Optimization Application: Support Vector Machines - *updated*
Prof. K. Dana

Instructions: Complete the following project with your own code in cvx. **Important: Do not use spline fitting/svm software or matlab commands for splines or svm.** Submit to sakai one zipped file containing **three files**: 1) *description_lastname.pdf*: written description of your approach including equations and the statement of the optimization problem. Be concise. 2) *code_lastname.txt*: Code 3) *plots.pdf*:s Plots showing the results. Label axes of each plot.

- *Support Vector Machine Classifier* Classify the pixels of a color image using a support vector machine classifier. *Test your program on two different input images: an image of an apple and an image of a hand.* Make the input images relatively challenging. That is, use an image where simple color thresholding will not work. *In your written description, show the image segmentation with thresholding to illustrate that simple thresholding is not a good classifier.*
  - **Training Data** Use a small rectangular region within the center part of the hand region (identify the location using *ginput*), and a few small rectangles in the background as training data. *All of the pixels in these regions can be used as labelled training data.*
  - **Feature Vector** The feature vector should be x, y, r, g, b or x, y, h, s, v. (Converting from rgb to hsv is a built-in matlab function). *Here, x, y are the image coordinates of the pixel. You may alternatively use d, the distance from the center, if it improves your results.*
  - **Build the Classifier** Set up the classification optimization problem using support vector classification is described in Chapter 8. Solve with cvx.
  - **Classify the data**
  - Plot the results. Plot the training data for each class on a 2D projection or set of projections of the feature space. Within the same plot, show the learned decision boundary (again projected to 2D space). Use the classifier to classify all the pixels in the color image. Show the segmented image for each case using a black/white image indicating background/foreground. Compare this classification with the simple classification using image thresholding.
  - Optional: Do the results improve or worsen when x, y is removed from the feature vector?