Exercises in Machine Learning
Playing with Kernels

Zdeněk Žabokrtský, Ondřej Bojar
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University, Prague

Tue Mar 31, 2015

Outline

- Linear Kernel: $k(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$
- Polynomial Kernel: $k(\mathbf{x}, \mathbf{y}) = (\gamma * \mathbf{x} \cdot \mathbf{y} + \text{coeff0})^{\text{degree}}$
- RBF Kernel: $k(\mathbf{x}, \mathbf{y}) = exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2); \gamma > 0$

  ... including their parameters
- Cross-validation Heatmap
- Multi-class SVM
  - For the PAMAP-easy dataset.
  - Regularization parameters.
  - Inseparable classes.

Based on `http://scikit-learn.org/stable/modules/svm.html` and other scikit-demos.

Regularization (C) in linear SVM

$$k(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$$

(Linear kernel = no kernel)

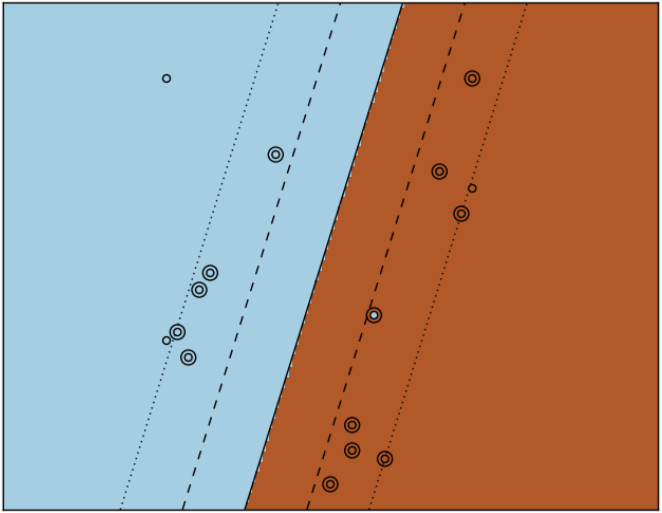
The parameter C in (linear) SVM:

- sets the weight of the sum of slack variables.
- serves as a regularization parameter.
- controls the number of support vectors.

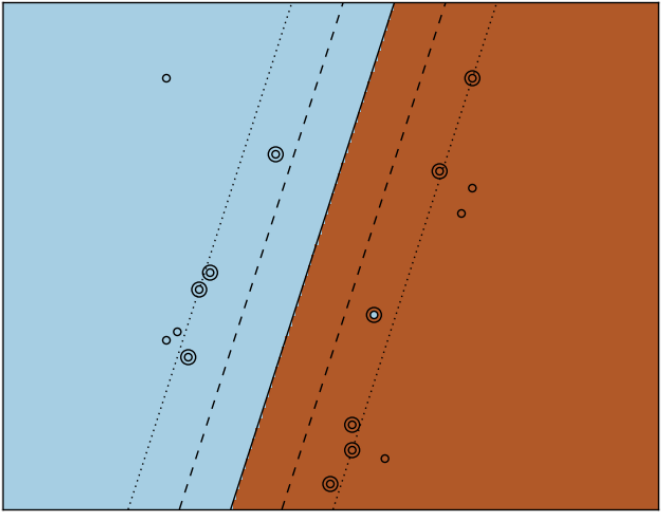
C	Penalty for Errors	Number of points considered	Bias	Variance
Low	Low	Many	High	Low
High	High	Few	Low	High

Think C for Varian$C$e.

