Support Vector Machines

Mathematically Sophisticated Classification

Todd Wilson

Statistical Learning Group
Department of Statistics
North Carolina State University

September 27, 2016

Presentation Outline

- SLG's history with SVM
- Overall history of SVM
- Sketch of SVM Derivation (weighted towards intuitive points)
- Implementation and Examples in R

Previously at SLG...

- November 2014: The "Golden Age" of SVM
  - Jami Jackson - Support Vector Machines
  - Brian Naughton - Support Vector Machines for Ranking Models
  - Huimin Peng - Support Vector Machines and Flexible Discriminant Analysis
- August 29 - September 6, 2016: The "Renaissance" of SVM
  - Dr. David Dickey - Introduction to Machine Learning
  - Cliffhanger between two presentations about SVM
- September 20, 2016: Andrew Giffin inquires about SVM

Inspiration for Today's Talk

- Nov. 2014 talks based on Chapter 12 of *The Elements of Statistical Learning* by Hastie, Tibshirani, and Friedman
  - Advanced text
  - SLG previously catered to a more advanced audience
- What makes this method a classifier?
  (Never asked for nearest neighbors, decision trees, etc.)
- Some intuition is seen in the derivation - let's explore this!
- Hopefully, today makes all previous talks accessible

History of SVM

- Vladimir Vapnik laid most of the groundwork for SVM while working on his PhD thesis in the Soviet Union in the 1960s
- Vapnik emigrated to U.S. in 1990 to work with AT&T
- Cortes and Vapnik (1995) finally introduced SVM to the world
- SVM has been very popular topic in machine learning since mid 1990s

Quotes

"[SVM] needs to be in the tool bag of every civilized person."
-Dr. Patrick Winston, MIT

"Wow! This topic is totally devoid of any statistical content."
-Dr. David Dickey, NCSU

Visualization of Problem

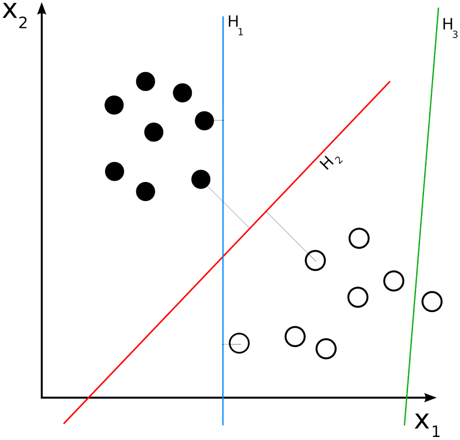

Image source: Open source via Wikibooks

Visualization of Problem

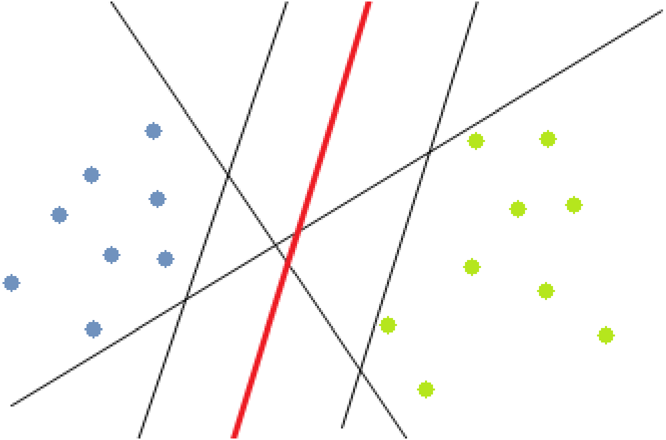

Image source: Open source via Wikibooks

Problem Setup

- Suppose we have binary data ($+$ and $-$) in the plane.
- Goal: Separate this data into two groups using a line.
- Strategy: Find the "street" which separates the data into two groups such that the street is as wide as possible and the equation that would correspond to the "median" of this street. Where a point is relative to this median will make our decision.
- Intuition: The points in the street "gutters," immediately on the sides of the street, might help us tell this story.

Updated Visualization of Problem

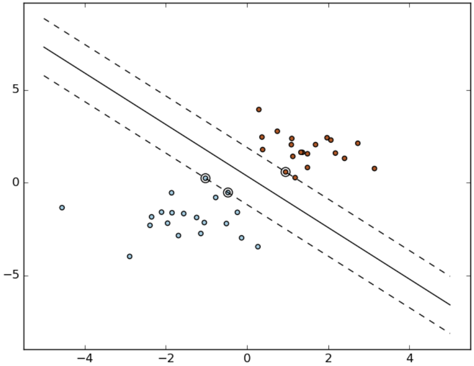

Image source: http://scikit-learn.org/stable/auto_examples/svm/plot_separating_hyperplane.html

SVM Derivation (Sketch)

- Consider $\boldsymbol{w}$, perpendicular to the median
- Also consider $\boldsymbol{u}$, which we would like to classify
- Dot product $\boldsymbol{w} \bullet \boldsymbol{u}$ finds the scalar projection of $\boldsymbol{u}$ onto $\boldsymbol{w}$
- If this dot product $> c$, then it crosses the median line, and $\boldsymbol{u}$ classified as $+$ – we write:

$$\boldsymbol{w} \bullet \boldsymbol{u} \geq c \Rightarrow \boldsymbol{u} \text{ is } +, \text{ or}$$

$$\boldsymbol{w} \bullet \boldsymbol{u} + b \geq 0 \Rightarrow \boldsymbol{u} \text{ is } +,$$

where $c = -b$.

