

LECTURE 21:

SUPPORT VECTOR MACHINES PT. 2

November 29, 2017

SDS 293: Machine Learning

Announcements 1/3

Interested in
CS Grad School?



Casey Fiesler

UC Boulder



Lane Harrison

WPI



Shannon Roberts

UMass



Samantha Williams

UVM

Friday 12/1 at noon
Ford 241

P.I.Z.Z.A

010010111

Announcements 2/3

- FP2 feedback:
 - If you submitted **before** break, went out this morning
 - If you submitted **after** break, will be out by tomorrow
- FP3 has been posted, including a detailed rubric

Announcements 3/3




The image shows a poster template with the following sections:

- Title**
- Authors**
- Introduction**: Introduce the problem here. Include any relevant details about the datasets.
- Results**: What did you discover? Definitely the most important section! Be sure to clearly label any figures, etc.
- Data & Methods**: Describe the data you used, and how you modeled it here.
- Conclusions & Future Work**: Sum up the project, and talk about what you'd like to do if you had more time to continue working on this.
- References**: Cite your data source and any additional papers, books, etc. here.
- Acknowledgements**: This project was completed in partial fulfillment of the requirements of SDS293: Machine Learning. This course is offered by the Statistical and Data Sciences Program at Sonoma College, and was taught by R. Jordan Crosser during Spring 2016.

A logo with a stylized 'S' and 'D' is located at the bottom center of the template.

- FP poster template is now available on Moodle
- You're welcome to make changes / reformat; just please keep it 3'x4' portrait
- **Printing:**
 - **Option 1:** upload PDF to Moodle on or before ~December 7th
 - **Option 2:** arrange printing on your own (Paradise Copies, etc.)

Outline

- Maximal margin classifier
 - Support vector classification
 - 2 classes, linear boundaries
 - 2 classes, nonlinear boundaries
 - Multiple classes
 - Comparison to other methods
 - Lab
- 

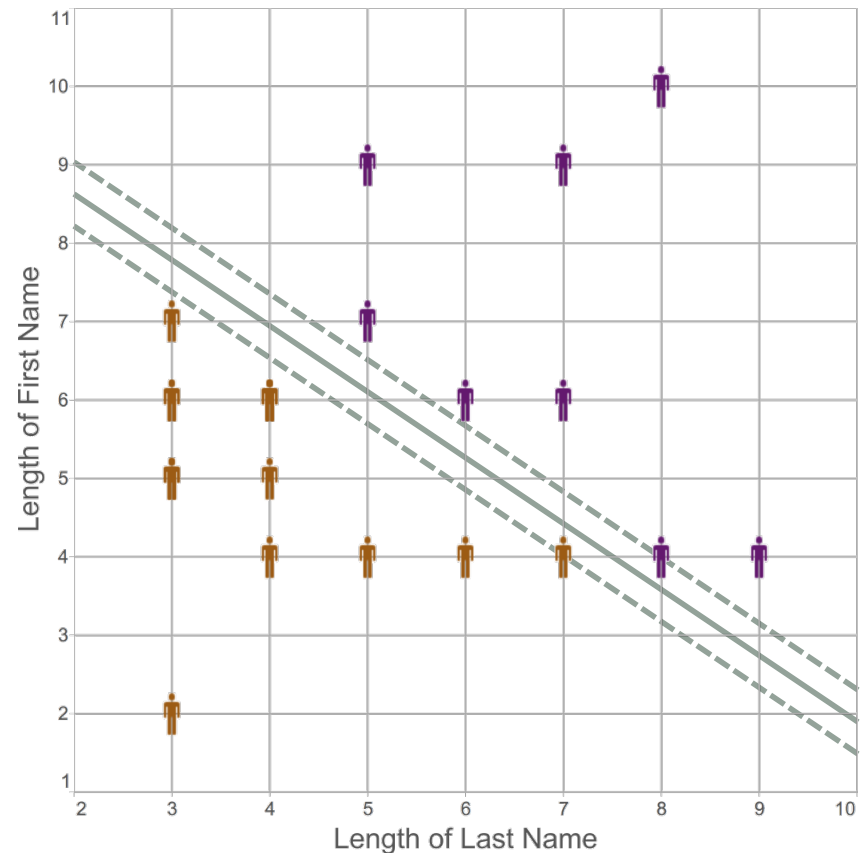
Recap: maximal margin classifier

Big idea:

