows that participants rate lower frustration in the visualization system. Most participants reported they had no clue of the system's current performance or future performance when using the traditional system, causing more frustration. Meanwhile, some participants using the visualization system to build classifier for *address* report higher frustration than their traditional system and *university* trial, because the visualizations made them realize that the classifier is unlikely to be successful, which caused their frustration.

Summary

The result from the NASA-TLX score shows statistically significant differences in performance, effort and frustration subscores. The visualization version of the system gave users better experience in all three criteria, as the information from the visualization makes participants easier and more confident in answering the quizzes. The additional information had no negative influence on the other three criteria. Overall, participants had a better user experience with the visualization version of the system.

5.2.4 Helpful Information

After the participants finished the labelling tasks, they rated the utility of each visualization on a Likert-Scale, and provided feedback. The mean of the ratings for each visualization is shown in Figure 5.6. Labelling Progress received a mean score of 6.3, the highest of the five. Influential Terms ranks second with a score of 6.25. Test Set Distribution received a mean score of 5.15. Current Prediction received a mean score of 5.1. Information Gain was reported as the least useful visualization with a mean score of 2.8. Below is some feedback from the participants:

Labelling Progress:

P1: *"The most useful visualization. Very useful to make predictions."*

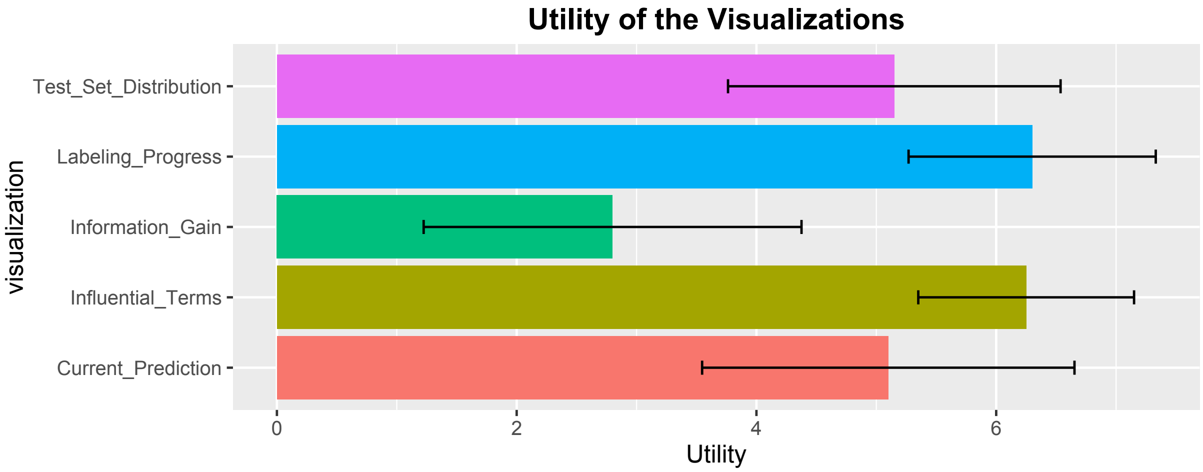


Figure 5.6: Utility of the Visualizations

P6: *"It shows exactly what I need to know. I can see the overall picture of the classifier's development. This visualization makes the biggest difference from the traditional version."*
P7: *"It helps you see how the changing is happening to the classifier."*
P11: *"It's easier to identify future trends with graphs."*
P12: *"It helps me see how close I'm at getting a good classifier.*"

Participants find this visualization useful as it tells them how much they have achieved in the past, and predict how the classifier is going to be in the future, giving them a direct feedback of the classifier's evolvement.

Influential Terms:
P1: *"It helps me understand how the computer behaves."*
P2: *"It tells me what the machine has learned so far, and what it is confused about."*
P4: *"This visualization helps me get more informed, so that I could make better guesses for the quiz."*
P12: *"I like the colors that distinguish the positive features and negative features."*

This visualization is extremely helpful for learning the stability of the features and predicting the classifier's decisions. It also helps users keep track of what the computer has learned so far.

Test Set Distribution:

P12: *"It tells me how correctly the system classifies the test set."*
P20: *"When there is only a few data in the uncertain area in the center, I can click on them to check why the classifier is confusing about them."*

And some negative feedback includes:
P3: *"The points are too small and are hard to see."*
P17: *"It's not useful for the quizzes"*

Most users take a look at this visualization, but the small size of the points and that it provides no direct information for the quizzes make users less interested in it.

Current Prediction:

P2: *"It tells me why the system predicts the current instance as positive or negative. The shade is important. It tells me how sure the decision is."*
P4: *"It clearly shows what features the classifier has no information about."*

And some negative feedback includes:
P8: *"I didn't look at it all the time."*
P13: *"I used the red line in the labelling progress more than this."*

Participants take a look at this visualization mainly because it helps them understand how naive Bayes works, and it is easy to see the large bars highlight the features of the current instance. However, it is only useful for the current instance, and does not help users understand the overall behavior of the classifier.