SVM Derivation (Sketch)

- Good first try at a decision rule, but not unique
- If our decision rule is good, then if we know which data points are $+$ and which are $-$, then our rule should classify those points as $+$ and $-$, respectively, every time:

$$\boldsymbol{w} \bullet \boldsymbol{x}_{+} + b \geq 1$$
$$\boldsymbol{w} \bullet \boldsymbol{x}_{-} + b \leq -1$$

SVM Derivation (Sketch)

- Introduce $y_i = 1$ for $+$ data and $y_i = -1$ for $-$ data
- This preserves the first equation for the $+$ data. However, for the $-$ data, this flips the inequality and makes the -1 into 1
- The two equations are now identical for any $\boldsymbol{x}$:

$$y_i(\boldsymbol{x} \bullet \boldsymbol{w} + b) \geq 1, \text{ or}$$
$$y_i(\boldsymbol{x} \bullet \boldsymbol{w} + b) - 1 \geq 0,$$

for a data point $\boldsymbol{x}$ that is in a gutter. How convenient!

SVM Derivation (Sketch)

- Consider two gutter vectors $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$
- To find the width of the street:
  - Consider the difference vector, $\boldsymbol{x}_+ - \boldsymbol{x}_-$
  - Dot product of this and a unit vector in the direction of the median ($\boldsymbol{w}$) will give us the width of the street
  - $\boldsymbol{w}$ now needs to be a unit vector – divide it by its magnitude
- Express the width of the street W as

$$W = (\boldsymbol{x}_+ - \boldsymbol{x}_-) \bullet \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} = \frac{(1 - b) + (1 + b)}{\|\boldsymbol{w}\|} = \frac{2}{\|\boldsymbol{w}\|}$$

Updated Visualization of Problem

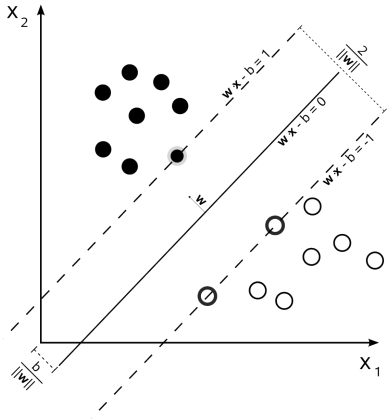

Image source: Open source via Wikibooks

SVM Derivation (Sketch)

- Widest street possible: maximize $2\|\boldsymbol{w}\|^{-1}$ subject to given constraints
- Same as maximizing $\|\boldsymbol{w}\|^{-1}$ subject to the given constraints
- Same as minimizing $\|\boldsymbol{w}\|$ subject to the given constraints
- (Note: width is naturally nonnegative and $f(w) = \frac{1}{2}w^2$ monotone increasing for $w \geq 0$)
- (Ahh, convexity...)
- Same as minimizing $\frac{1}{2}\|\boldsymbol{w}\|^2$ subject to the given constraints

SVM Derivation (Sketch)

- How do we solve this minimization problem?
  - QP
  - Verifying the KKT conditions
  - Lagrange multipliers
- We use MLM:

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_i \alpha_i[y_i(\boldsymbol{x}_i \bullet \boldsymbol{w} + b) - 1],$$

  where the constraints in brackets are restricted to be zero
- Differentiate L with respect to the vector $\boldsymbol{w}$ and the constant b

SVM Derivation (Sketch)

$$\frac{d}{d\mathbf{w}}L = \mathbf{w} - \sum_i \alpha_i y_i \mathbf{x}_i = 0 \qquad \Rightarrow \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$\frac{d}{db}L = \sum_i \alpha_i y_i = 0 \qquad \Rightarrow 0 = \sum_i \alpha_i y_i$$

SVM Derivation (Sketch)

- Placing these solutions back into L, we obtain

$$L = \frac{1}{2}\left[\sum_i \alpha_i y_i \mathbf{x}_i\right] \bullet \left[\sum_j \alpha_j y_j \mathbf{x}_j\right] - \left[\sum_i \alpha_i y_i \mathbf{x}_i\right] \bullet \left[\sum_j \alpha_j y_j \mathbf{x}_j\right] - \sum_i \alpha_i y_i b - \sum_i \alpha_i$$

SVM Derivation (Sketch)

$$L = -\sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i \bullet \boldsymbol{x}_j)$$

Note: This complicated math has actually given us some insight into how the solution is developed. Our maximum only depends on dot products of data vectors.