find the dividing hyperplane
with the **widest margin**

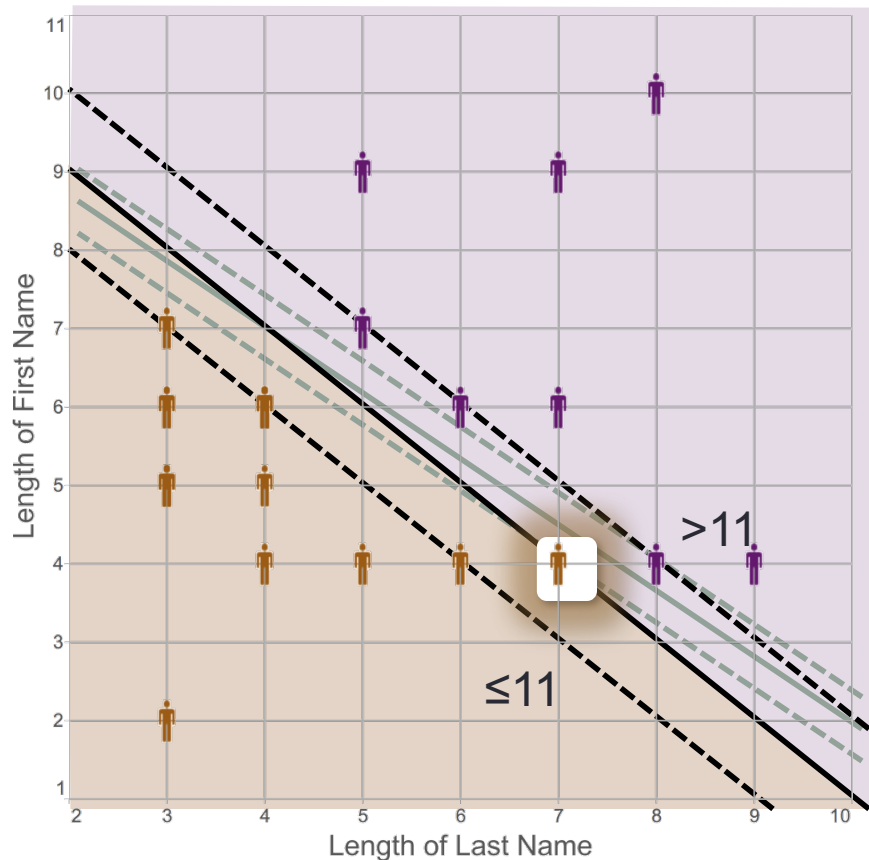
$\max(M)$ such that

$$y_i (\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq M$$



Problem: sometimes we can't be perfect...

Recap: support vector classifier



Big idea:
we might be willing to **sacrifice a few**
in order to give the rest
a better margin