Information Gain:

P4: *"It's not related to the quizzes."*
P9: *"I didn't think too much on similarity. This visualization is scattered and hard to read."*
P11: *"It tells me how much uncertainty the data has, but it's redundant with the blue line in the Labelling Progress."*

It also received some positive feedback, such as
P2: *"It's very useful. I can see how the graph changes and where most data lies."*

Most participants did not look at it mainly because the quizzes do not ask information directly from it, and they did not really care why an instance is selected to label.

To summarize, participants find the visualizations that provide information for answering the quizzes most useful. They are also interested in the classifier's evolution, the numbers that directly show the classifier's performance, and the information that directly reflects the classifier's inner model. On the contrary, they are less interested in the visualization of dataset's distribution, as they cannot directly make a decision from that visualization.

5.2.5 Other Feedback

In the interview, we also asked the participants their preference for the two system versions. All participants preferred the visualization version, as it provides more information about the classifier and helps answer quiz questions. We also asked their preference between the two system versions if they were just asked to label without answering the quizzes. 16 participants prefer the visualization version, saying they want to see what is happening in the system, and more visual information rather than plain numbers and texts, 2 participants reported same level of preference, and 2 others prefer the traditional version as the visualizations might distract them from labelling. This result suggests that these visualizations help label both for those who want to build a classifier and for the labellers who are simply performing the labelling task.

Improvements

Participants also provide some suggestions to improve the visualizations:

- Test Set Distribution:
  - Show the percentage of true positive, true negative, false positive and false negative.
  - Use bar graph to indicate the number and percentage.
  - Add a vertical line in the center to separate positive and negative predictions.

  Participants want to see the exact numbers as a direct way to know the current performance. This is especially helpful when their task is to reach a quantitative goal. Participants also want to know exactly which class each instance was predicted as, especially when there are many points in the center area, so that they can judge the classifier's accuracy.

- Current Prediction:
  - Show a number indicating the stability.

The channel of color saturation is less effective than the channel of position on common scale [22]. When two shades are similar, it is hard to discern which one is darker. Although classifier builders do not necessarily need to compare each feature's stability, adding the numbers would provide them with a quantitative way to analyze the data.

- Influential Terms:
  - Sort the terms alphabetically, or add a search bar, so that it is easy to look up a term.
  - Show all the terms in the set.
  - Use the color green for positive instances.

  Participants mainly use this visualization to answer the quiz questions, so they want to quickly locate a term by browsing a sorted list or by entering search terms. This is useful, since machine learning builders may also want to look up an interesting term to see how much the classifier has learnt about it. It is also useful to show all the terms, so that the users have full access to the classifier's status. One of the participants suggested that green is a better positive indicator. However, we chose blue and red to avoid red-green color blindness.

- Information Gain:
  - Add click feature as in the Test Set Distribution.
  - Show positive/negative decision for those unlabelled data, not only just uncertainty.

  The current system does not have a click feature, due to the overlap of points. We could consider implementing this in the future, as this feature could allow users to see each instance in the training set according to their informativeness. Users may even be given the chance to choose the data they want to label themselves. Showing the positive/negative decision may provide more information, but whether it is a distraction still needs study.

- Labelling Progress:
  - Use the prediction's confidence in the trend line, indicating how wrong the mistakes are.
  - Show longer trends without having to scroll, or add scale up/down feature.

  We designed a trend line using the prediction's confidence in one of our prototypes, but do not use it in the system because, in pilot studies, participants struggled to interpret the trend line. We may add it as an option to allow users to see it if they need the information. Features like scale up/down will give users more freedom to control the trend line view, and is a useful add-on to improve user experience.

Some of the suggestions would help us design better visualizations in the future, while some of them appeared in our prototypes but were replaced by current design for better user experience. They also indicate that the participants understood, and were actually using the visualizations, so that they can notice their deficiencies and suggest improvements.

5.3 Summary

This chapter describes the design and result of our experiment to examine the usefulness of the Label-and-Learn system. We found that the visualizations have no negative effect on the quality of labels. As well, visualizations improve users' mental model in predicting classifier's success, understanding the features' stability, and the classifier's decision. Users also preferred the visualization version as they reported more confident in performance in the NASA-TLX. Participants also expressed overall preference in the visualization system in an interview. The results from our experiment show that the visualizations help users build a better mental model of the classifier and have a better labelling experience.

Chapter 6

Conclusions

6.1 Discussion

We described our experiment to validate our Label-and-Learn system in the previous chapter. The experiment produced an overall good result of the visualizations, both in helping users understand the classifier and in improving user experience. In this section, we will discuss the experiment result by answering the four research questions:

- In comparison to the traditional interface, do the visualizations influence the quality of user's labels?
- Can the visualizations help users better understand the current classifier's performance and better predict the classifier's success than the traditional interface?
- Do the visualizations give users better labelling experience?
- What kind of information about the classifier do the users want to know? Which visualization is the most helpful?

For the first question, we previously anticipated that the visualizations might distract the labeller's attention from labelling, might add bias to the user's perception, and might negatively affect the correctness of the labels. However, the result does not show an effect on the quality of the labels from the visualizations. We could conclude that adding visualizations to the labelling system does not negatively affect the labelling task, and could be widely applied in future designs.

