Introduction

Have you ever imagined using Twitter to help dealing with natural disasters? Today, it is possible. Artificial intelligence combined with social media seems to have no limitations, and I do research in Machine Learning.

[click to next slide]

So What is Machine Learning?

Machine learning is the subfield of Artificial intelligence that gives computers the ability to learn without being explicitly programmed, as it was defined by Arthur Samuel - the American pioneer in the field of computer gaming and artificial intelligence who was actually born in Emporia, Kansas and attended College of Emporia.

click to next slide

[Examples of Machine Learning Tasks]

You probably use a learning algorithm dozens of times a day without knowing it. Every time you use a web search engine like Google to search the internet, one of the reasons that works so well is because a learning algorithm, one implemented by Google, has learned how to rank web pages.

Every time you use Facebook and it recognizes your friends' photos, that's also machine learning.

Every time you read your email and your spam filter saves you from having to wade through tons of spam email, that's also a learning algorithm.

Handwriting recognition. It turns out one of the reasons it's so inexpensive today to route a piece of mail across the countries, in the US and internationally, is that when you write an envelope like this, it turns out there's a learning algorithm that has learned how to read your handwriting so that it can automatically route this envelope on its way, and so it costs us a few cents to send mail thousands of miles.

Learning algorithms are also widely used for self-customizing programs. Every time you go to Amazon or Netflix, and it recommends the movies or products to you, that's a learning algorithm.

If you think about it, they have million users; there is no way to write a million different programs for your million users. The only way to have software give these customized recommendations is to have a machine learning model that customizes itself to your preferences.

[click to next slide]

So one key motivation for machine learning is that we cannot program everything, and some tasks are difficult to define algorithmically. For example, it is almost impossible to come up with a non-machine learning algorithm to tell if there is an arbitrary scene contains a bird.

Before I dive any deeper into my research, I’d like to introduce some terminology.

Terminology

We refer to the data as instances.

instances: The data is usually presented as a table. Each row corresponds to an instance, or example, or observation. For instance, here, each row is a record of some parameters of the weather on a particular day.

[click to next slide]

labels: the desired value that we try to predict. The outcome – whether they play tennis given the weather – is what we ultimately want to predict. We call it “label” or class.

[click to next slide]

features: While label is a dependent attribute, outlook, temperature and others are independent attributes. We can also call them features.

[click to next slide]

training data: To train a machine learning model, you need some data. The data that you use to train a model is called training data. The class labels for the instances in the training data are known.

[click to next slide]

testing data: the data that is not used in the training stage, and those class labels we need to predict, is called testing data. In practice, to evaluate performance of a model, we split original training data into two parts: for training and testing (since we need ground-truth labels to evaluate performance once the model is trained).

[click to next slide]

What is being trained? Basically, we try to estimate the weights for the parameters of a model that would accurately predict the class label for unseen instances.

Why Twitter for Disaster Management?

In my research, I use Twitter data, specifically, user-generated tweets about disasters such as floods, hurricanes, terrorist attacks and others, to build specific programs called classifiers that could identify tweets about disasters, and help disaster management teams gather useful information in real time.

So why Twitter?

Social media have been providing a very efficient, quick access to news and updates in real-time. And with Twitter, you do not need to wait for journalists to arrive at the scene, you can get most recent information posted by others in real-time.

Let me give you two real life examples.

During and immediately following Hurricane Sandy, users sent more than 20 million Sandy-related Twitter posts, or tweets, despite the loss of cell phone service during the peak of the storm.

Following the Boston Marathon bombings, when the Boston Police Department posted its final "CAPTURED!!!" tweet of the manhunt, more than 140,000 people retweeted it.

Here you can see example of tweets about some of the natural disasters. 🡪 point to the slide

Motivation for Machine Learning

Speaking about tweets, they are very ambiguous because it is human language. For instance, the hashtag #Matthew might refer to the Hurricane Matthew as well as to the Apostle Matthew.

So let’s think: if you wanted to build a program that would identify tweets about a current ongoing disaster, how would you do that? In general, if you are a Twitter user, you would probably try to use hashtags to find relevant content.