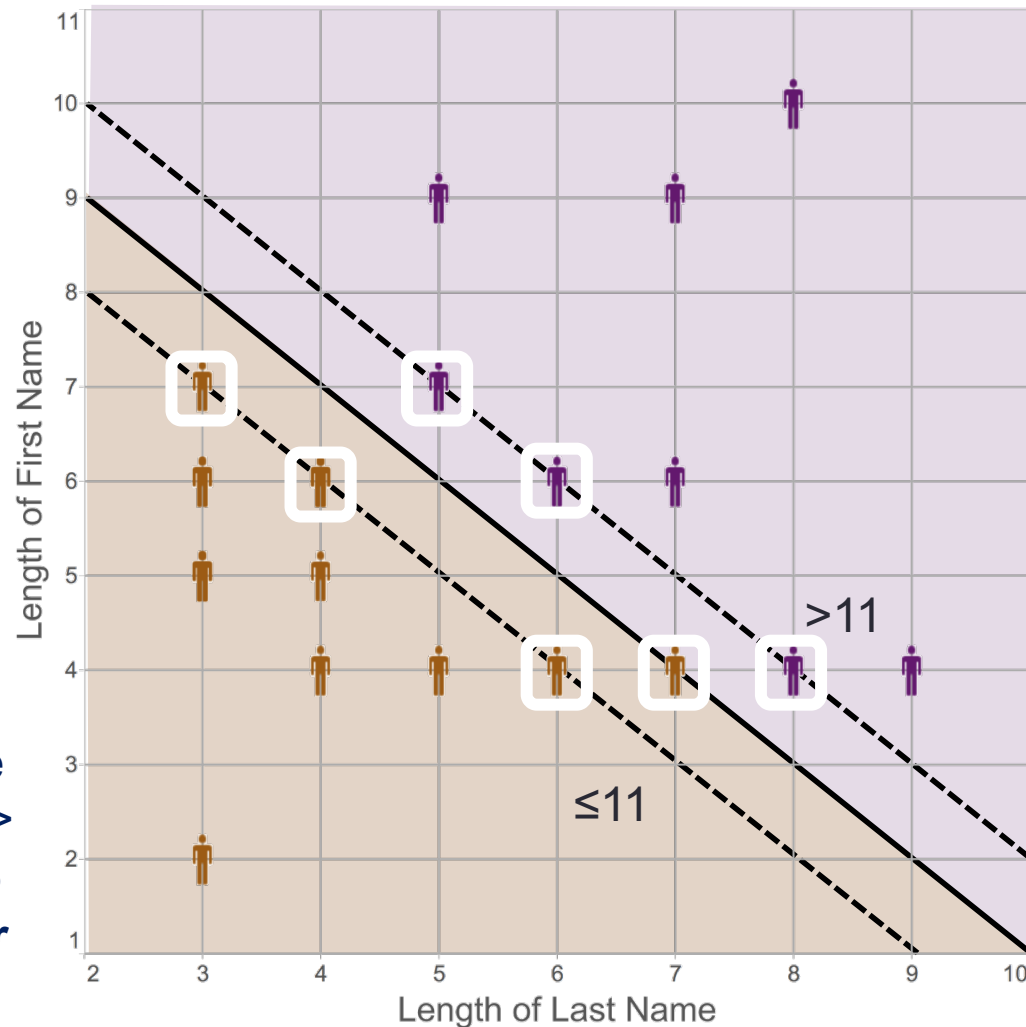
$\max(M)$ such that

$$y_i (\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq M (1 - \varepsilon_i)$$

$$\varepsilon_i \geq 0, \quad \sum_{i=1}^n \varepsilon_i \leq C$$

Bigger value = further from margin = **more confident**

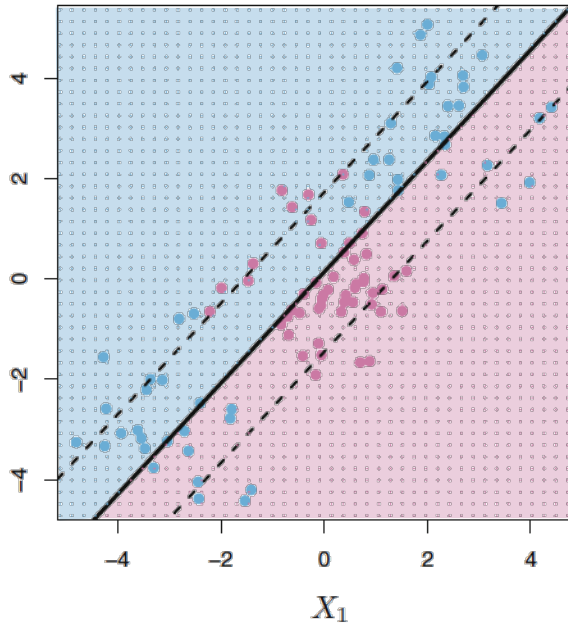
Recap: support vectors



Decision rule is based only on the support vectors => **SVC is robust to strange behavior far from the hyperplane!**

