2D Visualizations for Support Vector Machine and Neural Network

Jizhou Xu Aileme Omogbai
Johns Hopkins University
{$jxu55, aomogba1$}@jhu.edu

Abstract

We present a web-based demo that helps to visualize the operations and results of some machine learning binary classification algorithms. For this project, we have chosen to implement two non-linear methods, namely Support Vector Machines (SVM) and Artificial Neural Networks. This visualization tool is implemented fully in Javascript and so the data collection, training and display of the results is done on the client-side. It displays a 2D canvas such that data points can be added with mouse clicks and the decision boundary generated by the algorithm is displayed. There is also provision for adjusting some parameters of the model. The visualization can be found at http://metoerite-j.github.io/Visual-Machine-Learning.

1 Introduction

In the field of machine learning, binary classification problems stem from the need to separate a set of data points into two different classes. In the supervised case, we have data that is already separated such that each data point belongs to one of the defined classes. For the unsupervised case, we do not know the classes for the example data available. In both cases, we wish to find a model that can learn parameters for the classification such that new unlabeled data points can be assigned to the right class. The model therefore creates "boundaries" such that a data point is labeled according to the side of the boundary that it falls in.

If the boundary used to separate the two sets of data points is a line, then the problem is a linear classification problem. The data is not usually linearly separable and so complex functions may be needed to define the boundary. Also, while it is intuitive to visualize the concept of the boundary in 2 dimensions, the data is usually of much higher dimensions. This means that very complex decision boundaries are needed to bring about separation in practical applications. Even in 2D however, it may be difficult to picture the boundary that is generated by non-linear classifiers, and that is what this project aims to help with.

We present a web visualization tool that shows the classification results of two popular non-linear classifiers. The user is presented with a 2D canvas that contains default points that have been generated with a normal distribution and classified randomly. The classes used for the demo are represented by the colors black/white and this used to label the data points and the regions on the canvas. So, a point is labeled either black or white. Also, the color of any position in the canvas represents the class that a test data point (i.e. a point we want to classify) will be assigned to by the current model. Therefore, the boundary between the white and black regions on the canvas represent the decision boundary that was generated by the algorithm used. At the current stage, we have implemented two popular non-linear classifiers: Support Vector Machines (SVM) and Artificial Neural Network (ANN). For SVM, we focus on the Sequential Minimal Optimization (SMO) algorithm and the Primal Estimated Sub-Gradient Solver for SVMs. For ANN, we uses the classic feed-forward back-propagation neural network as our model.

SVMs are effective and popular classification learning tool. The task of learning a support vector machine is cast as a constrained quadratic programming problem. stochastic sub-gradians descent algorithm for solving the optimization problem cast by SVMs.

2 SMO

2.1 Background

SMO is one of the most popular algorithm for training SVMs. It tackles the dual representation of the SVM optimization problem, and employs an active set of constraints thus working on a subset of dual variables. Normally, training a Support Vector Machine (SVM) requires the solution of a very large quadratic programming (QP) optimization problem. SMO breaks this large QP problem into a series of smallest possible QP problems. These small QP problems are solved analytically, which avoids using a time-consuming numerical QP optimization as an inner loop. The amount of memory required for SMO is linear in the training set size, which allows SMO to handle very large training sets. Because large matrix computation is avoided, SMO scales somewhere between linear and quadratic in the training set size for various test problems, while a standard pro jected conjugate gradient (PCG) chunking algorithm scales somewhere between linear and cubic in the training set size. SMO's computation time is dominated by SVM evaluation, hence SMO is fastest for linear SVMs and sparse data sets.

2.2 The SMO Algorithm

For the production of the electronic manuscript you must use Adobe's Portable Document Format (PDF). This format can be generated from postscript files: on Unix systems, you can use `ps2pdf` for this purpose; under Microsoft Windows, you can use Adobe's Distiller, or if you have cygwin installed, you can use `dvipdf` or `ps2pdf`. Note that some word processing programs generate PDF which may not include all the necessary fonts (esp. tree diagrams, symbols). When you print or create the PDF file, there is usually an option in your printer setup to include none, all or just non-standard fonts. Please make sure that you select the option of including ALL the fonts. *Before sending it, test your* PDF *by printing it from a computer different from the one where it was created*. Moreover, some word processor may generate very large postscript/PDF files, where each page is rendered as an image. Such images may reproduce poorly. In this case, try alternative ways to obtain the postscript and/or PDF. One way on some systems is to install a driver for a postscript printer, send your document to the printer specifying "Output to a file", then convert the file to PDF.

For reasons of uniformity, Adobe's **Times Roman** font should be used. In LaTeX2e this is accomplished by putting

```
\usepackage{times}
\usepackage{latexsym}
```

in the preamble.

Additionally, it is of utmost importance to specify the **US-Letter format** (8.5in $\times$ 11in) when formatting the paper. When working with `dvips`, for instance, one should specify `-t letter`.

Print-outs of the PDF file on US-Letter paper should be identical to the hardcopy version. If you cannot meet the above requirements about the production of your electronic submission, please contact the publication chairs above as soon as possible.