[click to next slide] 🡪 face of Matthew Perry

So using hashtags to identify relevant context might be extremely noisy.

That’s why we turn to machine learning to help us

[click to next slide]

Supervised Learning Model Workflow

Supervised Machine Learning is focused on building predictive models given much labeled data*.*

And the term supervised learning refers to the fact that we gave the algorithm a data set in which the "right answers" were given. That is, we gave it a data set of houses in which for every example in this data set, we told it what is the right price so what is the actual price that, that house sold for and the toss of the algorithm was to just produce more of these right answers such as for this new house, you know, that your friend may be trying to sell.

First, we need to collect the data, and preprocess it, remove noise and unnecessary information. We also need to come up with features that best describe our data. This process is called feature extraction.

Then, we feed the preprocessed data into the model for training. Once the training stage is over, the model is ready to make predictions on unseen testing data.

We perform the same steps on unseen data as we did on training data.

Only now we are not feeding any labels to the model, but instead we ask it to predict the class labels for each of the instances that we feed.

Data Description

The 60,000 tweets (10,000 in each disasters) posted during 6 crisis events in 2012 and 2013 have been labeled by crowdsourcing workers according to relatedness (as on-topic or off-topic).

Here is an example of what original data looks like (just a small sample).

click to next slide (“original tweets”)

Data Preprocessing

We remove non-printable characters.

Links, email addresses, and usernames are replaced with a URL/email/username placeholder for each type of entity, respectively, under the assumption that those features could be predictive.

Numbers, punctuation signs and hashtags are kept under the assumption that numbers could be indicative of an address, while punctuation/emoticons and hashtags could be indicative of emotions.

Duplicate tweets and empty tweets (that have no characters left after the cleaning) are removed from the data sets.

click to next slide

Data Description

As you can see, the number of instances is reduced after preprocessing.

click to next slide

Supervised Machine Learning Assumption

A supervised model (or classifier) trained on data (training data) from a particular domain (i.e. disaster) is expected to give accurate predictions on unseen data (testing data) from the same domain, assuming that the training and test data have similar characteristics. However, labeled data is not easily available for a current target disaster.

However, labeled data from a prior source disaster is presumably available, and can be used to learn a supervised classifier for the target disaster.

So the idea is to use tweets about previous disasters to train the classifier to identify tweets about a new current on-going disaster for which we don’t have much data right away.

In other words, the classifier trained to identify tweets about “hurricanes” may not perform well identifying tweets about “wildfire”.

The reason for this is because Machine learning is very different from human learning. Humans are able to learn from very few examples and apply the learned knowledge in novel conditions. In contrast, machine learning methods only perform well when given much data.

Unfortunately, the source disaster data and the target disaster data may not share the same characteristics, and the classifier learned from the source may not perform well on the target. Domain adaptation techniques, which use unlabeled target data in addition to labeled source data, can be used to address this problem.

We study single-source and multi-source domain adaptation techniques, using Nave Bayes

classifier.

----------------------------------------

Unfortunately, the source disaster data and the target disaster data may not share the same characteristics, and the classifier learned from the source may not perform well on the target. Domain adaptation techniques, which use unlabeled target data in addition to labeled source data, can be used to address this problem.

We study single-source and multi-source domain adaptation techniques, using Nave Bayes

classifier.

Experimental results on Twitter datasets corresponding to six disasters show that domain

adaptation techniques improve the overall performance as compared to basic supervised

learning classifiers.

Domain adaptation is crucial for many machine learning applications, as it enables the

use of unlabeled data in domains where labeled data is not available.

-----------------

The idea is to use tweets about previous disasters to train the classifier to identify tweets about a new current on-going disaster for which we don’t have much data right away.

So I have to apply domain adaptation techniques to adapt the classifier trained on one disaster to perform well on another.

Experimental results show that various domain adaptation techniques improve the overall performance of the classifier.

Thus, domain adaptation becomes essential as it enables the classifier perform well on domains where data is limited or unavailable.

And imagine, next time you are posting about a storm coming up, you might be helping save someone’s life!