SVM Derivation (Sketch)

- So, suppose we were given an unknown vector $\boldsymbol{u}$ to classify. Then, our decision rule states

$$\sum_i \alpha_i y_i(\boldsymbol{x}_i \bullet \boldsymbol{u}) + b \geq 0 \Rightarrow \boldsymbol{u} \text{ is } + .$$

- This decision rule is fine in the ideal case of linear separability...

SVM Derivation (Sketch)

- ...what happens if the data are not linearly separable?
- Transform the data into a space where it *is* separable
- Consider the transformation $\phi(\boldsymbol{x})$
- We saw that our decision rule takes dot products into accounts
- Better to understand $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i) \bullet \phi(\boldsymbol{x}_j)$ than $\phi(\boldsymbol{x})$
- Dr. Winston: "This is a miracle."

SVM Derivation (Sketch)

- We call K the kernel function
- When data not linearly separable, we must specify kernel
- Some popular kernels, for two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$:
  - linear kernel: $K(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u} \bullet \boldsymbol{v} + 1)^n$
  - radial basis kernel: $K(\boldsymbol{u}, \boldsymbol{v}) = \exp\{\frac{\|\boldsymbol{u}-\boldsymbol{v}\|}{\sigma}\}$.

Updated Visualization of Problem

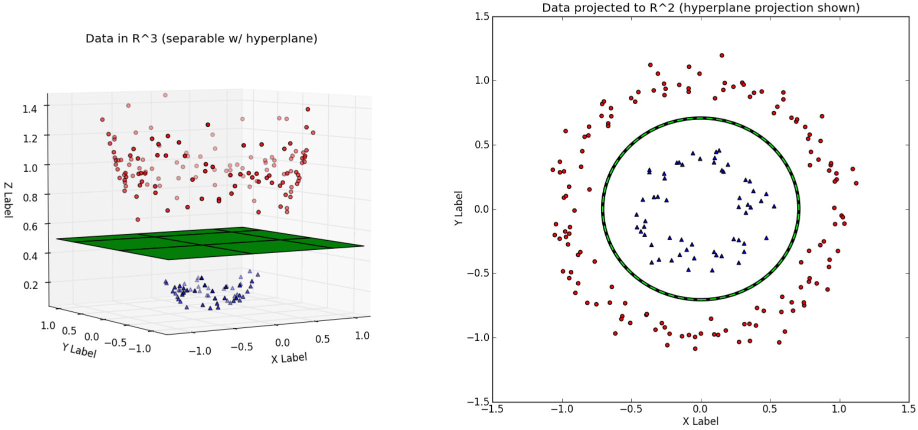

Image source: http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Generalization to Higher Dimensions

- Dr. Dickey shared several examples with $\phi(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$.
- We found a plane that separated the data, then projected that plane in three-space back into a line in the Cartesian plane.
- We eventually run out of dimensions to visualize and name.
- Hence, traditional SVM lingo talks about a "separating hyperplane" in "hyperspace," even if we have easier nomenclature.

Where does SVM get its name?

- In actuality, the separating hyperplane is usually determined by only a handful of data points.
- The points that help determine the hyperplane are called "support vectors."
- The hyperplane itself is a classifying "machine."

What can I understand now?

- Check out SLG's other SVM talks from Nov. 2014!
- Jami's talk is a great introduction to the notation used by *ESL*.
- Brian shows how to use SVM to obtain rankings.
- Huimin shows the interaction of SVM and LDA.
- All the terminology has been motivated in this talk.

Implementation in R

- R package: `e1071` (most recent update: 2015)
- Package written by David Meyer et. al., TU Wien, Austria
- Package based on C++ implementation, `libsvm`, by Chang and Lin (2001)
- Some examples in RMarkdown file

References

- Chang, C. C. and Lin, C. J. (2001). LIBSVM: a library for support vector machines.

  Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm

  Detailed documentation (algorithms, formulae, . . .) can be found at http://www.csie.ntu.edu.tw/ cjlin/papers/libsvm.ps.gz
- Cortes, C. and Vapnik, V. (1995). Support-vector network. Machine Learning, 20, 125.
- L., Melvin. "Support Vector Machines (SVM) Overview and Demo using R." YouTube, 5/22/16. Accessed 9/20/16.
- Meyer, David. (2015). Support Vector Machines: The interface to `libsvm` in package `e1071`. R Project.
- Winston, Patrick H. *6.034, Artificial Intelligence, Fall 2010*. (MIT OpenCourseWare: Massachusetts Institute of Technology.) https://www.youtube.com/watch?v=_PwhiWxHK8o (Accessed 9/20/16). License: Creative commons BY-NC-SA