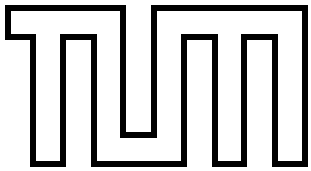


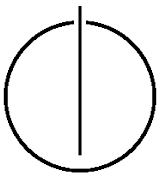
FAKULTÄT FÜR INFORMATIK

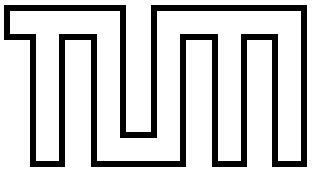
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Utilizing Crowd Intelligence for Online Detection of Emotional Distress

Siddhant Goel





FAKULTÄT FÜR INFORMATIK

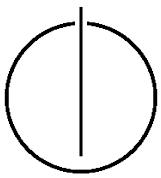
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Utilizing Crowd Intelligence for Online Detection of Emotional Distress

Insert thesis title in German here

Author:	Siddhant Goel
Supervisor:	Prof. Dr. Claudia Eckert
Advisor:	Han Xiao, M.Sc.
Date:	March 15, 2013



Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

I assure the single handed composition of this master's thesis only supported by declared resources.

München, den March 15, 2013 Siddhant Goel

Acknowledgments

If someone contributed to the thesis... might be good to thank them here.

Abstract

An abstracts abstracts the thesis!

Contents

Outline of the Thesis

Part I: Introduction

CHAPTER 1: INTRODUCTION This chapter presents an overview of the thesis and it purpose. Furthermore, it will discuss the sense of life in a very general approach.

CHAPTER 2: RELATED WORK Related Work

CHAPTER 3: PROBLEM DEFINITION Problem Definition

Part II: Theoretical Background

CHAPTER 1: CLASSIFICATION METHODS

CHAPTER 2: TEXT REPRESENTATION

CHAPTER 3: ENSEMBLE LEARNING

Part III: Experiments

CHAPTER 1: EXPERIMENTS

CHAPTER 2: APPLICATION

CHAPTER 3: RESULTS

Part IV: Conclusion

CHAPTER 1: CONCLUSION

Part I.

Introduction

1. Introduction

Here starts the thesis with an introduction. Please use nice latex and bibtex entries [**?**]. Do not spend time on formating your thesis, but on its content.

1.1. Latex Introduction

There is no need for a latex introduction since there is plenty of literature out there.

2. Related Work

Related Work goes here

3. Problem Definition

Define the problem here

Part II.

Methodology

4. Classification Methods

In this chapter, we introduce the theoretical background necessary for understanding the following sections of this thesis. We start with introducing classification algorithms, in particular support vector machines which we use in building our system. We then explain the ways in which text input is passed on to machine learning frameworks. Finally, we introduce the concept of ensemble learning, which remains our main point of focus in this work, and some of the most popular methods it encompasses. We pay particular attention to bagging, boosting, and stacking.

4.1. Introduction

Machine Learning is a branch of computer science that deals with building and analyzing systems that are able to learn from data. Algorithms based upon such analysis involve constructing a model from a given dataset, and then using this model to perform other tasks. Machine Learning techniques can broadly be divided into two categories - supervised learning, and unsupervised learning.

- Supervised Learning
  Methods falling under this category operate in two phases -
  - In the first step, we assume the existence of a training data, which is used to build the model so that it takes into account the structure of the given dataset
  - In the second step, we use this model to make predictions on the testing data (the real world data) that the model is not aware of yet

  Our primary focus in this thesis remains on supervised learning methods.

- Unsupervised Learning
  Methods falling under this category operate in a single phase - the model starts with zero knowledge about the structure of the given dataset. As we feed data into the model, it continuously learns the structure of the given dataset and bases its predictions based on this knowledge. The main difference between this family of algorithms and supervised learning algorithms is the presence/absence of training data.

Classification is one of the fundamental problems in machine learning. Given a dataset D, we are required to separate the samples contained within the dataset into two (or more than two, depending on the input) classes. Formally, given a dataset in which each instance is of the form $[(d_1, d_2, ...d_n), l_j]$, where each d_i is the feature value of feature $k \in [1, n]$, and l_j is the label of the sample which can take a limited number of possible values, the aim is to calculate the value of l_j, given the feature information. This separation can usually be done using a supervised learning method, in which case we're given the training data (on which the model is built) and are required to predict the labels of the testing data, or

using unsupervised methods, where the model is required to identify the categories of the samples without any external help.

The performance of a particular classifier depends on a number of factors, one of which is the characteristic of the data to be classified. Not all classifiers are good for all classes of problems. Some classifiers suit a particular problem more than others; choosing a classifier for a problem still remains a decision which may or may not be completely scientific, even though there have been a number of tests being done to correlate classifier performance with data characteristics.

4.2. Support Vector Machines

Support Vector Machines (SVM) form a fairly popular class of machine learning algorithms used mainly for binary classification and regression analysis. When used for classification problems, they assume the input (training) data to be in a vector-space format, and build a large margin classifier using this data to assign future samples into one category or another. Given the training data, the goal of an SVM is to find a decision boundary (a hyperplane, or a set of hyperplanes in a high or infinite dimensional space) that can separates the two classes of data. A hyperplane that has the largest distance to the nearest training data point of any class usually gives a good performance.

Often, an SVM is not able to find a decision boundary which can separate data points beloning to the two classes, simply because the input data is not linearly separable. In such cases, SVMs use what is called as the *kernel trick*, transforming the input data to a much higher dimensional input space using a kernel function, which makes separation in that space easier. Since such transformation functions may be expensive in cost, the kernel functions satisfy some specific properties which makes their use feasible, one of which is that even though calculating $K(x)$ may be expensive, calculating $K(x, y)$, which is the dot product of vectors x and y in the higher dimensional input space is much cheaper in cost.

5. Text Representation

Text Representation

6. Ensemble Learning

Describe Ensemble Learning here

Part III.

Experimental Results

7. Experiments

Experiment settings, dataset, system built, approach, and everything practical goes here

8. Application

Documentation about the system goes here

9. Results

Results

Part IV.

Conclusion

10. Conclusion

Conclude

Appendix

A. Appendix

Appendix