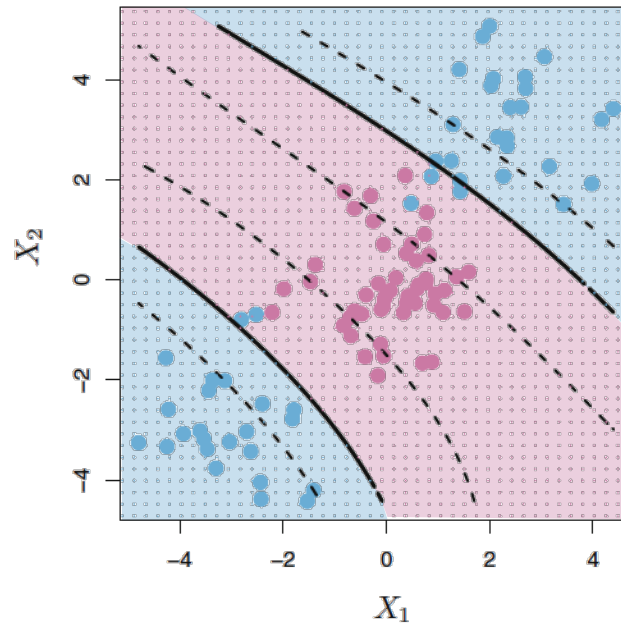
Recap: kernel

Linear



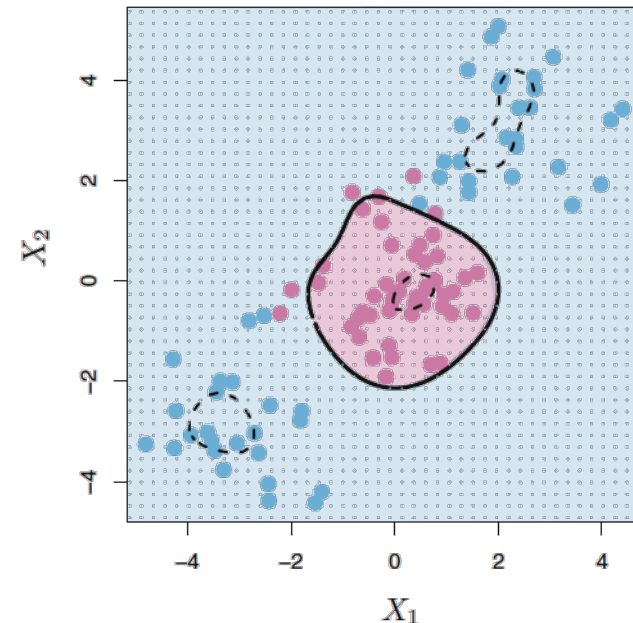
$$K(x_i, x_{i'}) = \sum_{j=1}^p x_{ij} x_{i'j}$$

Polynomial



$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^p x_{ij} x_{i'j} \right)^d$$

Radial



$$K(x_i, x_{i'}) = e^{-\gamma \sum_{j=1}^p (x_{ij} - x_{i'j})^2}$$

Big idea: using different ways of measuring “similarity” allows you to partition the feature space in different ways

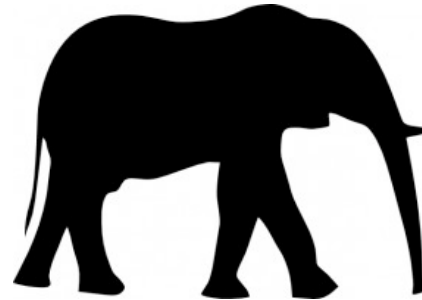
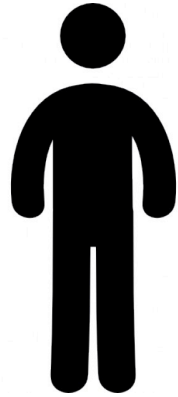
Discussion

- **Question:** what's the problem?
- **Answer:** regardless of kernel shape, a SMV can only divide the data into **two** parts... but the data in the real world sometimes has multiple classes

So what can we do?