The second question asks the visualizations' helpfulness in understanding the classifier's performance. Experimental results showed a significant effect on predicting the mismatch numbers in the next 50 predictions from the visualization. Although this effect is more significant in one dataset than in other, visualizations do help users predict classifier's performance.

We did not find a significant effect of the system version on other predictions such as whether it is time to stop labelling. Part of the reason is the experiment design. As described in section 5.1.3, we can stop labelling after 250 labels for the *university* dataset, and stop after 200 labels for the *address* dataset. However, after the quiz at 200 labels, participants only had the chance to express their decision at 400 labels, the end of the experiment, where most participants could reach the conclusion that it is time to stop labelling after the accuracy in the test set has stayed the same for a significant period.

Visualizations had a significant effect on analyzing the stability of the features and predicting the classifier's decisions with the reasoning. This is anticipated, as visualizations show the users a lot of information needed to understand the current classifier's performance.

The third question examines user experience. The results from the NASA-TLX and the interviews indicate that the visualizations offer users a better labelling experience, especially making them more confident in decision making and predicting, and more effortless in figuring out the classifier's performance. The visualizations do not exert extra mental or physical load to the participants. On the contrary, participants value the visualizations during labelling. This is encouraging, as the visualizations not only help users finish tasks, but also make the labelling process more enjoyable.

To answer the last question, we examined the participants' feedback on each visualization. Labelling Progress and Influential Terms are ranked as the most helpful visualizations. As well, users are more interested in seeing information that help them answer the quizzes, give direct feedback indicating the classifier's performance, and give detailed information about the model. On the other hand, they are less interested in the distribution of the data. However, we still believe that the distributions are useful. Their low score in utility may result from poor visual design that makes them less attractive, as well as the experiment design which created a passive atmosphere of "finding answers to quizzes", rather than "building a successful classifier with a robust model".

6.1.1 Bias of the Tasks

As there are two tasks involved in the experiment: a *university* task where a workable classifier can be built and an *address* task where a workable classifier cannot be built, they bring in some influence to the result. We can find significant effect from the dataset in some of the results. One reason we bring in two tasks is to avoid learning effect of the dataset when using the two systems. Another reason is that, we want to see whether our system can help users identify the classifier's performance of both workable classifiers and non-workable classifiers. Results show that participants generally build better mental model when working on workable classifiers. As we counter-balanced the tasks and the systems and still found a significant effect from the visualizations, we can say that the differences in the two tasks did not affect the result that visualizations help users better understand data.

6.1.2 Bias of the Quiz

As participants reported in the interview, during the experiment, they were using the visualizations to search answers for the quiz. There might be some biases brought by the quiz that make participants prefer the visualization version and received higher score with the visualizations. In fact, the quiz is based on the information needed by real developers, hoping to guide the participants explore more about the classifier rather than label passively. The "Influential Terms" is the most influenced by the quiz as it provides the information about Stability and Decision questions in the quiz, but we believe it is also one of the most needed by developers.

Table 6.1 is an extension from Table 4.1, adding each visualization's utility in helping the labelling process. By studying this table, we found some relations between the utility of the visualizations and their corresponding category in our taxonomy:

- To help users predict classifier's success, we should design visualizations that show All Information, Labelled Data, Classified Data, and Prediction Confidence or Classifier's Assessment, are Instance-Centric, and support Interaction.
- To help users identify the feature's stability, we should design visualizations that show Labelled Data, are Feature-Centric, and show Original Data.
- To help users understand the classifier's decisions and reasons, we should design visualizations that are Feature-Centric, show Original Data, and show Prediction Confidence.

Visualization	Selected / All Information	Labelled / Unlabelled Data	Classified / Unclassified Data	Feature / Instance-Centric
Current Prediction	Selected	Unlabelled	Classified	Feature
Influential Terms	Selected	Labelled	Unclassified	Feature
Test Set Distribution	All	Labelled	Classified	Instance
Labelling Progress	All	Labelled	Classified	Instance
Information Gain	All	Unlabelled	Classified	Instance

Visualization	Interaction	Original Data	Prediction Confidence	Classifier's Assessment	Highlighting
Current Prediction	-	✓	✓	-	✓
Influential Terms	-	✓	✓	-	-
Test Set Distribution	✓	-	✓	-	-
Labelling Progress	✓	-	-	✓	-
Information Gain	-	-	✓	-	✓

Visualization	Predict Success	Identify Stability	Decision and Reasons	User Experience
Current Prediction	-	-	✓	-
Influential Terms	-	✓	✓	-
Test Set Distribution	✓	-	-	✓
Labelling Progress	✓	-	-	✓
Information Gain	-	-	-	✓

Table 6.1: Visualizations and their utility

- To improve user experience, we should design visualizations that show All Information and Classified Data, are Instance-Centric, support Interaction, shows Prediction Confidence or Classifier's Assessment.

As discussed above, visualizations can help users build better mental models, give them a better experience, and have no negative effect on the labelling quality. It is promising and exciting to bring visualizations to the labelling process of machine learning. Further research in this area is needed, and should focus on better interaction design of the system and more machine learning tasks.

