CS 189/289: Introduction to Machine Learning - Discussion 7

1. Classifying Using a Kernelized SVM

   We have trained an SVM with a Gaussian RBF (radial basis function) kernel:

   $$k(x_i, x) = e^{\frac{-(x_i - x)^2}{2\sigma^2}}$$

   Now we have a set of support vectors $\{x_i\}$ (for all i where x_i is a support vector) and the associated set training labels $\{y_i\}$ and alpha values $\{\alpha_i\}$.

   How do we classify a new point x?

Solution:

$$\begin{aligned}
\hat{y} &= \text{sign}(\langle \theta, \phi(x) \rangle), \text{ and } \theta = \sum_{i=1}^{n} \alpha_i y_i \phi(x_i) \\
&= \text{sign}\Big(\langle \sum_{i=1}^{n} \alpha_i y_i \phi(x_i), \phi(x) \rangle\Big) \\
&= \text{sign}\Big(\sum_{i=1}^{n} \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle\Big) \\
&= \text{sign}\Big(\sum_{i=1}^{n} \alpha_i y_i k(x_i, x)\Big) \\
&= \text{sign}\Big(\sum_{i=1}^{n} \alpha_i y_i e^{\frac{-(x_i, x)^2}{2\sigma^2}}\Big)
\end{aligned}$$

2. Midterm Discussion Questions

Solution: We have listed some of the answers to these open questions, there are certainly more.

What's the difference between perceptron and SVM?

Solution: The perceptron algorithm seeks to find any hyperplane that will separate the training data. SVM will try to find the separating hyperplane that maximizes the distance (margin) from the hyperplane to the closest training point on either side.

What's the difference between SVM and logistic regression?

Solution: In the end, both methods find a linear desicion boundary between two classes. SVM is trying find a separating hyperplane that maximizes the distance (margin) from the hyperplane to the closest training point on either side. Logistic regression models the probability that a given data point (x) belongs to one of two classes (y_1), specifically $P(Y = y_1|X = x)$. Because logistic regression models this probability, we have more than just a classification of a point ($Y = y_1$ or $Y = y_0$); we also have a measure of the probability, or confidence, that the point belongs to a specific class.

What's the difference between generative models and discriminative models?

Solution: Both models seek to find the probability of the output Y given the data X, $P(Y|X)$. Descriminitive techniques model $P(Y|X)$ directly, focusing on a direct mapping from the input data to the output data. Generative techniques model the joint probability of X and Y, $P(X, Y)$, typically modelling $P(X|Y)$ and $P(Y)$ and using Bayes' rule to arrive at $P(Y|X) \propto P(X|Y)P(X)$. Both approaches have models and both use data. The discriminative approach tries to rely heavily on the data, while the generative approach tries to take advantage of prior knowledge or assumptions of how the data was created.

For example, in classification, a discriminitive approach could model $P(Y|X)$ directly as a logistic function, $P(Y = y_1|X = x) = \frac{1}{1+e^{\beta^T x+\beta_0}}$. In this case, the

discriminative model includes parameters β and β_0 that are chosen to best fit the logistic function to the data pairs X and Y. A generative approach to the same classification problem could model the class-conditional probability as Gaussian and the class probabilty as uniform. In this case the generative model includes parameters for the mean and variance of the two class-conditional probabilites.

Under what conditions are MLE and MAP estimation the same?

Solution: Maximum likelihood estimation (MLE) and maximum a posteriori (MAP) are the same when the prior probability on the parameters θ is a uniform distribution, $P(\theta) = C$.

MLE tries to find the parameters θ that maximize the probabilility of the data X, given those parameters, $P(X|\theta)$ (the likelihood term). MAP tries to find the parameters that maximize the posterior distribution, the probability of the parameters given the data, $P(\theta|X)$. When $P(\theta)$ is uniform, $P(\theta|X) \propto P(X|\theta)P(\theta)$ becomes $P(\theta|X) \propto P(X|\theta)$ and MLE and MAP are solving the same problem.

List some ways we could overfit with some of the techniques we have learned in class.

Solution:

- Using the testing data for training
- Using training data that is not representative of the test data
- Using a high degree polynomial in regression
- In SVM, choosing a C value that is too high, which has a huge penalty on any points on the wrong side of the margin boundary and forcing the margin to be smaller to accomadate potentially noisy outliers
- Using an RBF Kernel with very small σ