Facebook Likes Predictor

# Project Report

# CSE 201

# Introduction

Our application uses the post history of the user to predict the number of likes the user will get on a new Facebook post. We use simple machine learning techniques to predict the number of likes. We use the first 80% of the user’s post to train the algorithm and the remaining 20% of the posts to test it. We then present our results along with the accuracy of the algorithm for the user. In addition, we provide the user with some additional statistics, specifically, the top 15 expected likers, a random assortment of previous posts and the accuracy of the algorithm within 3, 5, 10 and 15 like margins.

# Algorithm

We use simple *linear regression* to predict the number of likes.

*hΘ(x) = Θ0 + Θ1x1 + Θ2x2 + .... + Θnxn = XΘT*

Here hΘ(x) is our hypothesis, which is a function of X and Θ. X, here, is our data-set, which is an m X n matrix, where m is the number of data samples, and n is the number of features. Θ is an n-dimensional vector, which stores parameters.

The crux of the algorithm lies in fitting these parameters to our data set. Since the number of features in our case is small, we use the *normal equation*.

<math>

Here, *y* is an m-dimensional vector which holds the number of actual likes for each training example.

# Implementation

We use the Facebook Graph API to fetch the posts of a particular user. To do this, we need the authentication token of the user, which we use to construct a GET request URL. Finally, we use this URL to fetch the posts in the form of a JSON Array. Using these JSON Objects, we create an array of posts. Each post is a training example with n features, which we set while parsing the JSON Objects. We use the following 14 features:

* People tagged (post)
* People tagged (msg)
* Source
* Type of Status
* Presence of Message
* Presence of Description
* Presence of Caption
* Created hour of day
* Created day of week
* Length
* Presence of Location
* Type of Post
* Number of Emotes
* Number of Hashtags

Once we have the array of Post objects, we can construct the relevant matrices. To do the matrix construction and operations, we use the Apache Commons Linear Algebra library.

# Performance

Any kind of statistical method is fundamentally about correlation and not about causality. In that sense, it is difficult to draw “deep” conclusions about the nature of online social dynamics. What this means is that algorithms can be arbitrarily optimized to perform on the given data, but it is always possible to encounter new data on which their performance is poor. The “intelligent” task in the whole exercise is the selection of features – and that is inevitably where room for further improvement lies.

Screenshots



