Machine Learning Decisions

written by Alvin Wan

1 Summary of Models

The following is a tabular summary of models and the information they offer.

Source: *http://cross-entropy.net*

Model	Classif or Regr	Gen or Disc	Par or Non-par
Gaussian Discriminant Analysis	Classification	Generative	Parameteric
Naive Bayes Classifier	Classification	Generative	Parametric
Linear Regression	Regression	Discriminative	Parametric
Logistic Regression	Classification	Discriminative	Parametric
Neural Network	Both	Discriminative	Parametric
K nearest neighbor classifier	Classification	Generative	Non-parametric
Decision Trees	Both	Discriminative	Non-parametric
Sparse Kernelized Lin/Log Regression	Both	Discriminative	Non-parametric
Support Vector Machine (SVM)	Both	Discriminative	Non-parametric

2 Survey of Classes, Models, Algorithms

We will begin with a general survey of classes of models, models, and algorithms. This overview overlaps with information below, but we hope that providing a holistic view of these options gives more insight.

2.1 Classes

As in Note 9 : Gaussian Discriminant Analyses, we will consider the following classes of models and when each is most applicable:

1. **Empirical Risk Minimization**

   These methods do not yield a probabilistic model and instead only compute a decision boundary.

2. **Generative Models**

   We first model the class-conditional probability $\Pr(X|Y)$ and using Bayes', then model the posterior $\Pr(Y|X)$; these models take fewer samples to reach the same accuracy as discriminative models but make assumptions about the data's distribution. The models are usually more interpretable.

3. **Discriminative Models**

   We directly model the posterior $\Pr(Y|X)$; these models take more samples to train but do not make assumptions about the class-conditional probability densities.

2.2 Models

This is a brief survey of models that we can pick from. We identify key characteristics of each model that may influence your decision; we leave the mathematical rigor of proving these solutions or gradients to other notes.

1. **Linear Regression** (derivation in Note 10)

   - Discriminative Model
   - Boundary is linear

2. **Logistic Regression** (derivation in Note 11)

   - Discriminative model
   - Boundary is linear; to compute, set class-conditional probability to $\frac{1}{n_C}$, where n_C is the number of classes
   - For binary classification, take MLE of Bernoulli class-conditional density.

3. **Least Squares**

   - Has a closed-form solution
   - Equivalent to MLE of Gaussian class-conditional probability density.

4. **Ridge Regression**
   - Has a closed-form solution
   - Equivalent to MLE of Gaussian class-conditional densities and priors.
5. **Lasso**
   - Does not have a closed-form solution
   - Equivalent to MLE of Gaussian class-conditional density and Laplace prior.
   - Induces model sparsity
6. **Perceptron**
   - Converges only if data is linearly-separable
7. **Quadratic Discriminant Analysis**
   - Generative Model
   - Creates a quadric surface for a decision boundary
8. **Linear Discriminant Analysis**
   - Generative Model
   - Creates a hyperplane for a decision boundary

2.3 Algorithms

1. **Gradient Descent**
   - Each iteration is relatively slow to compute.
   - For least-squares and least-squares variants the gradient for matrices is very similar to the gradient for single samples.
2. **Stochastic Gradient Descent**
   - Each iteration is relatively fast to compute.
   - Converges slower than gradient descent.

3. **Newton's Method** (i.e., Newton-Raphson)
   - Converges in one iteration for a quadratic loss function.
   - Generally converges faster than gradient descent, where well-defined.
   - Compute $w_{k+1} = w_k - \frac{f(w_k)}{f'(w_k)}$ to find the roots.
   - Compute $w_{k+1} = w_k - \frac{f'(w_k)}{f''(w_k)}$ to find the extrema.

3 Picking based on Data

In the following sections, we explore various choices of algorithms and models, based on our data.

3.1 Picking Algorithms: Linearly Separable or Not

Note that given a set of data, there always exists a linear boundary in some higher dimensional space. Thus, we consider a dataset to be linearly separable *given* a set of features. The following algorithms work only for linearly-separable data. By this, we mean that if the data is not linearly-separable, these algorithms will not converge or terminate:

1. Perceptron
2. Hard-margin Support Vector Machines

The following algorithms *will* converge but will have poor results, because the boundary is linear:

1. Linear Discriminant Analysis (proof of linear boundary in Note 9)

   LDA computes a single quantity, instead of iterating until convergence. Thus, it will yield a value but not necessarily an accurate one.

The following algorithms will yield reasonable values, as they compute a non-linear combination of feature vectors.

1. Quadratic Discriminant Analysis (proof of quadratic boundary in Note 9)

3.2 Picking Models: Features v. Samples

We consider an $n \times d$ matrix of samples X. We can make several decisions based on the dimensions of X, specifically whether $n > d$ or when $d > n$. This is because $X^T X$ is $d \times d$ and XX^T is $n \times n$. Let us consider different models for gradient descent and stochastic gradient descent.

Note that if $n < d$, we can't take the inverse of $X^T X$, since $rank(X) = rank(X^T X) \leq n < d$, meaning that $X^T X$ cannot be full rank.

If $n < d$, we generally use the following tricks:

- Plug in $w = w_n + X^T \alpha$ into the objective function, and compute the optimal α. In this case, we compute ($O(n^2 d)$) an $n \times n$ matrix, XX^T and invert it in $O(n^3)$. Computing w^* involves computing $X^T X$ ($O(nd^2)$) and inverting the $d \times d$ in $O(d^3)$.
- Apply the matrix inversion trick, so that our gradient - for example, for ridge regression - is $X(X^T X + \lambda I)^{-1} y$

4 Bias-Variance Tradeoffs

We note first that for any regression problem, mean squared error will decompose into the following terms; see Note 12 and Note 13 for derivations of this:

$$\textsc{Bias} + \textsc{Variance} + \textsc{Irreducible Error}$$

1. Add regularization term: increases bias, decreases variance
2. Add new feature, or use more expressive kernel: decreases bias, increases variance
3. Add more data: variance decreases