Goal: assign observation to 1 of 4 groups



One-versus-one classification

- **Big idea:** build an SVM for each pair of classes



- **To predict:** classify each observation using all of the $(k \text{ choose } 2)$ classifiers, keep track of how many times the observation is assigned to each: majority wins

One-versus-all classification

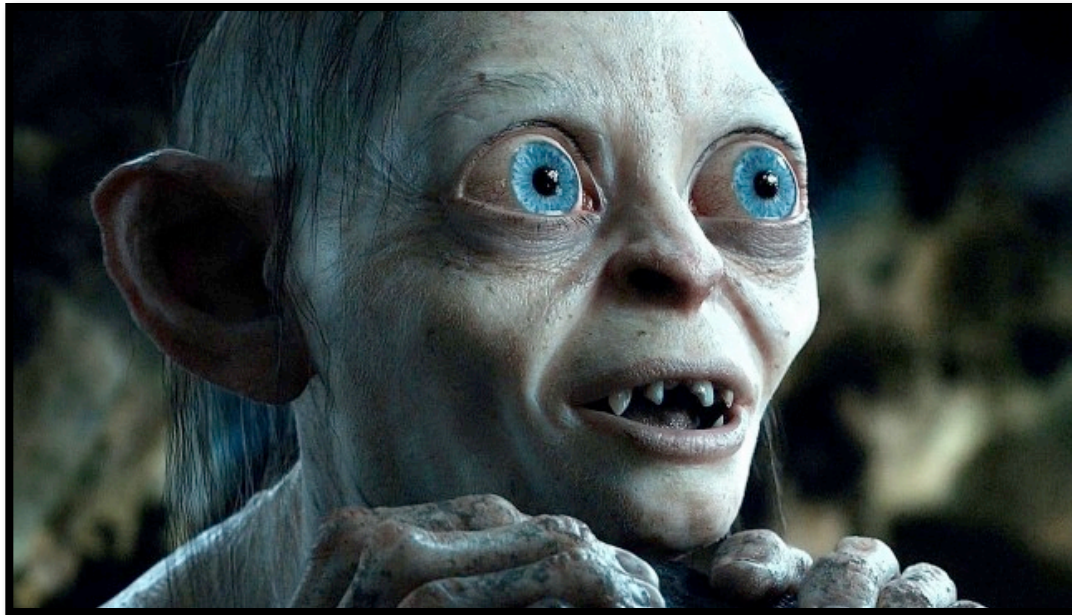
- **Big idea:** build an SVM for each class



- **To predict:** classify each observation using each of the k classifiers, keep track of how confident we are of each prediction: most confident class wins

Quick history lesson

- Mid-90s: SVMs come on the scene
- **Reaction:** OoOoOoooOooooohh...
 - Good performance
 - “Mysterious”– felt entirely different from classical approaches



Flashback: loss functions

- Many of the methods we've seen so far take the form:

$$\min_{\beta} \{ L(X, y, \beta) + \lambda P(\beta) \}$$



“loss”

(consequence for poor prediction)



“penalty”

(consequence for being complicated)

- With a little manipulation, we can rewrite our SVM as:

$$\min_{\beta} \left\{ \sum_{i=1}^n \max \left[0, 1 - y_i \left(\beta_0 + \dots + \beta_p x_{ip} \right) \right] + \lambda \sum_{j=1}^p \beta_j^2 \right\}$$