6.2 Future Work

In this section, we explore areas of future work. These include improvements to the system, improvements to the experimental design, and follow-on research questions. These areas of future work are all driven by our initial study data and highlight the potential of continuing work in this domain.

6.2.1 Possible Improvements in the System

Our first area of future work involves potential improvements to the overall system. Opportunities exist to improve speed of the system, the design of visualization and the overall user experience. We explore each of these potential improvements in turn.

Improve the System's Speed

In our experiment, we evaluated the quality of the user's labels by comparing the labelling errors in two system versions. We could also evaluate the efficiency of the user's labelling task. We observed that participants could label as fast as 1 instance per second in the traditional system. However, the visualization system itself may take up to 2 seconds to update, as we need to sample $O(10^3)$ data for each feature to get the posterior probability. So the user's labelling speed is not comparable given the current data. We could improve the sampling algorithm, cache unchanged result, use parallel computing, or find a better model to calculate the posterior distribution. This could help users label more smoothly, and allow us to quantitatively analyze the effects the visualizations bring to the labelling process.

Improve the Visualization Design

One of the reasons that participants did not find the distribution visualizations useful is that the points are small and scattered, so it is hard for the distribution visualizations to catch participants' attention. Because participants never explored them, they were rated as low utility. One idea to make them more attractive is to support animation. Rather than abruptly changing one scatterplot to another, we could make the change gradual by adding animation to each point, showing how its prediction is changing with new information. Another idea is to visualize the distribution with a force-directed graph[15], as shown in Figure 6.1, an aesthetically pleasing way to show how the classifier's inner structure and the points' relationship changes over time.

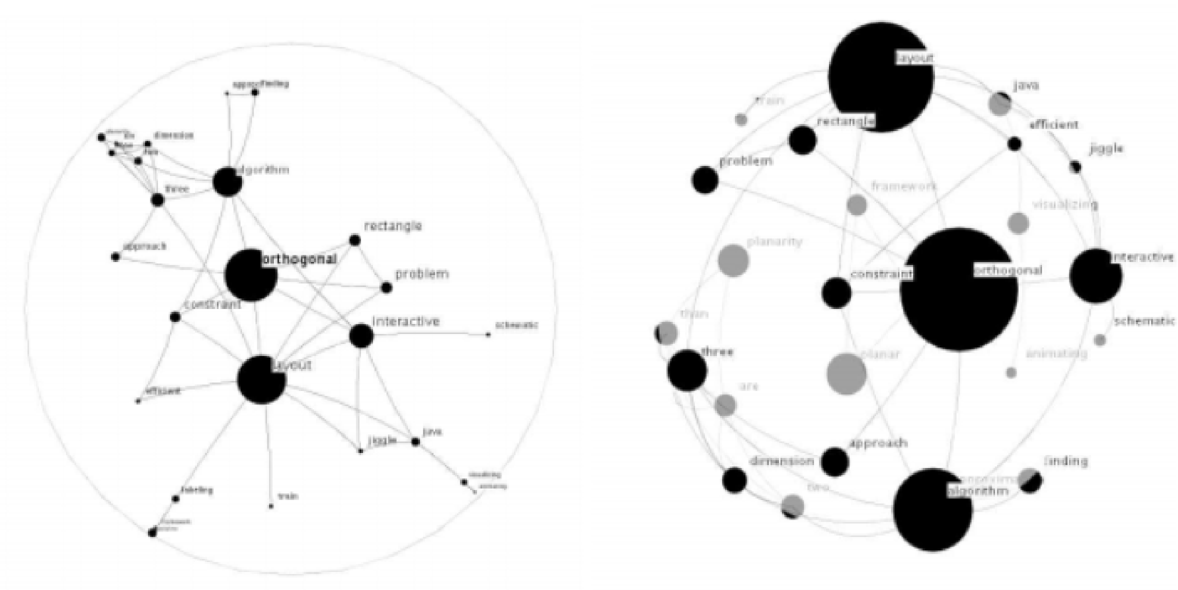


Figure 6.1: Examples of Force-Directed Graph

Improve the User Interaction Experience

As the system is just in its first version, aspects of the design could be enhanced. We have collected suggestions from our participants to improve the interface, including:

- We could add a search bar to Most Influential Terms visualization, so that users can look up whatever word they are interested in. We could also add a feature that could sort the terms in different ways: by alphabet, by importance, by stability, or by confidence. If we could improve the speed to calculate the posterior distribution, it would be good to have an overview of all the features the classifier has seen, so that the users know what proportion of the features are positive, negative, uncertain, and unstable. However, we would still like to keep the most important 24 terms at the top, so that users can easily see the most influential features that help the classifier make decisions.
- We could add a scale up/down feature to the Labelling Progress, which would help users easily see the overview of the trend as well as finer detail. We would still keep the current slider, allowing users to view the detail of each moment smoothly. We

could also add more interaction to this visualization, such as allowing users to click on a history timestamp to view what data they labelled that caused the change in the trend line. These features would help them locate the errors in their labelling if some abrupt changes show up in the trend line.