2.3 Layout

Format manuscripts two columns to a page, in the manner these instructions are formatted. The exact dimensions for a page on US-letter paper are:

- Left and right margins: 1in
- Top margin:1in
- Bottom margin: 1in
- Column width: 3.15in
- Column height: 9in
- Gap between columns: 0.2in

Papers should not be submitted on any other paper size. Exceptionally, authors for whom it is *impossible* to format on US-Letter paper, may format for *A4* paper. In this case, they should keep the *top* and *left* margins as given above, use the same column width, height and gap, and modify the bottom and right margins as necessary. Note that the text will no longer be centered.

2.4 How We Implemented it

Center the title, author's name(s) and affiliation(s) across both columns. Do not use footnotes for affiliations. Do not include the paper ID number assigned during the submission process. Use the two-column format only when you begin the abstract.

Title: Place the title centered at the top of the first page, in a 15 point bold font. (For a complete guide to font sizes and styles, see Table 1.) Long title should be typed on two lines without a blank line intervening. Approximately, put the title at 1in from the top of the page, followed by a blank line, then the author's names(s), and the affiliation on the following line. Do not use only initials for given names (middle initials are allowed). Do not format surnames in all capitals (e.g., "Leacock," not "LEACOCK"). The affiliation should contain the author's complete address, and if possible an electronic mail address. Leave about 0.75in between the affiliation and the body of the first page.

Abstract: Type the abstract at the beginning of the first column. The width of the abstract text should be smaller than the width of the columns for the text in the body of the paper by about 0.25in on each side. Center the word **Abstract** in a 12 point bold font above the body of the abstract. The abstract should be a concise summary of the general thesis and conclusions of the paper. It should be no longer than 200 words. The abstract text should be in 10 point font.

Text: Begin typing the main body of the text immediately after the abstract, observing the two-column format as shown in the present document. Do not include page numbers.

Indent when starting a new paragraph. For reasons of uniformity, use Adobe's **Times Roman** fonts, with 11 points for text and subsection headings, 12 points for section headings and 15 points for the title. If Times Roman is unavailable, use **Computer Modern Roman** (LaTeX2e's default; see section 2.2 above). Note that the latter is about 10% less dense than Adobe's Times Roman font.

2.5 Sections

Headings: Type and label section and subsection headings in the style shown on the present document. Use numbered sections (Arabic numerals) in order to facilitate cross references. Number subsections with the section number and the subsection number separated by a dot, in Arabic numerals.

Citations: Citations within the text appear in parentheses as (Gusfield, 1997) or, if the author's name appears in the text itself, as Gusfield (1997). Append lowercase letters to the year in cases of ambiguities. Treat double authors as in (Aho and Ullman, 1972), but write as in (Chandra et al., 1981) when more than two authors are involved. Collapse multiple citations as in (Gusfield, 1997; Aho and Ullman, 1972).

References: Gather the full set of references together under the heading **References**; place the section before any Appendices, unless they contain references. Arrange the references alphabetically by first author, rather than by order of occurrence in the text. Provide as complete a citation as possible, using a consistent format, such as the one for *Computational Linguistics* or the one in the *Publication Manual of the American Psychological Association* (American Psychological Association, 1983). Use of full names for authors rather than initials is preferred. A list of abbreviations for common computer science journals can be found in the ACM *Computing Reviews* (Association for Computing Machinery, 1983).

The LaTeX and BibTeX style files provided roughly fit the American Psychological Association format, allowing regular citations, short citations and multiple citations as described above.

Appendices: Appendices, if any, directly follow the text and the references (but see above). Letter them in sequence and provide an informative title: **Appendix A. Title of Appendix**.

Acknowledgment sections should go as a last (unnumbered) section immediately before the references.

2.6 Footnotes

Footnotes: Put footnotes at the bottom of the page. They may be numbered or referred to by asterisks or other symbols.[1] Footnotes should be separated from the text by a line.[2] Footnotes should be in 9 point font.

[1] This is how a footnote should appear.

[2] Note the line separating the footnotes from the text.

Type of Text	Font Size	Style
paper title	15 pt	bold
author names	12 pt	bold
author affiliation	12 pt	
the word "Abstract"	12 pt	bold
section titles	12 pt	bold
document text	11 pt	
abstract text	10 pt	
captions	10 pt	
bibliography	10 pt	
footnotes	9 pt	

Table 1: Font guide.

2.7 Graphics

Illustrations: Place figures, tables, and photographs in the paper near where they are first discussed, rather than at the end, if possible. Wide illustrations may run across both columns and should be placed at the top of a page. Color illustrations are discouraged, unless you have verified that they will be understandable when printed in black ink.

Captions: Provide a caption for every illustration; number each one sequentially in the form: "Figure 1. Caption of the Figure." "Table 1. Caption of the Table." Type the captions of the figures and tables below the body, using 10 point text.

3 Length of Submission

The NAACL HLT 2010 main conference accepts submissions of long papers and short papers. The maximum length of a long paper manuscript is eight (8) pages of content and one (1) additional page of references *only* (appendices count against the eight pages, not the additional one page). The maximum length of a short paper manuscript is four (4) pages including references. For both long and short papers, all illustrations, references, and appendices must be accommodated within these page limits, observing the formatting instructions given in the present document. Papers that do not conform to the specified length and formatting requirements are subject to be rejected without review.

Acknowledgments

Do not number the acknowledgment section.

References

[Aho and Ullman1972] Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

[American Psychological Association1983] American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

[Association for Computing Machinery1983] Association for Computing Machinery. 1983. *Computing Reviews*, 24(11):503–512.

[Chandra et al.1981] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

[Gusfield1997] Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.