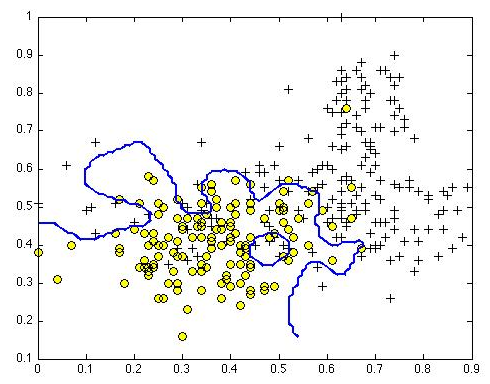
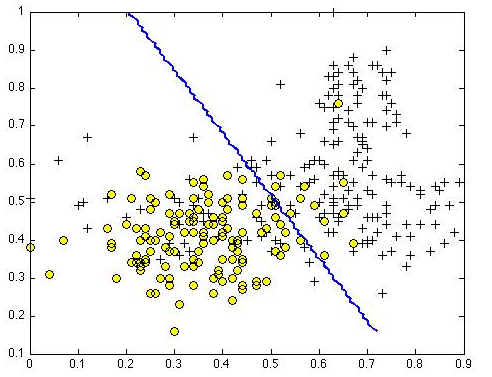
***Wk. 7 Quiz***

* Suppose you have trained SVM classifiers with a Gaussian kernel, and they learned the following decision boundary on the training set:

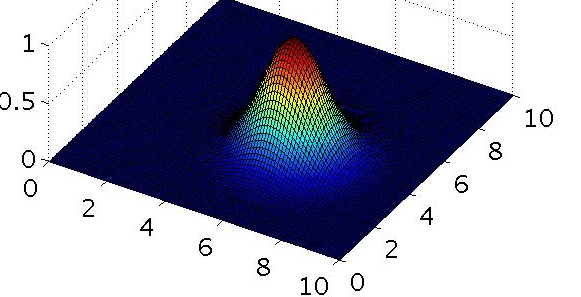
When you measure the SVM's performance on a CV set, it does poorly. Should you try increasing or decreasing C? Increasing or decreasing σ2?

* **LEFT= It would be reasonable to try decreasing C or to try increasing σ2**
* overfit 🡪 want higher bias to be more less to data points
* **RIGHT = It would be reasonable to try increasing C or to try decreasing σ2**
* underfit 🡪 want higher variance to be more sensitive to data points
* The formula for the Gaussian kernel is given by:

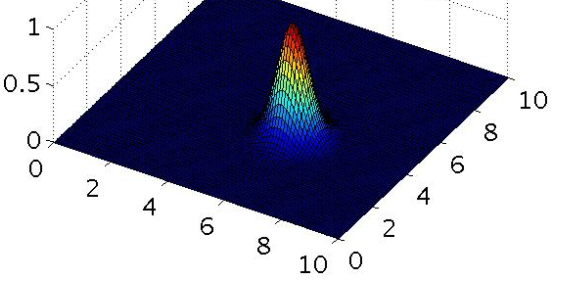


The figure below shows a plot of:





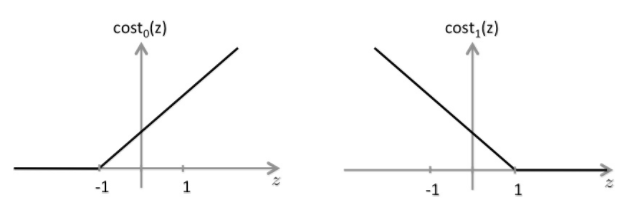
Which of the following is a plot of f1 when σ2 = 0.25?



* The SVM solves



where the functions cost0(z) and cost1(z) look like this:



The first term in the objective is:



This 1st term will be 0 if 2 conditions hold true. Which are the 2 conditions that guarantee this term equals 0?

* **For every example with y(i) = 1, we have that θ(t)x(i) ≥ 1 (2nd half = 0, 1st half = 0 only when cost1(z) = 0 🡪 look at graph)**
* **For every example with y(i) = 0, we have that θ(t)x(i) ≤− 1 (1st half = 0, 2nd half = 0 only when cost0(z) = 0 🡪 look at graph)**
* Suppose you have a dataset w/ n = 10 features + m = 5000 examples. After training a logistic regression classifier w/ gradient descent, it has underfit the training set + does not achieve desired performance on the training or CV sets. Which of the following might be promising steps to take?
* **Reduce the number of examples in the training set.**
* **Create / add new polynomial features**.
* Use an SVM with a linear kernel, without introducing new features.
* Use an SVM with a Gaussian kernel, without introducing new features 🡪 too slow to run
* Increase the regularization parameter λ (decrease C)
* Use a different optimization method since using gradient descent to train logistic regression might result in a local minimum.
* Try using a neural network with a large number of hidden units.
* Which of the following statements are true? Check all that apply.
* **It is important to perform feature normalization before using the Gaussian kernel.**
* **The maximum value of the Gaussian kernel (i.e., sim(x,l(1))) is 1.**
* ***If the data are linearly separable, an SVM using a linear kernel will return DIFFERENT parameters θ regardless of the chosen value of C (i.e., the resulting value of θ depends on C).***
* **Suppose you are using SVMs to do multi-class classification and would like to use the one-vs-all approach. If you have K different classes, you will train K different SVMs.**
* If the data are linearly separable, an SVM using a linear kernel will
* return the same parameters θ regardless of the chosen value of
* C (i.e., the resulting value of θ does not depend on C).
* Suppose you have 2D input examples (ie, x(i)∈R2). The decision boundary of the SVM (with the linear kernel) is a straight line.
* If you are training multi-class SVMs with the one-vs-all method, it is
* not possible to use a kernel.