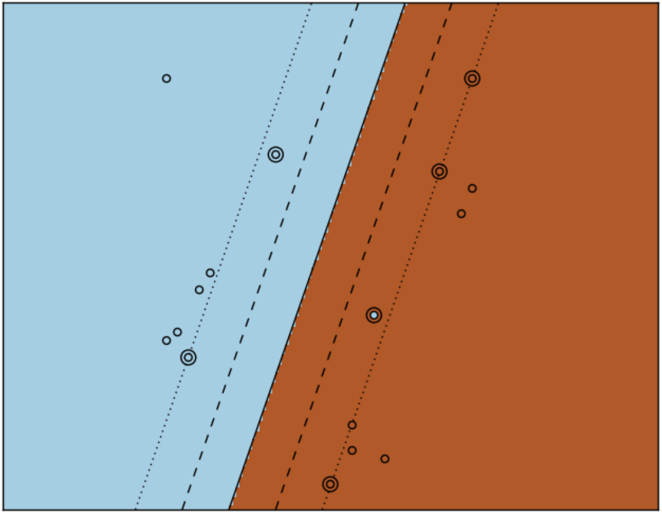
SVM Linear C=0.1



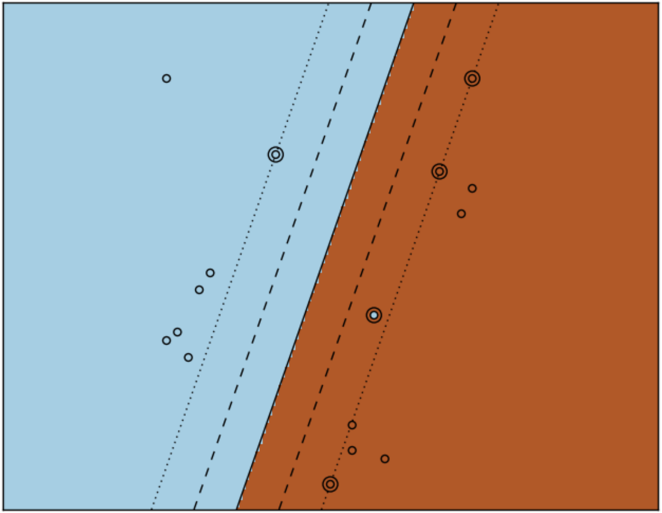
SVM Linear C=0.2



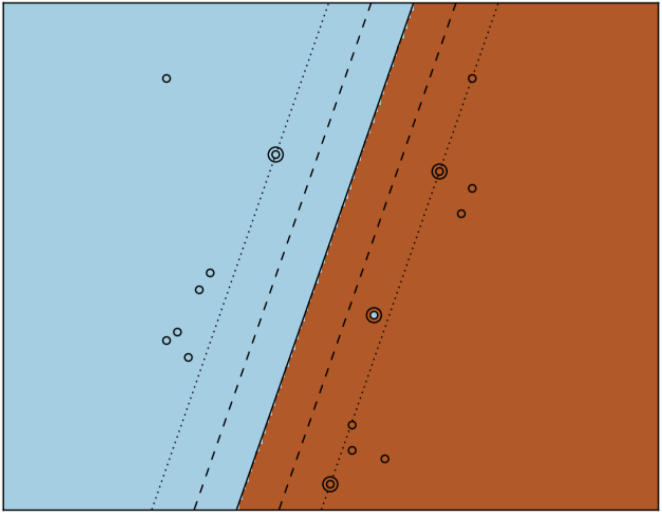
SVM Linear C=0.5



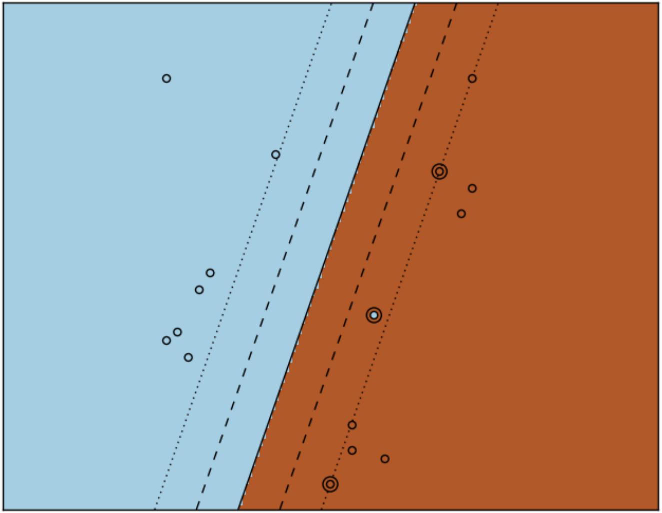
SVM Linear C=1



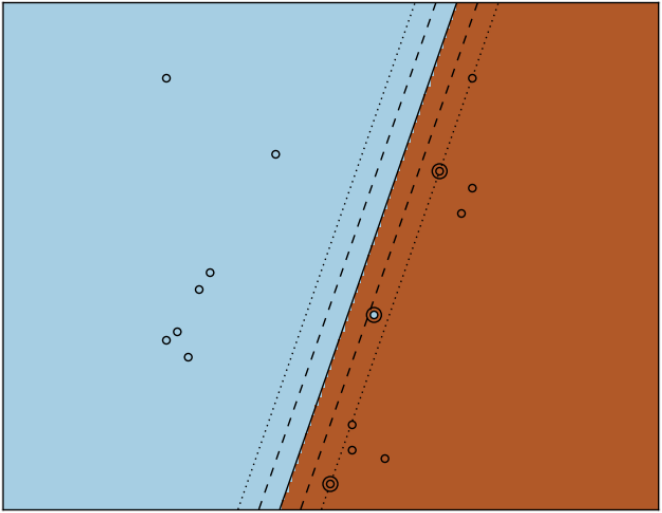
SVM Linear C=5



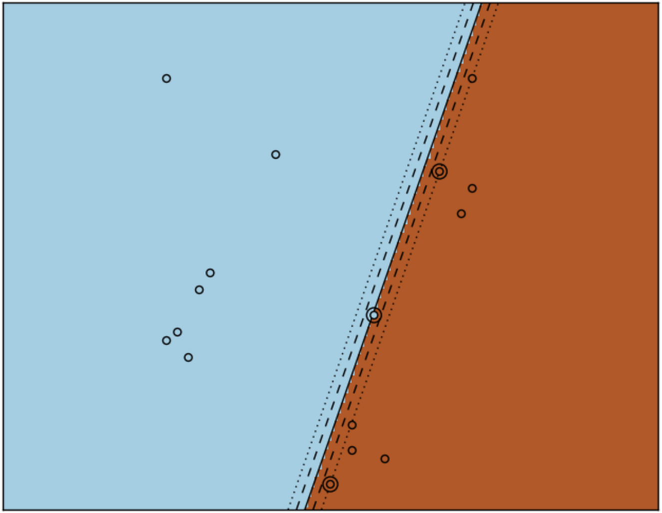
SVM Linear C=10



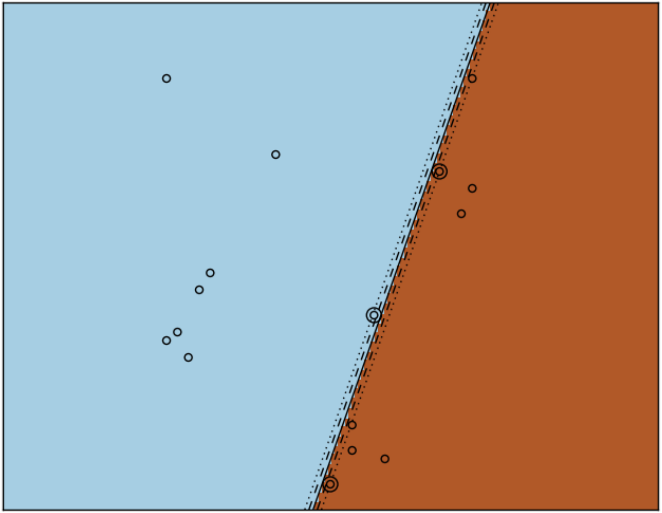
SVM Linear C=20



SVM Linear C=50



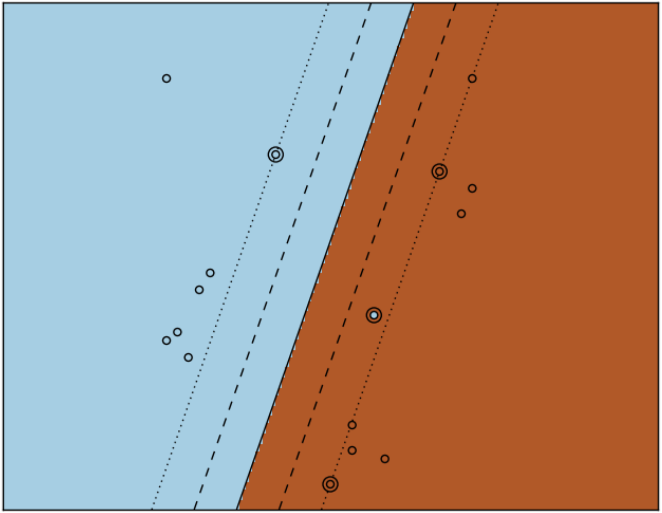
SVM Linear C=100



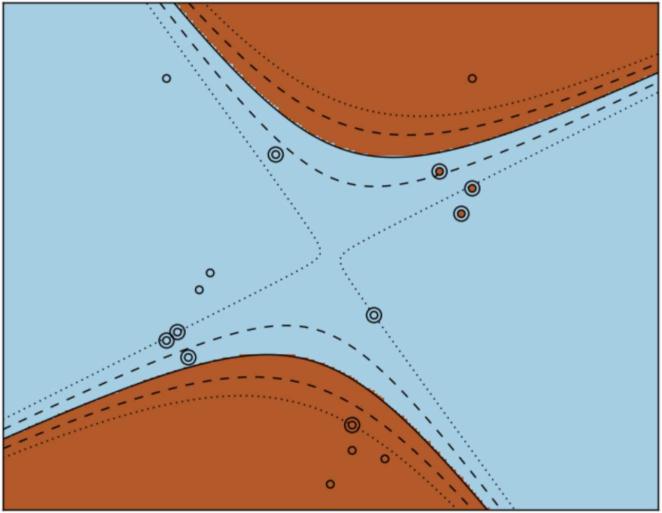
Polynomial Kernel

$$k(\mathbf{x}, \mathbf{y}) = (\gamma * \mathbf{x} \cdot \mathbf{y} + \mathsf{coeff0})^{\mathsf{degree}}$$

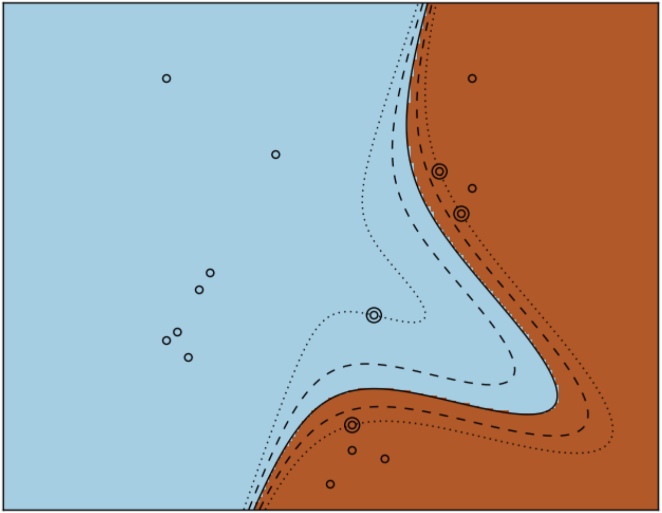
SVM Poly (degree 1)



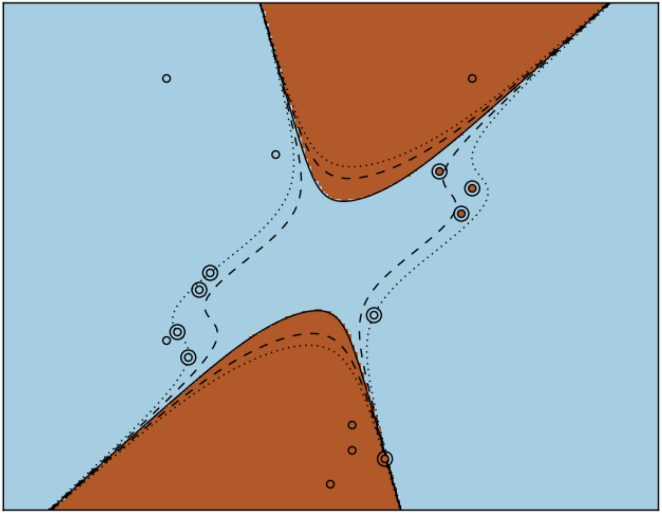
SVM Poly (degree 2)



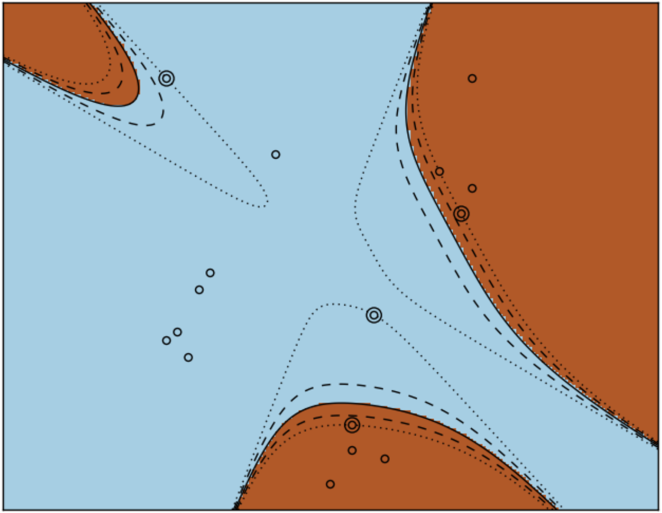
SVM Poly (degree 3)



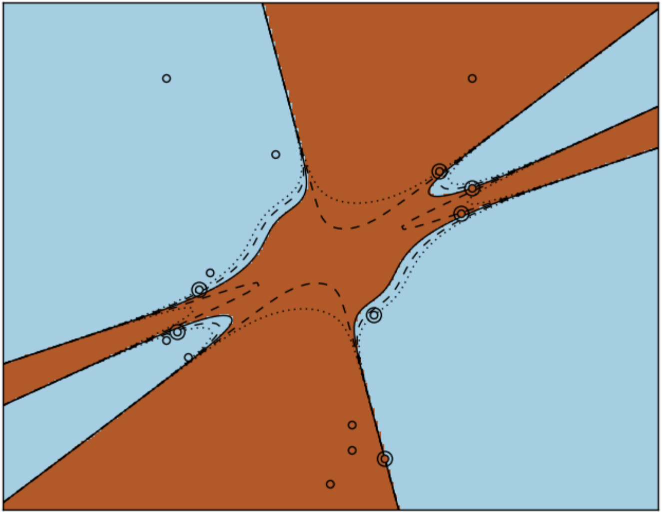
SVM Poly (degree 4)



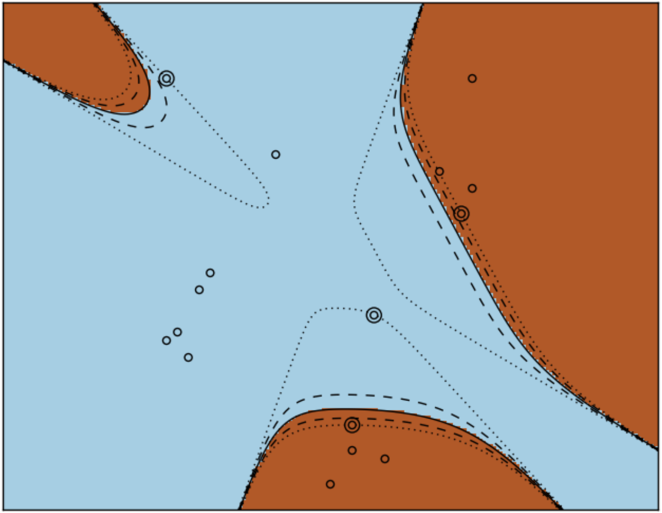
SVM Poly (degree 5)



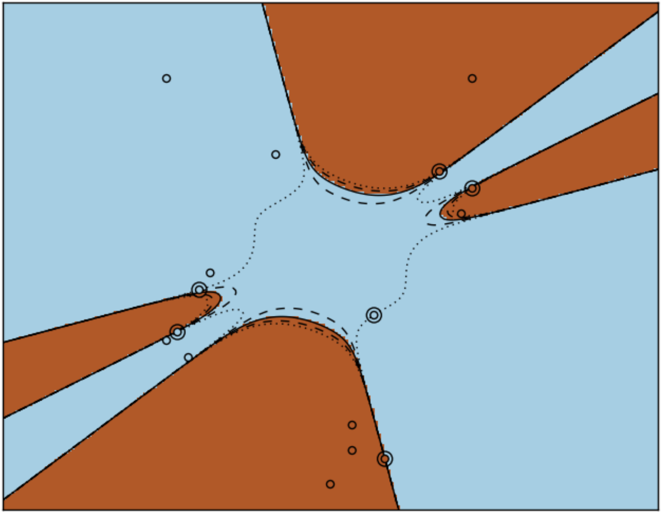
SVM Poly (degree 6)