6.2.2 Possible Improvements in the Experiment Design

The next area of future work involves potential improvements in the experiment design. More work can be done in choosing the best time to stop labelling, encouraging the use of Information Gain, and including machine learning experts as participants. We explore each of these potential improvements in turn.

Time to Stop Labelling

The result does not show an effect of the system version in helping users decide when to stop labelling. Most participants did quite well on this question regardless of the system version, which is largely due to the increasing interval of the quizzes, as well as the arbitrary end point scheduled for labelling. For a future improvement, we could ask this question in a different way to get the time they would stop labelling more precisely, as well as keeping them thinking about this question during labelling. For example, the system could pop up a dialogue after every 10 labels asking them if this is a good time to stop labelling. Users can answer "stop labelling", or close the dialogue, or suspend the dialogue from popping up for a certain number of labels if they think the classifier currently needs many more labels. We could also extend the time and the number of labels of the experiment, in case some users want to label more data.

Encourage the Use of Information Gain

We found in the result that most participants did not like the Information Gain visualization simply because they did not use it, or more specifically, the quizzes did not ask questions about it. One way to keep participants engaged in this visualization is to include questions such as "What percentage of data in the unlabelled pool is the system uncertain about?" and "Which unlabelled data would be the most valuable to the classifier?" in the quizzes. This approach would make the participants explore this visualization. In fact, we observed in the experiment that many participants only looked at the visualizations when they needed answers for the quizzes, instead of being interested in the classifier's model.

Future research should consider creating an environment where the users are more actively engaged in building a successful classifier for themselves, and are motivated to learn the current model of the classifier instead of simply searching for answers for quizzes.

Include Machine Learning Experts

There was one student who scheduled an experiment with us, but when he arrived at the lab, we found that he had been doing research in machine learning. As he could be considered as an expert, we excluded this participant. However, he showed great interest in the study, since he found labelling a painful stage in machine learning research and needs such a system to improve the labelling experience. Another study to evaluate the system with machine learning experts could be valuable. Machine learning experts still constitute the majority of classifier builders, and although there are lots of visualizations designed for them, there is no visualization that supports leveraging labelling process. As the experts have high knowledge and experience with machine learning, they would know what information would help them decide on the quality of the classifier, and may glean significant utility from each visualization as well as offer more insightful and constructive suggestions on enhancements.

6.2.3 Possible Research Directions

Finally, we discuss possible research directions in this field as another area of future work. Researchers could consider conducting surveys of user requirements, including other models in the system, and generalizing the system to other tasks. We discuss each of these research opportunities in turn.

Survey of User Requirements

During the experiment, we noticed that some participants did not really look at the visualizations unless they needed to answer the quiz. This may be because we did not create an atmosphere to engage the participants in building a successful classifier, and may also be because users do not really need some of the information if not for the quiz. A systematic way to design and implement systems requires understanding user requirements. Requirements could be collected by interviewing labellers (including those who actually wants to build a classifier, and those who want to finish labelling) and novices who have tried using a machine learning library to build their desired classifier, asking them what kind of information they wish to know.

Include Other Models

The classifier implemented in the system used a naive Bayes model, and both Current Prediction and Influential Terms are visualizations specifically designed for this model. As most users reflected at the end of the *address* task, naive Bayes is not a good model for this task. To provide users with more perspectives of the data, we could consider introducing other models such as decision tree[31], SVM[13], or HMM[28], to our system. Lim and Dey have done some research and implemented a toolkit[19] for visualizing these models. By comparing the performance of different models, users can find out if the intrinsic problem is from the dataset, or from the mismatch of the data and model. Further explorations could consider allowing users to select features, or automatically selecting features for each specific task, as implemented in Amershi's ReGroup[2].

Generalize to Other Tasks

Our system is an experimental system that is designed to verify the utility of the visualizations in a lab environment for a simple task. Given that the visualizations have proved useful in lab testing, we should consider applying the visualizations to more complicated real-life tasks to help actual labellers and researchers. One way to do this is to build a visualization platform for labelling. This is not easy, as we need to overcome the difficulties of visualizing more advanced models, visualizing indirect features, and visualizing huge amount of data. As a first step, however, we could design systems for more specific tasks, including audio, image classification, and other machine learning models; when we have enough experience in designing for these tasks, we could generalize them in a platform to help whoever wants to build a classifier prototype by labelling data.

6.3 Final Remarks

This thesis contributes to the design and selection of visualizations for machine learning tasks. To synthesize research in this field and contribute to this goal we highlight the following research contributions:

- We first summarize the existing visualizations and create a taxonomy for them depending on the data they show, the advanced features they add on for specific task, and the goal of using the visualizations. This taxonomy would help machine learning researchers select the best visualizations for their tasks, and would also guide designers to select the best method and features to visualize the data.

- We also present our Label-and-Learn system, which is designed to help machine learning novices to understand the classifier's model while they perform the labelling task. The system includes a traditional version and a visualization version, and there are five real-time updating visualizations presented in the system. Four of the five visualizations are our innovative designs, and three of them were ranked top in the utility rating by the participants. The result of user study shows no negative impact on the labelling quality from the visualizations, and the visualizations actually improve user experience, help them understand the reasoning and performance of the current classifier, and predict the likelihood of its success in the future.