hinge loss

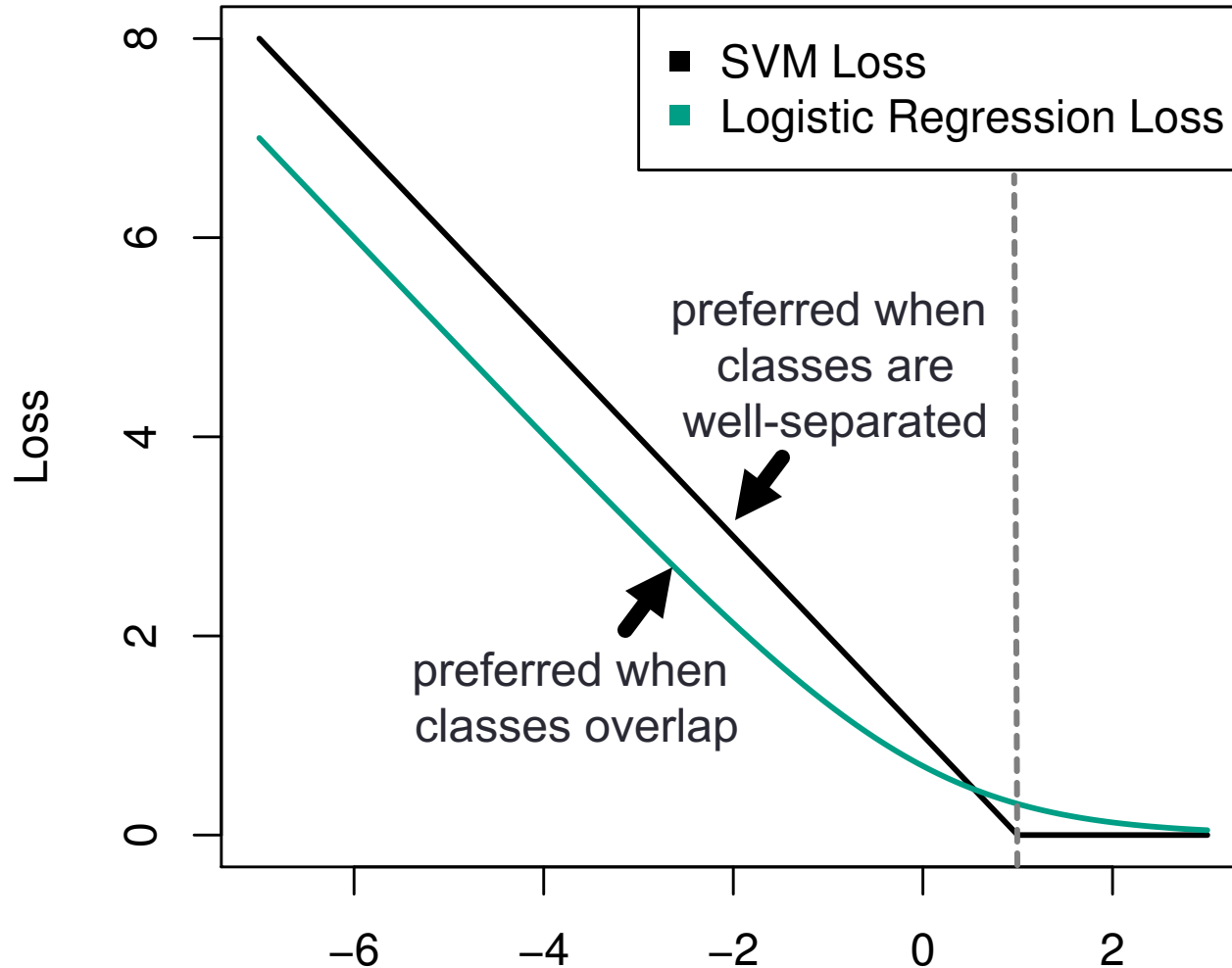
Alas...

- Despite appearances, SVMs are quite closely related to logistic regression and other classical statistical methods



- And what's worse, most other classification methods can use non-linear kernels, too! (recall Ch. 7...)

SVMs vs. logistic regression loss functions



$$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip})$$

Lab: Multiclass SVMs

- To do today's lab in R: `e1071`, `ROCR`
- To do today's lab in python: <nothing new>
- Instructions and code:

[\[course website\]/labs/lab15-r.html](#)

[\[course website\]/labs/lab15-py.html](#)

- Full version can be found beginning on p. 366 of ISLR

Coming up

- **Next week:** unsupervised techniques