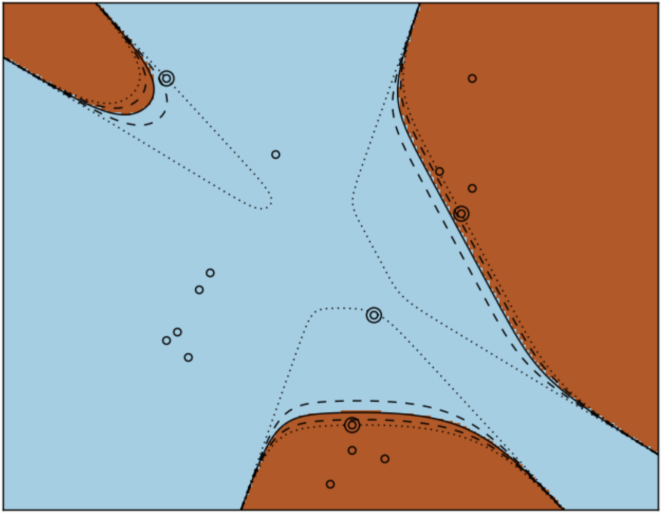
SVM Poly (degree 7)



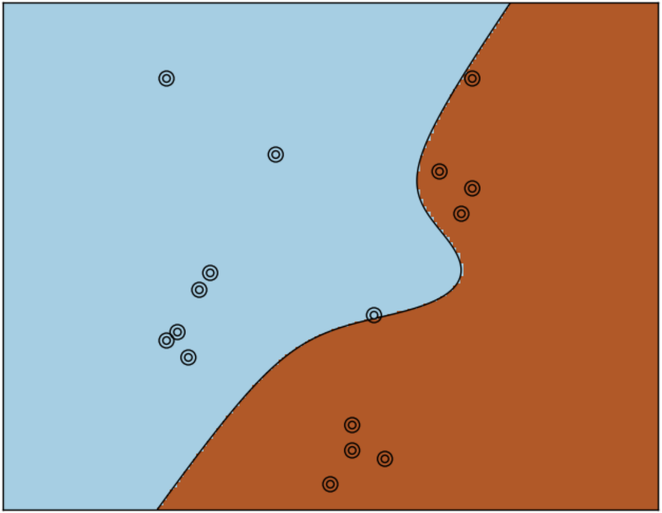
SVM Poly (degree 8)



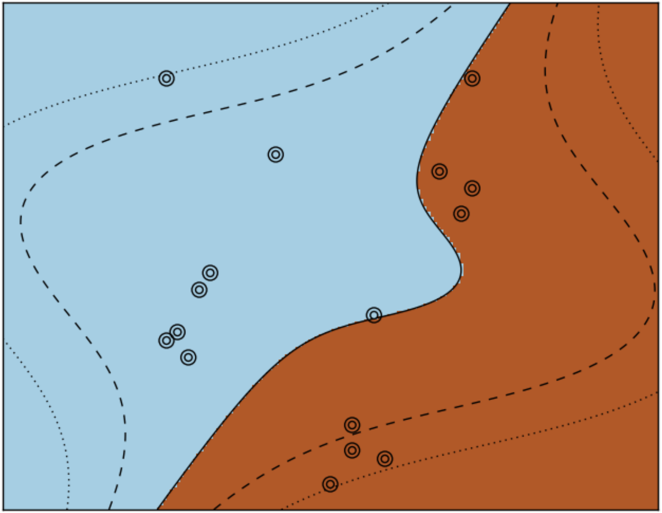
SVM Poly (degree 9)



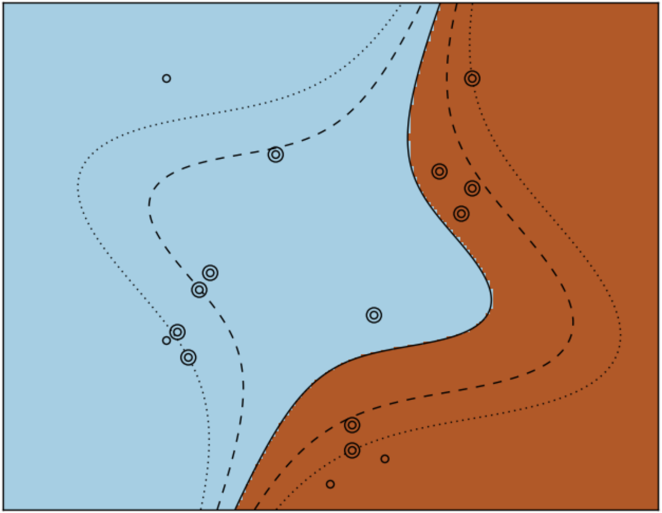
SVM Poly (degree 3, gamma 0.05)



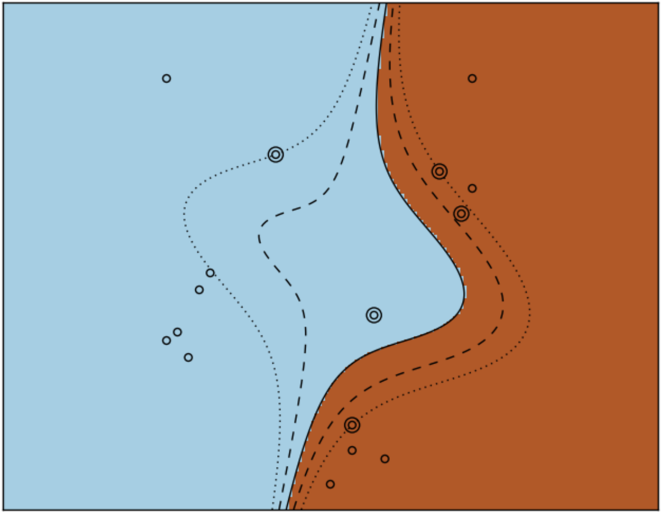
SVM Poly (degree 3, gamma 0.1)



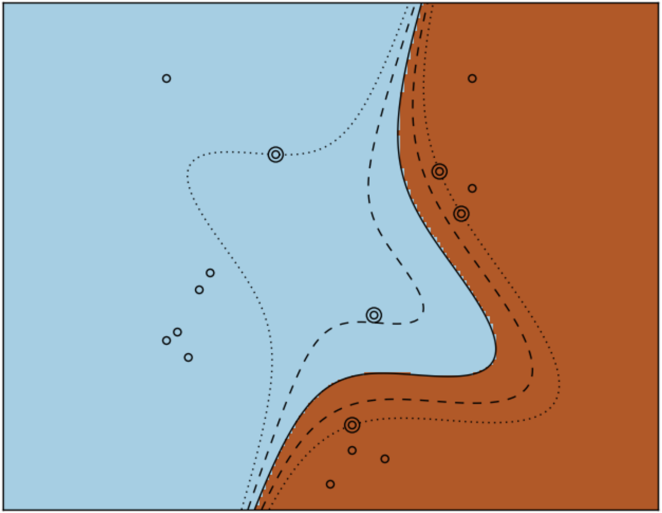
SVM Poly (degree 3, gamma 0.2)



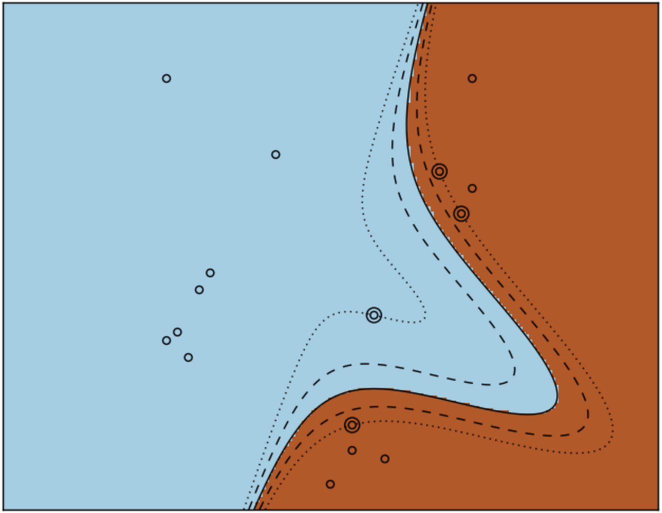
SVM Poly (degree 3, gamma 0.5)



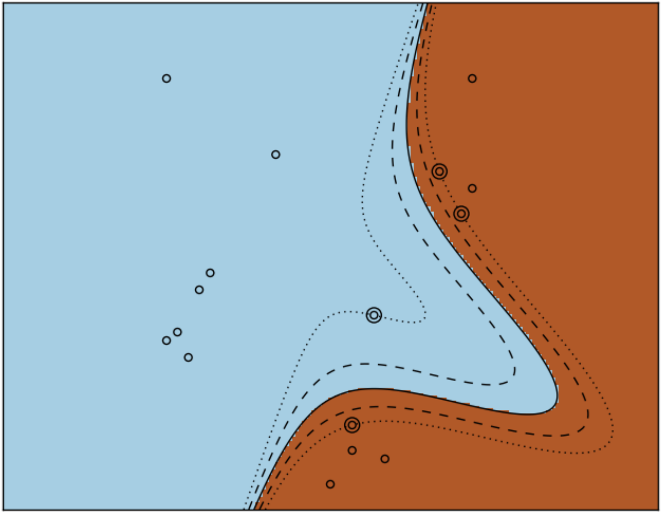
SVM Poly (degree 3, gamma 0.7)



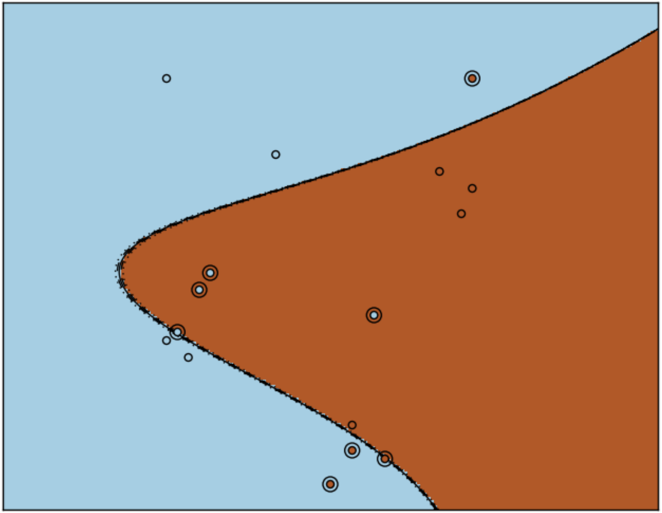
SVM Poly (degree 3, gamma 1)