This thesis takes a step forward on machine learning visualizations. As machine learning is being widely used in many deployed systems, and the tasks often involve large volumes of data, visualizing this information can serve as an effective channel to help users quickly understand the knowledge contained in the dataset and understand the model being developed in the classifier. Although research has successfully explored visualizing a specific machine learning classifier, our work is the first to survey this space and create a taxonomy. We hope this taxonomy guides researchers to design better visualizations for machine learning tasks. Better visualizations can lower the barrier of machine learning, as well as help researchers work on their tasks more effectively. Our Label-and-Learn system is a demonstration of the potential of visualizations. We designed the system targeting the novice's understanding of the classifier, and used it at the labelling stage. The system proved to be helpful in improving user's understanding and experience. We were able to transform the tedious task of labelling to a self-motivating and informative process of learning. With the visualizations, machine learning novices are able to predict the classifier's future performance at an early stage to avoid wasting time in futile labelling.

References

[1] Saleema Amershi. *Designing for Effective End-User Interaction with Machine Learning*. PhD thesis, University of Washington, 2012.

[2] Saleema Amershi, James Fogarty, and Daniel Weld. Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 21–30, New York, NY, USA, 2012. ACM.

[3] Saleema Amershi, Bongshin Lee, Ashish Kapoor, Ratul Mahajan, and Blaine Christian. Cuet: Human-guided fast and accurate network alarm triage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 157–166, New York, NY, USA, 2011. ACM.

[4] Barry Becker, Ron Kohavi, and Dan Sommerfield. Visualizing the simple bayesian classifier, 1997.

[5] Jason Brownlee. Discover feature engineering, how to engineer features and how to get good at it. http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it. Accessed: 2016-03-01.

[6] Jerry Alan Fails and Dan R. Olsen, Jr. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pages 39–45, New York, NY, USA, 2003. ACM.

[7] James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. Cueflik: Interactive concept learning in image search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 29–38, New York, NY, USA, 2008. ACM.

[8] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Mach. Learn.*, 28(2-3):133–168, September 1997.

[9] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[10] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Human mental workload*, 1(3):139–183, 1988.

[11] Florian Heimerl, Steffen Koch, Harald Bosch, and Thomas Ertl. Visual classifier training for text document retrieval. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2839–2848, 2012.

[12] Florian Heimerl, Steffen Lohmann, Simon Lange, and Thomas Ertl. Word cloud explorer: Text analytics based on word clouds. In *Proceedings of the 2014 47th Hawaii International Conference on System Sciences*, HICSS '14, pages 1833–1842, Washington, DC, USA, 2014. IEEE Computer Society.

[13] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[14] Ian Jolliffe. *Principal component analysis.* Wiley Online Library, 2002.

[15] Stephen G. Kobourov. Spring embedders and force directed graph drawing algorithms, 2012. cite arxiv:1201.3011Comment: 23 pages, 8 figures.

[16] Todd Kulesza, Saleema Amershi, Rich Caruana, Danyel Fisher, and Denis Charles. Structured labeling for facilitating concept evolution in machine learning. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 3075–3084, New York, NY, USA, 2014. ACM.

[17] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, pages 126–137, New York, NY, USA, 2015. ACM.

[18] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *In Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1994.

[19] Brian Y. Lim and Anind K. Dey. Toolkit to support intelligibility in context-aware applications. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, pages 13–22, New York, NY, USA, 2010. ACM.

[20] M. A. Migut, M. Worring, and C. J. Veenman. Visualizing multi-dimensional decision boundaries in 2d. *Data Min. Knowl. Discov.*, 29(1):273–295, January 2015.

[21] Martin Možina, Janez Demšar, Michael Kattan, and Blaž Zupan. Nomograms for visualization of naive bayesian classifier. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD '04, pages 337–348, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[22] Tamara Munzner. *Visualization Analysis and Design*. A.K. Peters visualization series. A K Peters, 2014.

[23] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.

[24] Kayur Patel, Naomi Bancroft, Steven M. Drucker, James Fogarty, Andrew J. Ko, and James Landay. Gestalt: Integrated support for implementation and analysis in machine learning. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 37–46, New York, NY, USA, 2010. ACM.

[25] Kayur Patel, Steven M. Drucker, James Fogarty, Ashish Kapoor, and Desney S. Tan. Using multiple models to understand data. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1723–1728. AAAI Press, 2011.

[26] Kayur Dushyant Patel. *Lowering the Barrier to Applying Machine Learning*. PhD thesis, University of Washington, 2012.

[27] Kayur Dushyant Patel, James Fogarty, James A. Landay, and Beverly Harrison. Examining difficulties software developers encounter in the adoption of statistical machine learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pages 1563–1566. AAAI Press, 2008.

[28] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSp Magazine*, 1986.

