Implications of using constant / variable alpha

Choosing a constant \alpha implies that the algorithm cannot adapt: if \alpha is too large, it will not converge, and if it is too small, convergence will be too slow.

Choosing a variable \alpha implies the algorithm can adapt over time, starting with a large learning rate to allow for fast movements toward a local minimum and then reducing it over time to allow the algorithm to converge with precision to a local minimum.

Fitting A SVM By Hand

The margin is the complete distance between the points, not their distance to the hyperplane

VC Dimension of a hyperdimentional sphere

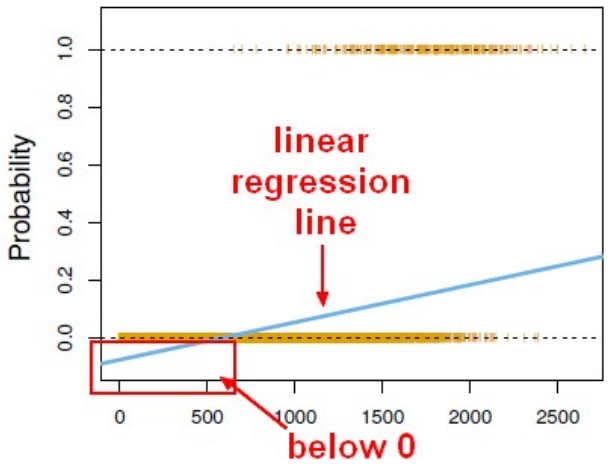
If the samples inside the sphere are always marked positive

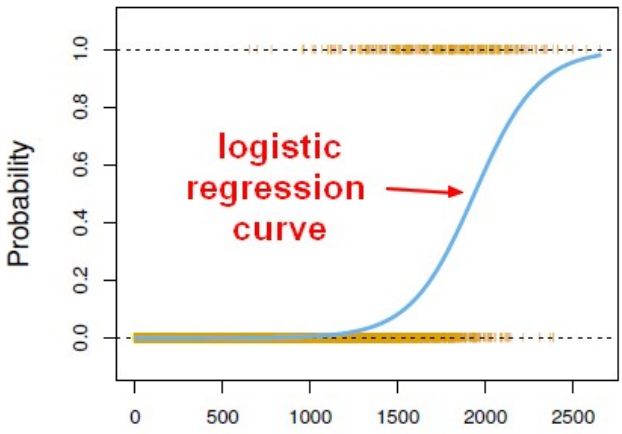
One point is shattered because we can always choose the radius of the sphere to include or not include the point depending on its label. Two points are not shattered because when the point closest to the origin is negative, we cannot correctly classify it without misclassifying the positive, farther point.

If we can select the direction of the surface arbitarily

Two points are shattered because now we can include one point inside the sphere, leave the other point out of it, and choose the direction appropriately.

difference between linear regression and logistic regression?





May get trapped in local minima?

Learning Methods	May get trapped in local minima?	Why?
Decision Tree	Yes	Decision trees apply some greedy decision based on information theory in order to make an initial decision, and then don't allow to back-track. It is not too hard to construct an example in which this is not the optimal choice.
Polynomial Regression	No	Because the least-squares objective function ("squared loss") is convex (concave).
Logistic Regression	No	Because the Objective Function / Cost Function – derived from Maximum Likelihood Estimation embedded with the Logisitic Sigmoid Function – is convex.
Perceptron	No	
Neural Network	Yes	
SVM	No	SVM finds the only one **maximum** margin.
K-means	Yes	Depends heavily on the initialization of centroids. See this.
Q-Learning	No	
kNN	No	It is not solved greedily but rather by finding all distances and choosing the minimum.
RBMs	Yes	

General Procedure of Training and Evaluating a Model

1. **Load raw** data file.
2. **Feature expansion** (Optional).
3. Split dataset into **training dataset and test dataset**.
4. Separately **standardize** two datasets.
5. Prepend a **column of 1's** to the X of the two datasets, if needed.
6. Train:

    1. For each trial:
        1. Shuffle the training dataset.
        2. Split X and y of the training dataset into 10 folds (assuming 10-fold cross-validation).
        3. For each fold "validation set" in these 10 folds:
            1. Train the model using the rest "training folds".
            2. Evaluate the "validation error" using the held-out validation set. Save it.
        4. Report the parameters found that yielded the lowest validation error.
    2. Average the best parameters from each trial.
7. Evaluate:
    1. Just feed the test dataset in.

How Biases and Variances Change with Increasing Parameter

Learning Methods and the Meta-Parameters	Bias	Variance
kNN: k↗	↑ (1NN has 0)	↓
Unpruned Decision Trees: depth d↗	↓ since it now fits training data better	↑ since samples with tiny differences would be distinguished now.
Logisitic Regression: n↗	-	↓ since it now better models the problem
Regularized Logistic Regression: λ↗	↑	↓
Regularized Logistic Regression: # of features d↗	↓	↑
GaussianSVM w/ small C: bandwidth σ↗	-	-
GaussianSVM w/ large C: bandwidth σ↗	-	↓
AdaBoost w/ decision stumps: boosting iterations T↗	↓ more weak hypotheses to model the problem better	↓ more weak hypotheses to model the problem better
Gaussian mixture model: number of mixture components k↗	↓	↑ because it's sensitive to the initial centroids.