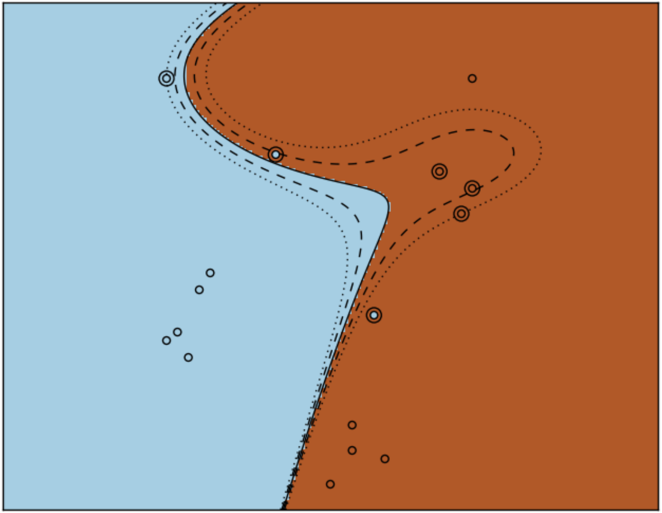
SVM Poly (degree 3, gamma 2)



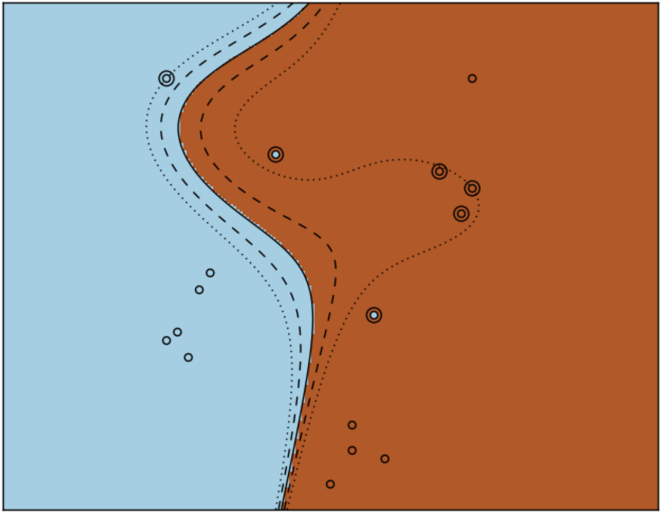
SVM Poly (d=3, g=0.5, coef=-2.0)



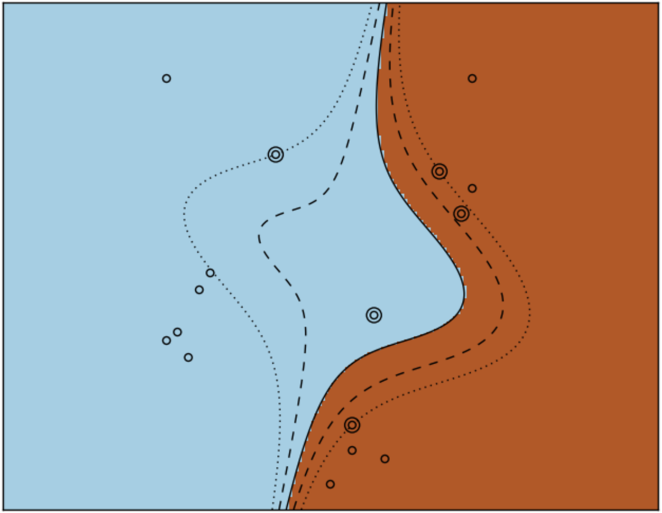
SVM Poly (d=3, g=0.5, coef=-1.0)



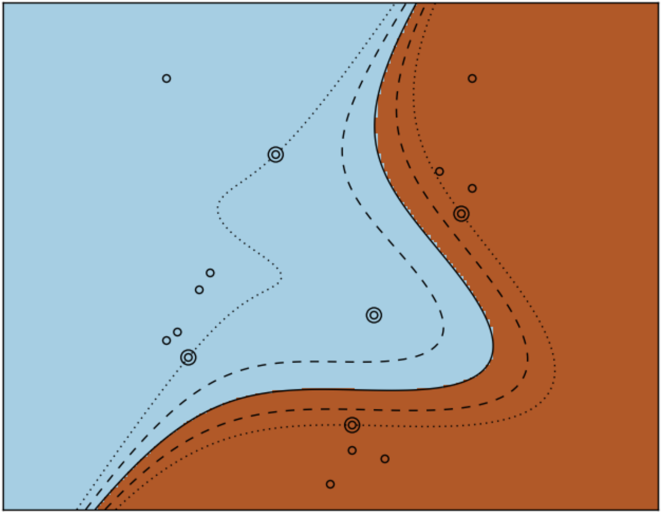
SVM Poly (d=3, g=0.5, coef=-0.50)



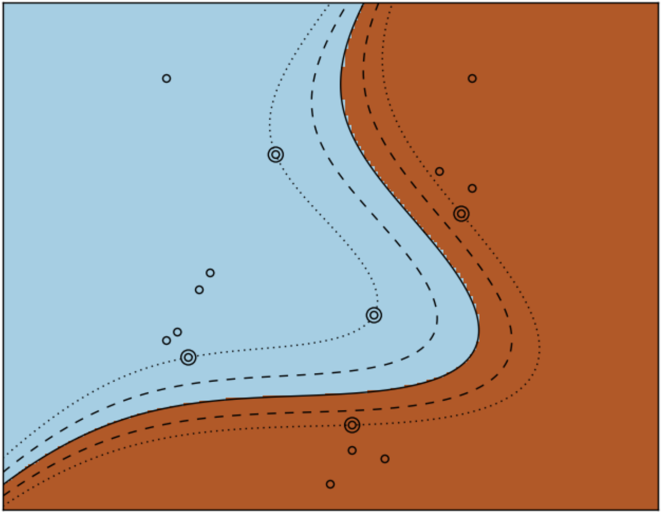
SVM Poly (d=3, g=0.5, coef=0)



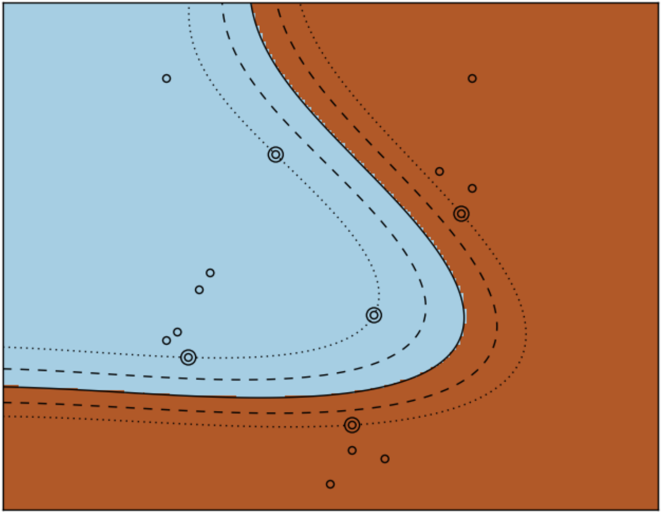
SVM Poly (d=3, g=0.5, coef=0.5)



SVM Poly (d=3, g=0.5, coef=1)

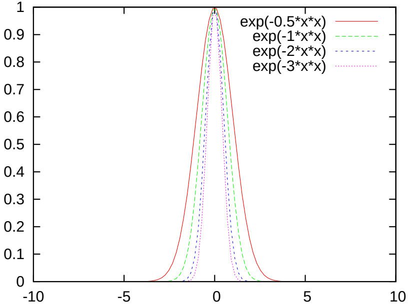


SVM Poly (d=3, g=0.5, coef=2)



RBF Kernels

$$k(\mathbf{x}, \mathbf{y}) = exp(-\gamma\|\mathbf{x} - \mathbf{y}\|^2); \gamma > 0$$



- Each training point creates its bell.
- Overall shape is the sum of the bells.
- Kind of “all nearest neighbours”.

RBF Kernel Parameters

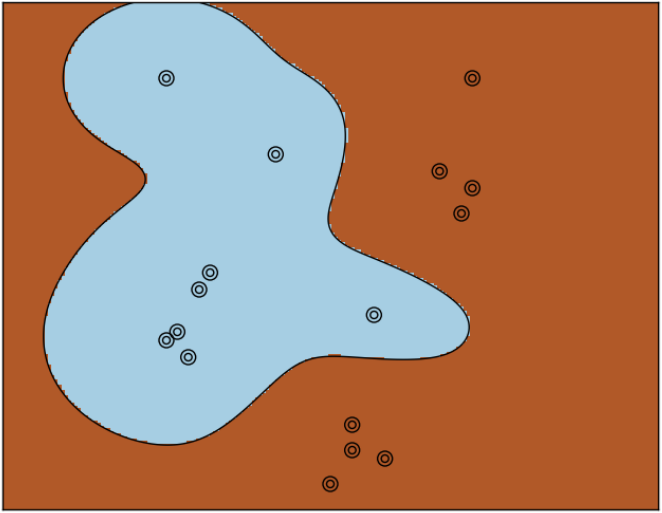
C	Decision Surface	Model	Bias	Variance
Low	Smooth	Simple	High	Low
High	Peaked	Complex	Low	High

gamma	Affected Points
Low	can be far from training examples
High	must be close to training examples

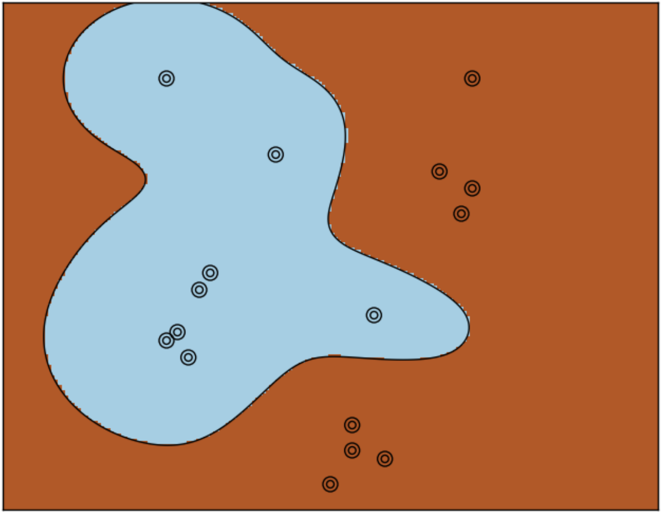
- Does higher gamma lead to higher variance?