[29] Lior Rokach and Oded Maimon. *Data Mining and Knowledge Discovery Handbook*, chapter Clustering Methods, pages 321–352. Springer US, Boston, MA, 2005.

[30] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 441–448, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[31] S. Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology, 1991.

[32] Christin Seifert and Michael Granitzer. User-based active learning. In *ICDMW 2010, The 10th IEEE International Conference on Data Mining Workshops, Sydney, Australia, 13 December 2010*, pages 418–425, 2010.

[33] Christin Seifert, Vedran Sabol, and Michael Granitzer. *Networked Digital Technologies: Second International Conference, NDT 2010, Prague, Czech Republic, July 7-9, 2010. Proceedings, Part I*, chapter Classifier Hypothesis Generation Using Visual Analysis Methods, pages 98–111. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[34] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[35] Burr Settles. *Active Learning.* Morgan and Claypool, 2012.

[36] Burr Settles, Mark Craven, and Lewis Friedland. Using multiple models to understand data. In *Proceedings of the NIPS workshop on cost-sensitive learning*, pages 1–10, 2008.

[37] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 287–294, New York, NY, USA, 1992. ACM.

[38] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, VL '96, pages 336–, Washington, DC, USA, 1996. IEEE Computer Society.

[39] Brian C. Smith, Burr Settles, William C. Hallows, Mark W. Craven, and John M. Denu. Sirt3 substrate specificity determined by peptide arrays and machine learning. *ACS Chemical Biology*, 6(2):146–157, 2011.

[40] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S. Tan. Ensemblematrix: Interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1283–1292, New York, NY, USA, 2009. ACM.

[41] Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Sally Jo Cunningham. Weka: Practical machine learning tools and techniques with java implementations, 1999.

[42] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 143–146, New York, NY, USA, 2011. ACM.

[43] Xiaojin Zhu, Andrew B. Goldberg, Ronald Brachman, and Thomas Dietterich. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.

APPENDICES

Appendix A

Quizzes for Mental Model

Participant # ________________

Periodic Assessment

University

1. How many mistakes(mismatches of its prediction and your label) will there be for the next 50 predictions? (If you have a range in mind, write down its median) _________

2. How many mistakes will the test set have after 50 more labels? (If you have a range in mind, just write down its median) _________

3. Which of the following features is the most unstable?
A. [University] [of] [California] [,] [] ([university] in previous set)
B. [U] [of] [Michigan] ['s] [students] ([of] in previous set)
C. [U] [of] [Michigan] ['s] [students] ([u] in previous set)

4. Do you think it's a good time to stop labelling now? (The classifier is performing well enough such that the test set is 100% predicted correctly. Or the accuracy of the test set won't be improved in the next 50 labels. Or all the instances have very low uncertainty) You can check both C and D.
A. Yes, we can stop labeling, the classifier is performing well enough.
B. Yes, we can stop labeling, the accuracy won't be improved
C. No, the classifier is not performing well enough
D. No, the accuracy can be improved

5. Do you think this classifier is going to be successful after, say, 1000 labels? (more than 80% of positive instances are predicted correctly, and more than 80% of negative instances are predicted correctly)
A. Yes B. No, the algorithm is not good enough

6. Indicate how you think the ***current*** classifier will label each of the following examples. If you see some conflicts among the features, check the **Uncertain** option.

(1) [ab4z][@][**Virginia**][.][EDU]
Negative ___ Positive ___ Uncertain ___
Check all the features that make you made the decision
A.[ab4z] (previous) B. [@] (previous) C. [virginia] (state) D.[.](next) E.[edu] (next) F. Prior Probability

(2) [John] [Stafford] [**Minnesota**] [State] [University]
Negative ___ Positive ___ Uncertain ___
Check all the features that make you made the decision
A.[john] (previous) B. [stafford] (previous) C. [minnesota] (state) D.[state] (next) E.[university] (next)
F. Prior Probability

(3) [University] [of] [**Alabama**] [in] [Huntsville]
Negative ___ Positive ___ Uncertain ___
Check all the features that make you made the decision
A. [university] (previous) B. [of] (previous) C. [alabama] (state) D.[in] (next) E.[huntsville] (next)
F. Prior Probability

Participant # _________________

Periodic Assessment

Address, 50

1. How many mistakes(mismatches of its prediction and your label) will there be for the next 50 predictions? (If you have a range in mind, write down its median) _________

2. How many mistakes will the test set have after 50 more labels? (If you have a range in mind, just write down its median) _________

3. Which of the following features is the most unstable?

A. [,] [in] [minnesota] [*] [94303] ([,] in previous set)
B. [,] [in] [minnesota] [*] [94303] ([in] in previous set)
C. [,] [in] [minnesota] [*] [94303] ([minnesota] in state set)
D. [,] [in] [minnesota] [*] [94303] ([*] in next set)

4. Do you think it's a good time to stop labelling now? (The classifier is performing well enough such that the test set is 100% predicted correctly. Or the accuracy of the test set won't be improved in the next 50 labels. Or all the instances have very low uncertainty) You can check both C and D.