- Choice critical for SVM performance.
- Advised to use GridSearchCV for C and gamma:
  - exponentially spaced probes
  - wide range

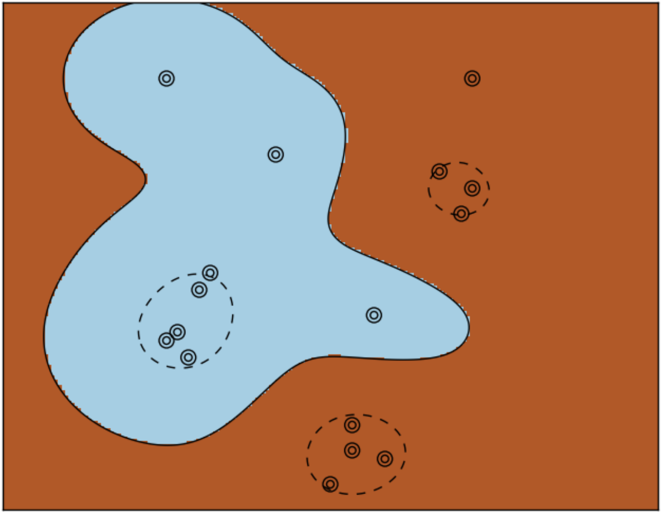
SVM RBF (C=0.05, gamma=2)



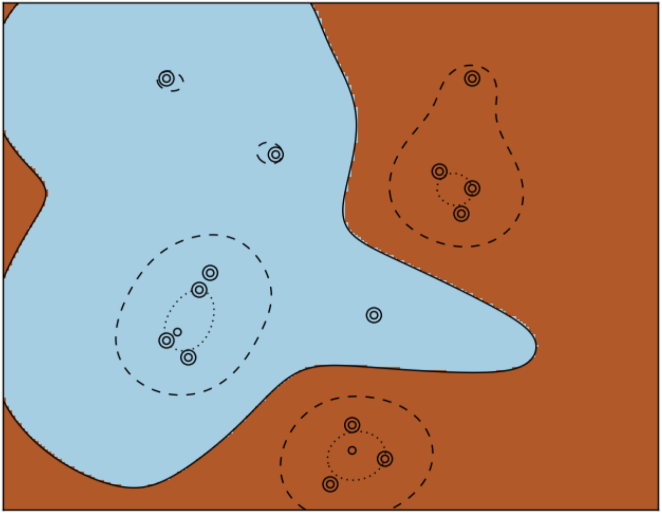
SVM RBF (C=0.1, gamma=2)



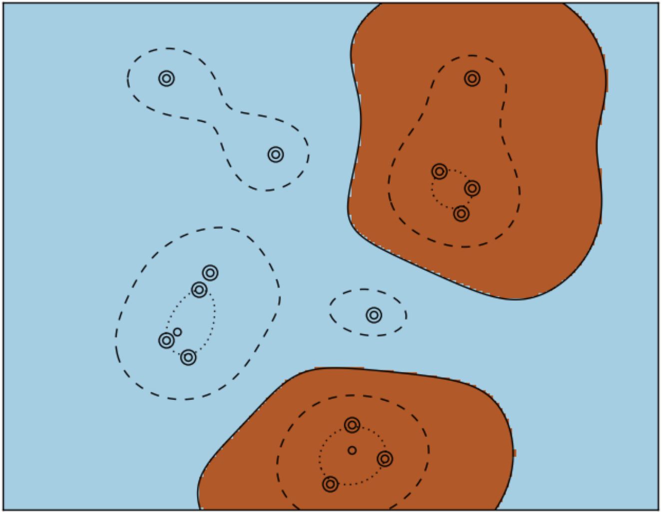
SVM RBF (C=0.2, gamma=2)



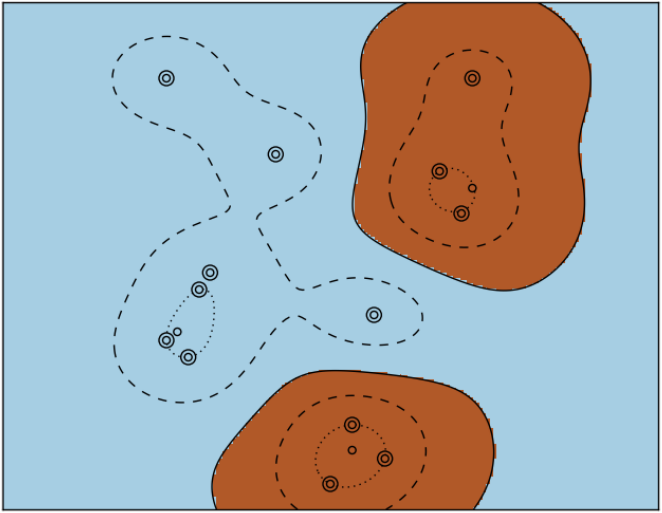
SVM RBF (C=0.5, gamma=2)



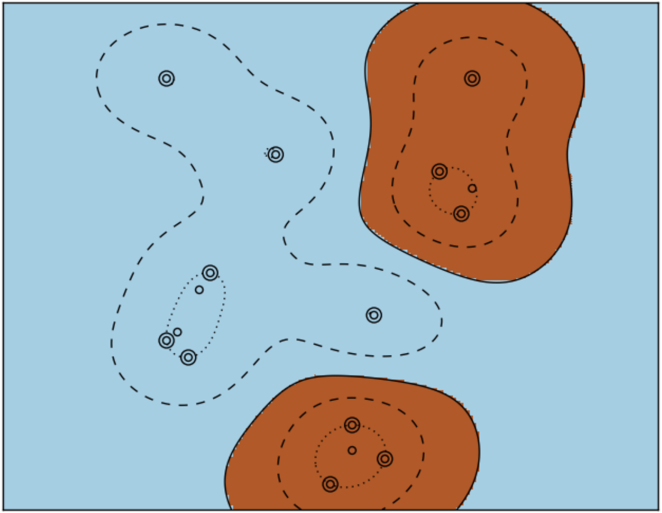
SVM RBF (C=0.6, gamma=2)



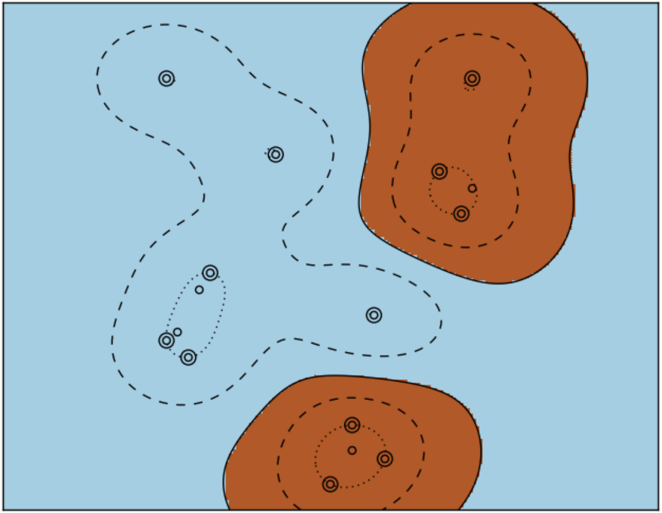
SVM RBF (C=0.7, gamma=2)



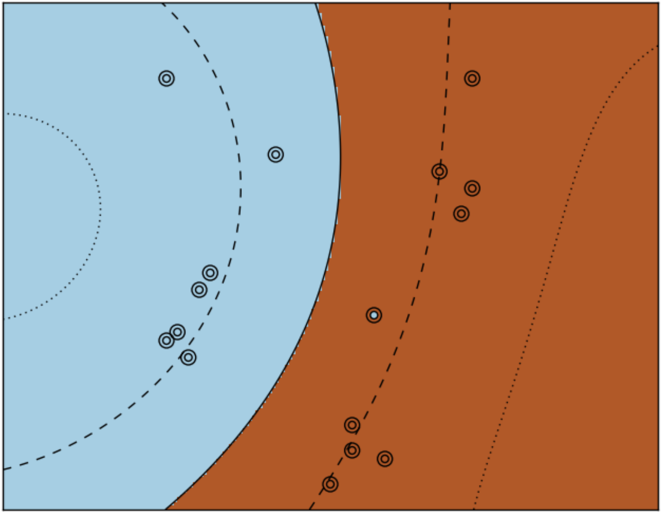
SVM RBF (C=1, gamma=2)



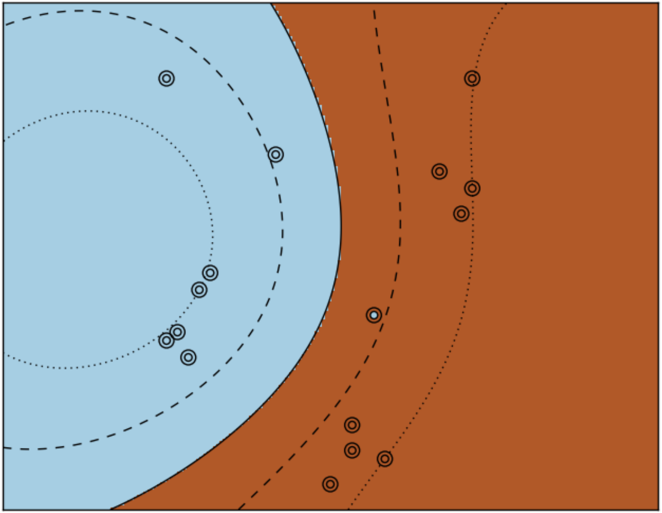
SVM RBF (C=2, gamma=2)



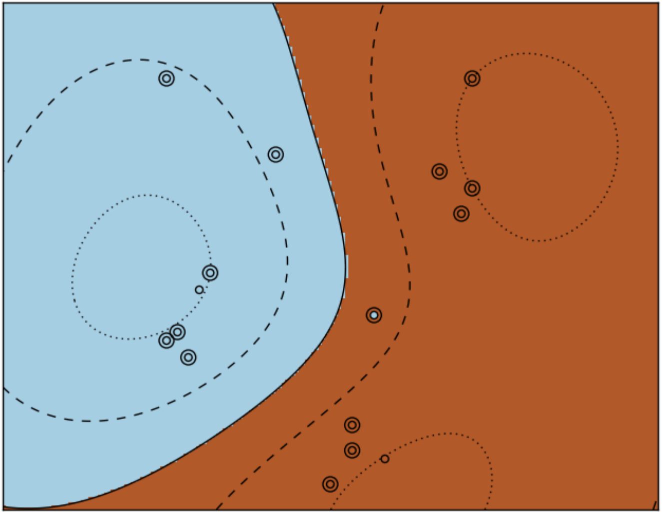
SVM RBF (C=0.5, gamma=0.05)



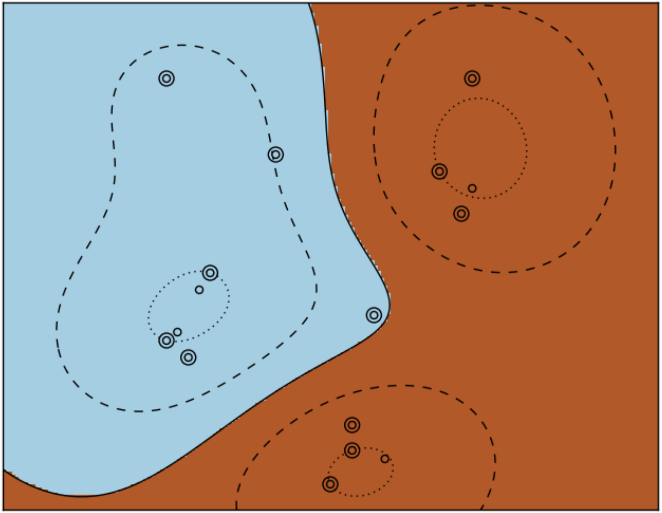
SVM RBF (C=0.5, gamma=0.1)



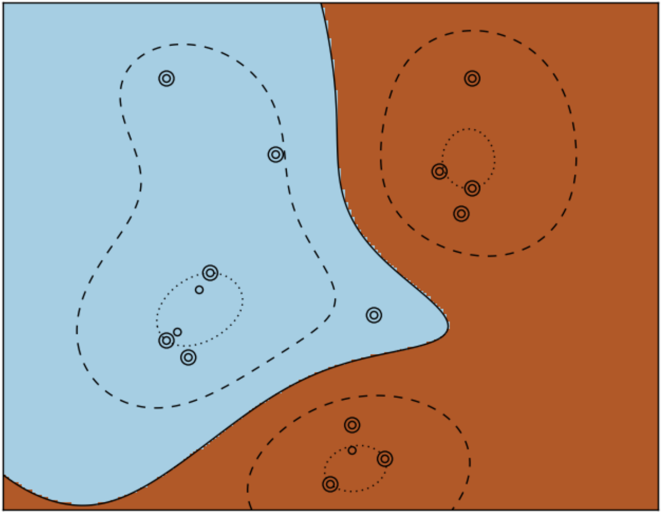
SVM RBF (C=0.5, gamma=0.2)



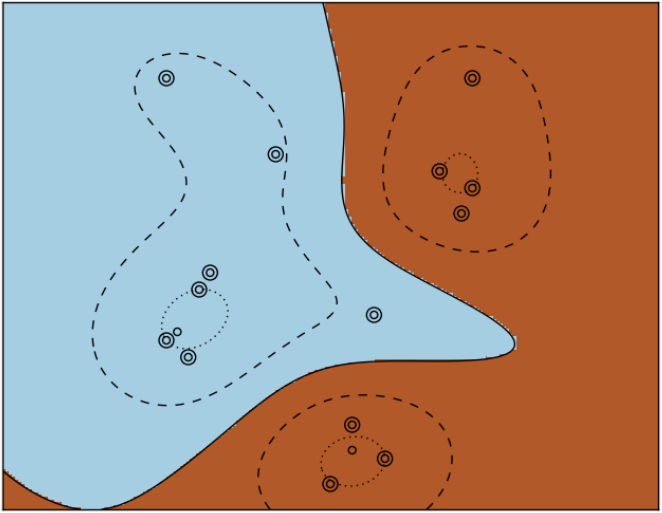
SVM RBF (C=0.5, gamma=0.5)



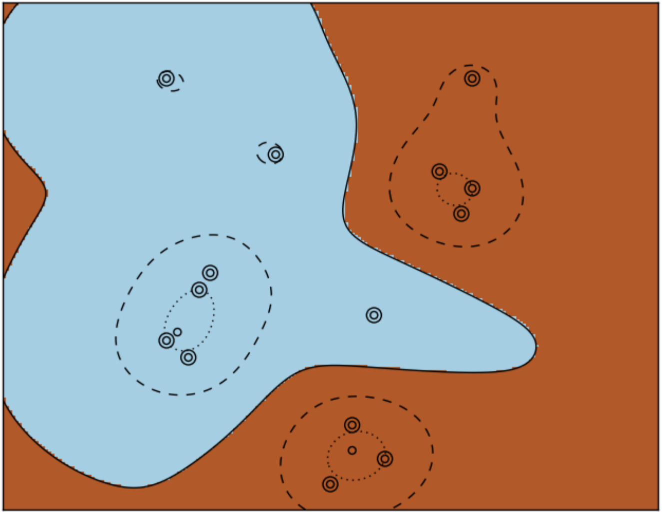
SVM RBF (C=0.5, gamma=0.7)



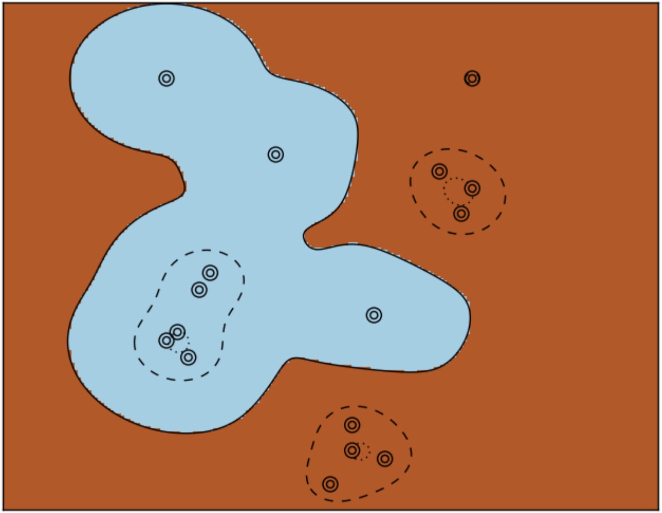
SVM RBF (C=0.5, gamma=1)



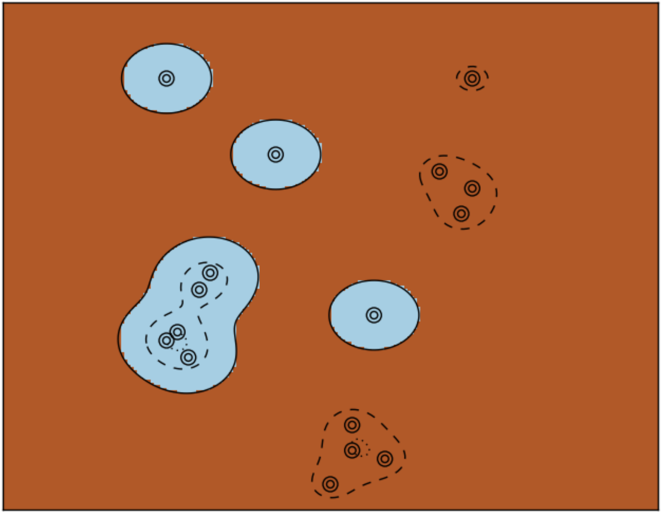
SVM RBF (C=0.5, gamma=2)



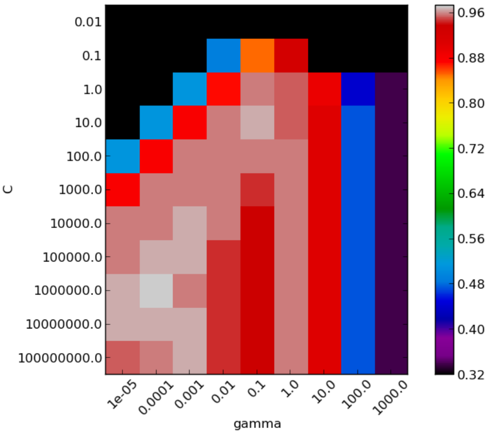
SVM RBF (C=0.5, gamma=5)



SVM RBF (C=0.5, gamma=10)



Cross-validation Heatmap



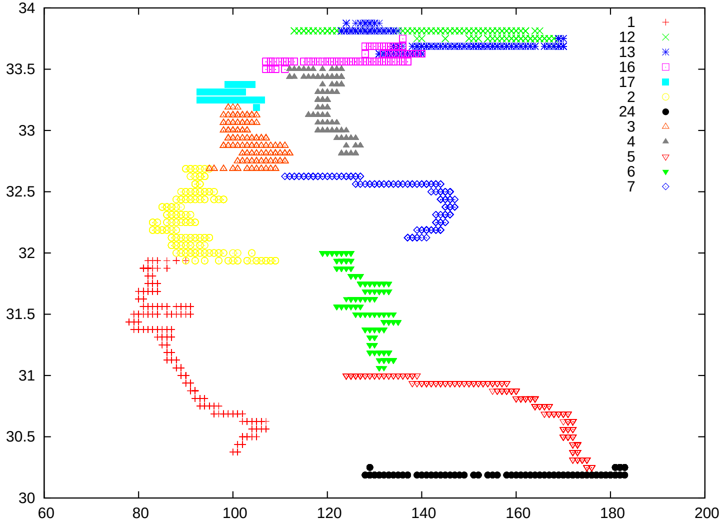
http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

Multi-class SVM

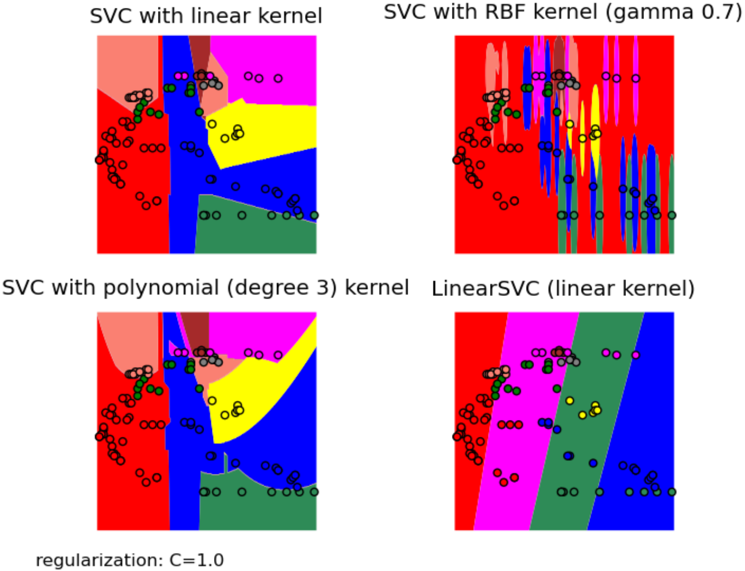
Two implementations in scikit-learn:

- SVC: one-against-one
  - $n(n-1)/2$ classifiers constructed
  - supports various kernels, incl. custom ones
- LinearSVC: one-vs-the-rest
  - n classifiers trained