A. Yes, we can stop labeling, the classifier is performing well enough.
B. Yes, we can stop labeling, the accuracy won't be improved
C. No, the classifier is not performing well enough
D. No, the accuracy can be improved

5. Do you think this classifier is going to be successful after, say, 1000 labels? (more than 80% of positive instances are predicted correctly, and more than 80% of negative instances are predicted correctly)

A. Yes B. No, the algorithm is not good enough

6. Indicate how you think the ***current*** classifier will label each of the following examples. If you see some conflicts among the features, check the **Uncertain** option.

(1) [eliot@lanmola.engr][.][**washington**][.][edu] (eliot)
Negative ___ Positive ___ Uncertain ___
Check all the features that make you made the decision
A.[eliot@lanmola.eng] (previous) B. [.] (previous) C. [washington] (state) D.[.] (next) E.[edu] (next)
F. Prior Probability

(2) [Arbor][,] [**Michigan**] [48109-0752] [(313)]
Negative ___ Positive ___ Uncertain ___
Check all the features that make you made the decision
A.[arbor] (previous) B. [,] (previous) C. [michigan] (state) D.[48109-0752] (next) E.[(313)] (next)
F. Prior Probability

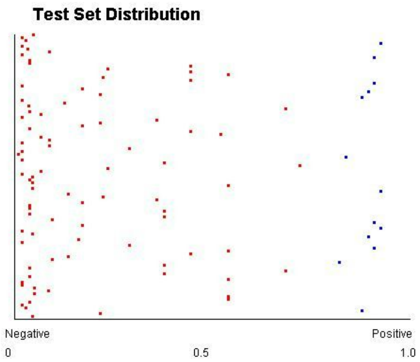
(3) [Cleveland] [,] [**Ohio**] [44135] [|]
Negative ___ Positive ___ Uncertain ___
Check all the features that make you made the decision
A. [cleveland] (previous) B. [,] (previous) C. [ohio] (state) D.[44135] (next) E.[|] (next)
F. Prior Probability

Appendix B

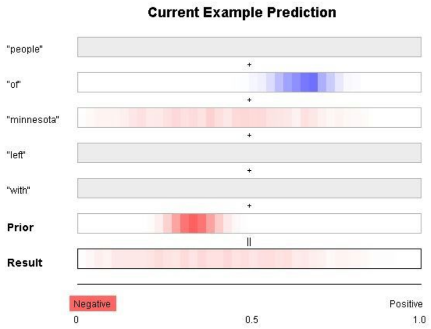
Questionnaire for Utility of the Visualizations

Participant #: ____________________

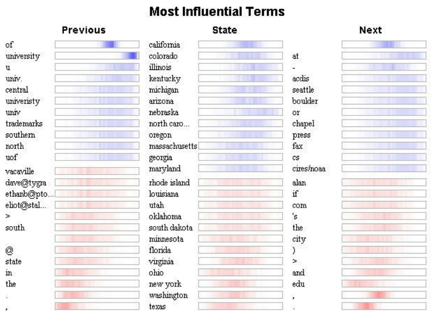
Utility of the Visualizations

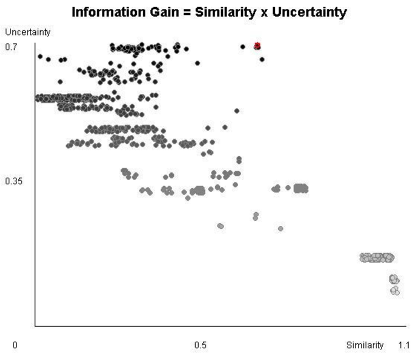


Very Unuseful Very Useful
1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7



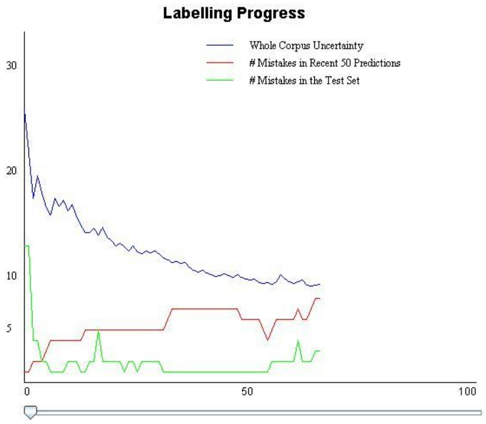
Very Unuseful Very Useful
1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7





Very Unuseful Very Useful
1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7

Very Unuseful Very Useful
1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7



Very Unuseful Very Useful
1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7

Appendix C

Interview Questions

Participant #: ___________________

Post-Experiment Interview

(filled by the researcher)

1. Which labeling interface do you prefer? What’s your choice if you do not need to answer the quizzes?

2. Which Labeling interface helps you more accurately predict the classifier’s behavior?

3. What are the benefits and limitations for each visualizations?

- Test Set Distribution

Benefits:

Limitations:

- Current Prediction

Benefits:

Limitations:

- Most Influential Terms

Benefits:

Limitations:

- Information Gain

Benefits:

Limitations:

- Labelling Progress

Benefits:

Limitations:

4. How could the **traditional** interface be improved?

5. How could the **visualization** interface be improved?