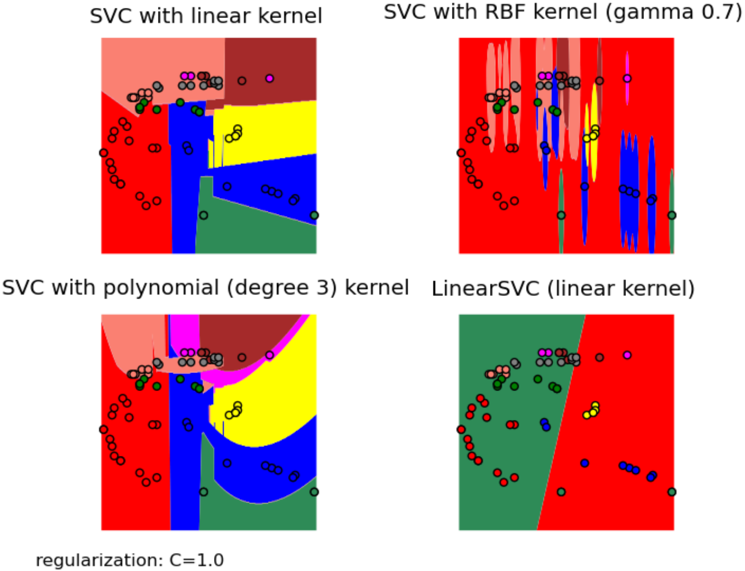
PAMAP-easy Training Data



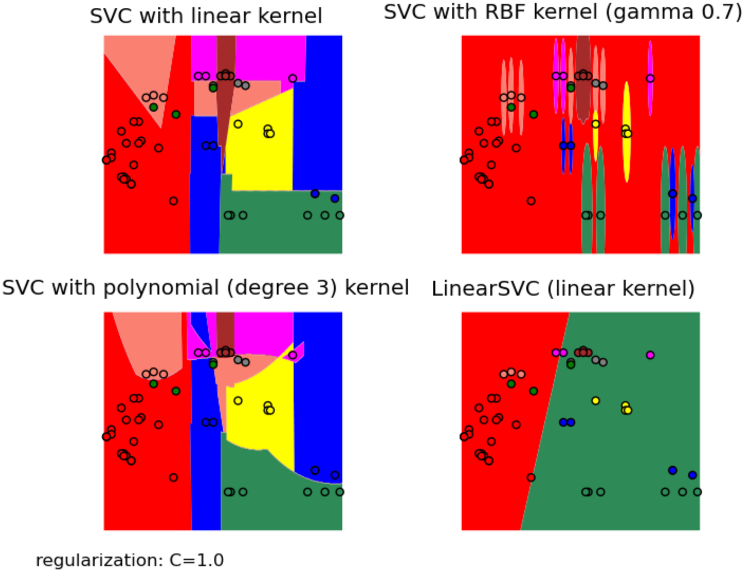
Default View (every 200)



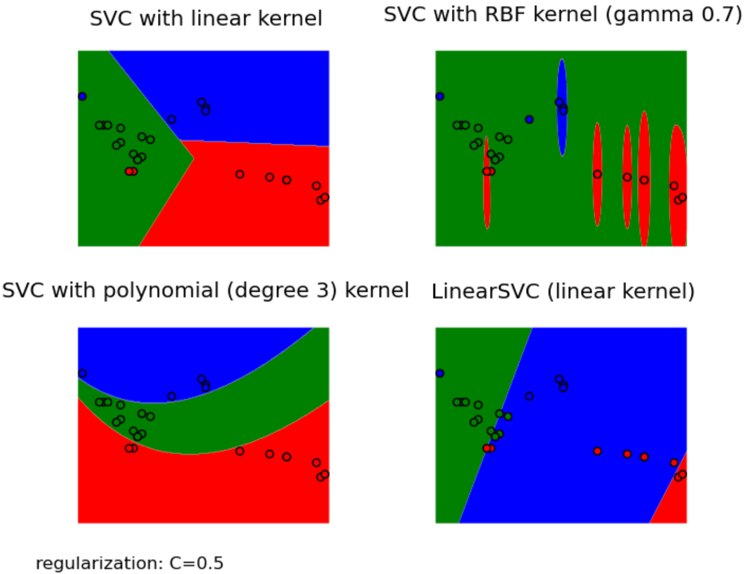
Default View (every 300)



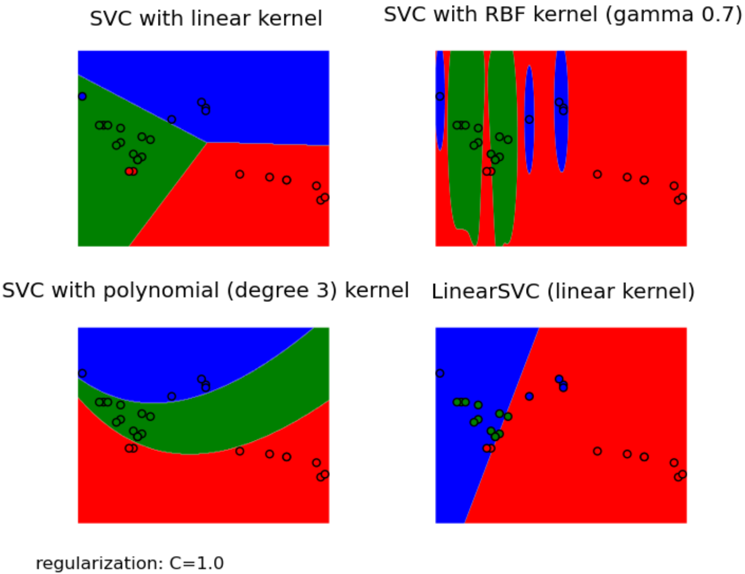
Default View (every 400)



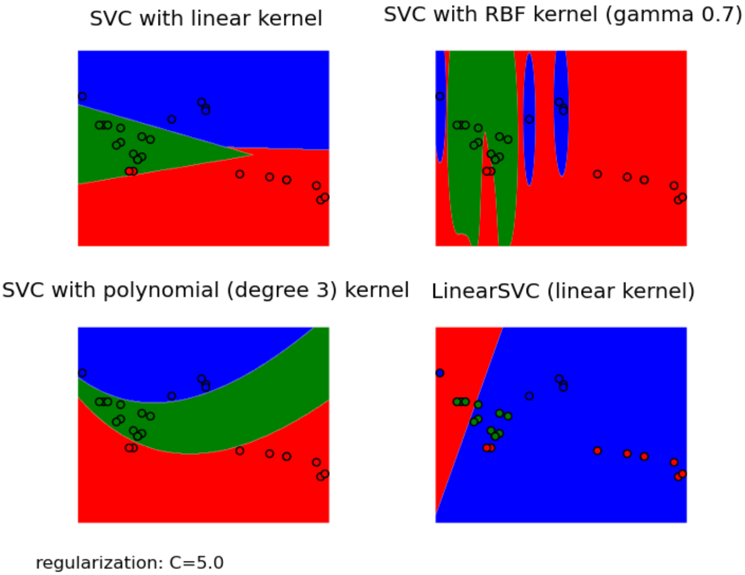
Regularization C=0.5



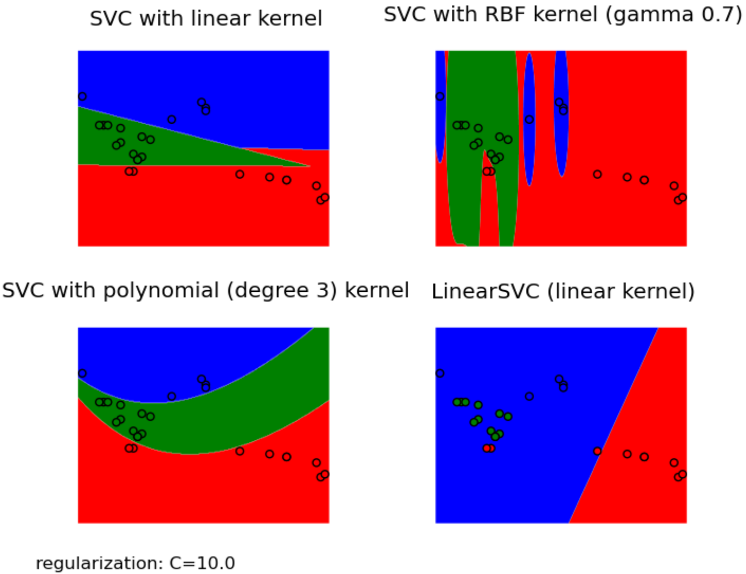
Regularization C=1



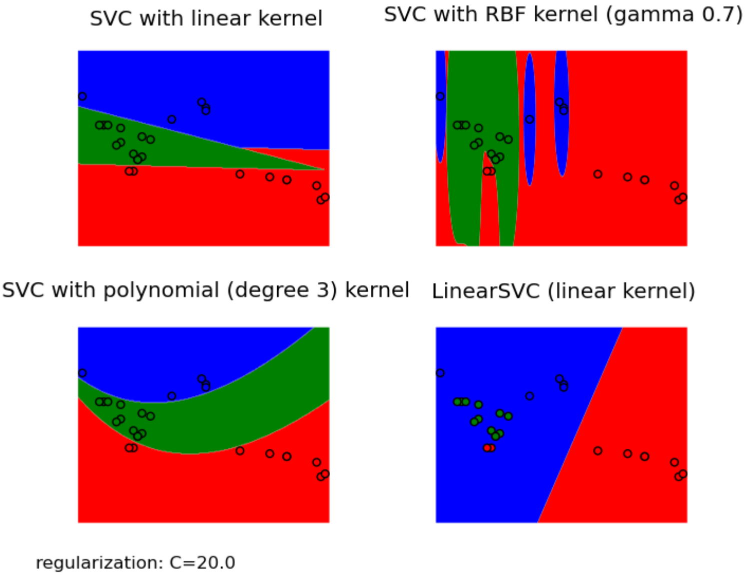
Regularization C=5



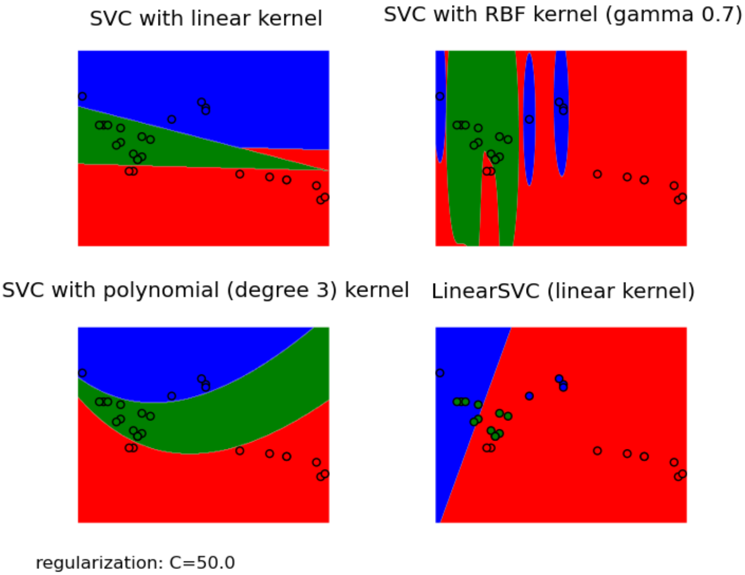
Regularization C=10



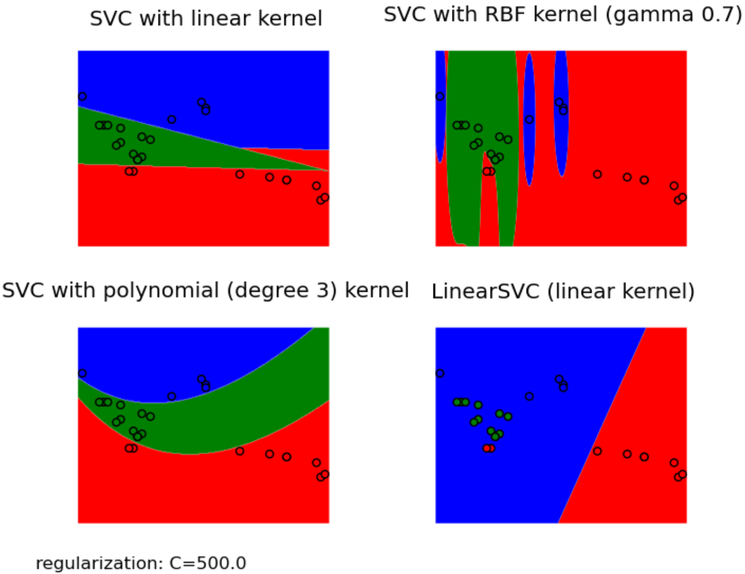
Regularization C=20



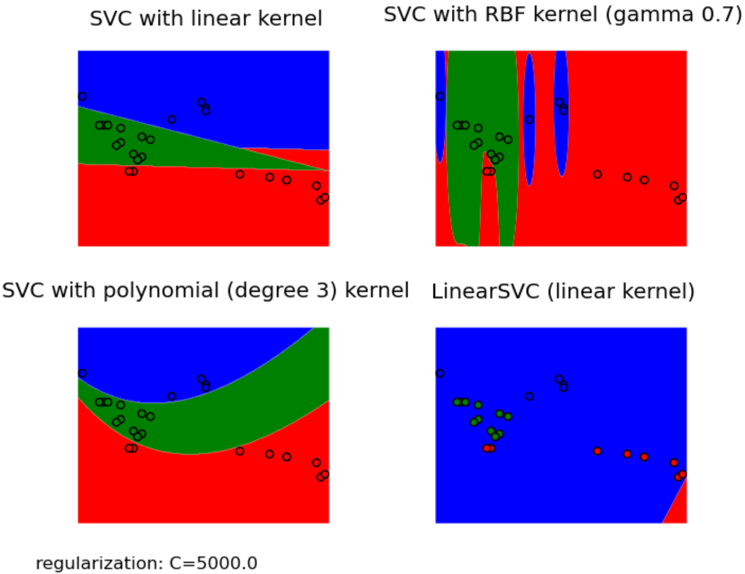
Regularization C=50



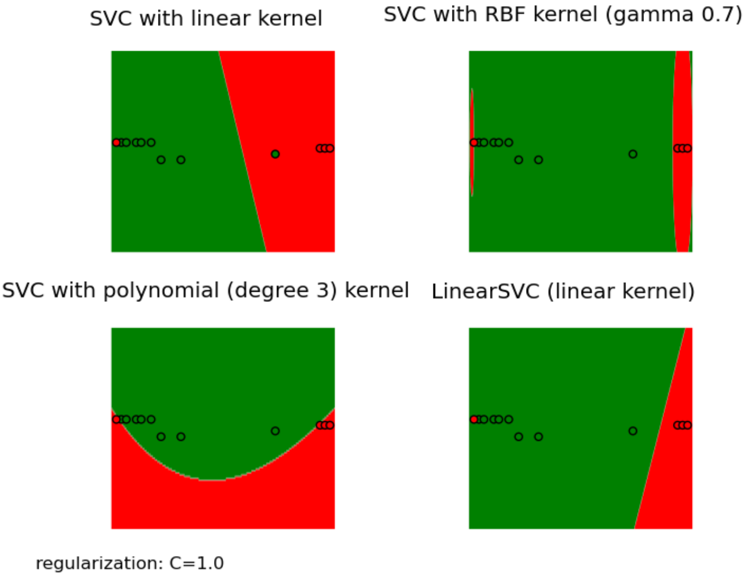
Regularization C=500



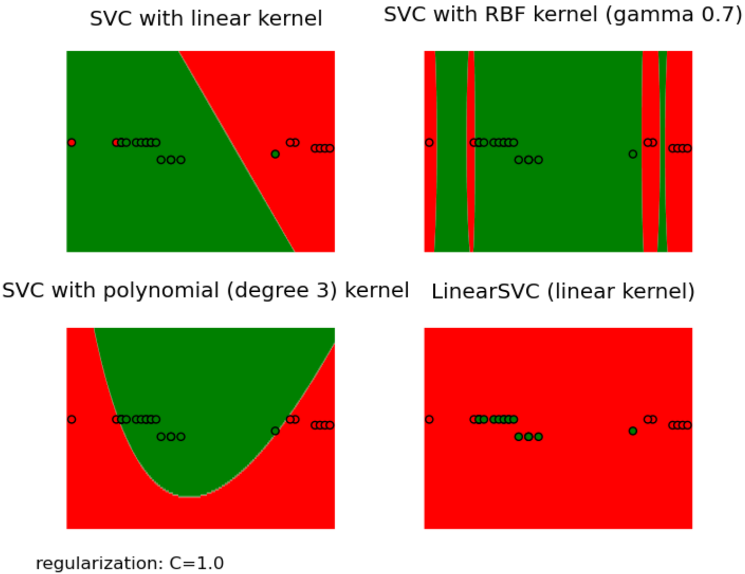
Regularization C=5000



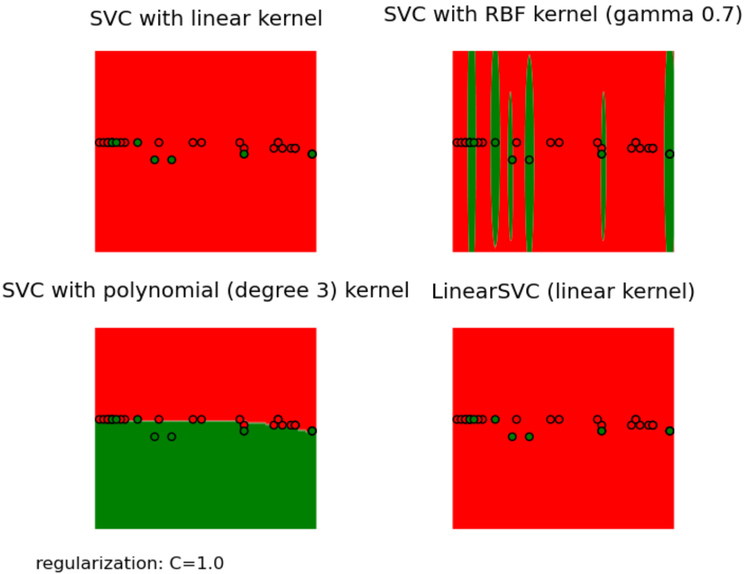
Inseparable classes 12,13 (every 200)



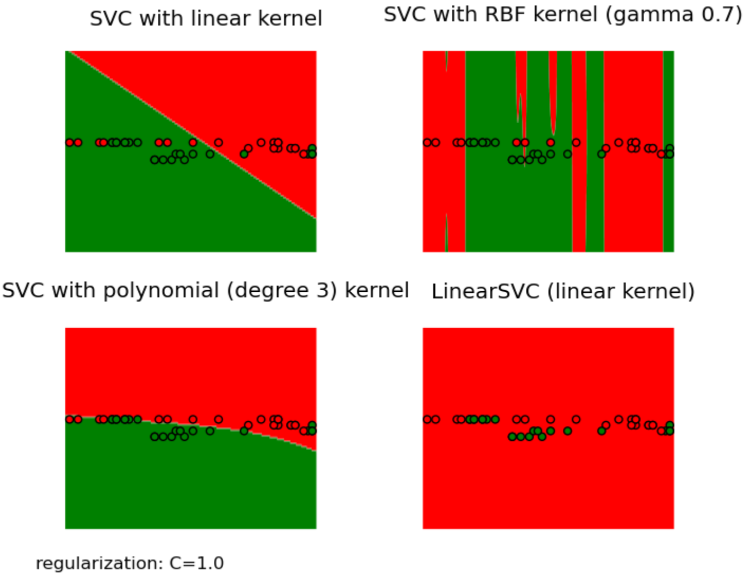
Inseparable classes 12,13 (every 100)



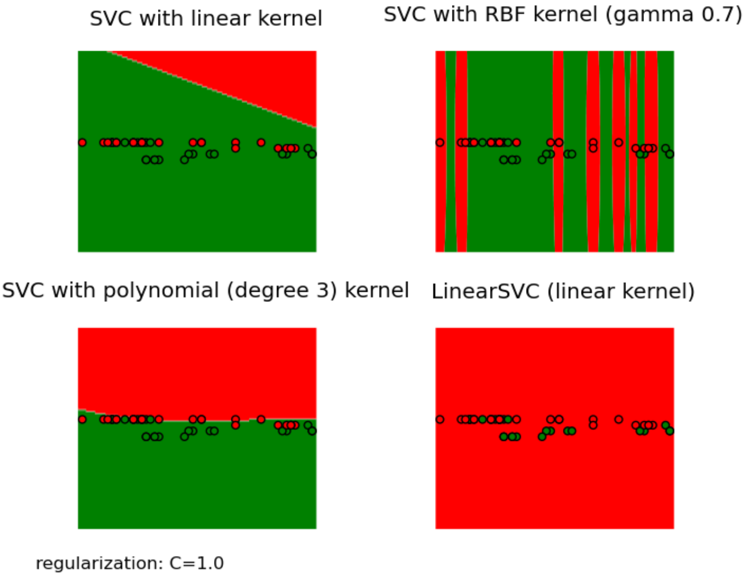
Inseparable classes 12,13 (every 80)



Inseparable classes 12,13 (every 60)



Inseparable classes 12,13 (every 55)



Task and Homework

- Required minimum: For PAMAP-Easy:
  - Cross-validate to choose between linear, poly and RBF.
  - Create the heatmap for RBF.
  - Use the GridSearchCV to find the best C and gamma.
- Optional: n-fold cross-validation for GridSearchCV:
  - Use just the training data, perhaps subsampled.
  - Run GridSearchCV n times, using each $1/n$ as the test set and the remaining $9/n$ as the training data.
  - Report min, max, average and standard deviation of achieved C, gamma and test error.

- Try to implement it as a generic command-line utility.
- Just call it from the Makefile.

Due: 2 weeks from now, i.e. April 14.