Machine Learning

Ravi Donepudi

November 11, 2023

These are notes I made studying for interviews for data science roles. Use them at your own risk.

1 Probability

1.1 Basic rules

1. For events x, y, we have $P(x \text{ or } y) = P(x) + P(y) - P(x \text{ and } y)$. If x and y are disjoint, this simplifies to $P(x \text{ or } y) = P(x) + P(y)$.
2. The conditional Probability[1] of x given y: $P(x|y) := \frac{P(x \text{ and } y)}{P(y)}$.
3. For events x, y, we have $P(x \text{ and } y) = P(x|y)P(y) = P(y|x)P(x)$. If x and y are independant events then $P(x \text{ and } y) = P(x)P(y)$.
4. For conditional probabilities, we have the rule that $P(x|y, z) = P(x|y)P($

2 Algorithms

2.1 Conventions

- Elements of $\mathbb{R}^d$ are assumed, by default, to be column vectors, i.e. $(x_1, \ldots, x_d)^T$.
- X typically denotes a vector $(x_1, \ldots, x_n)^T$ of input variables and x
- d denotes the number of features (typically not including the bias).
- n denotes the number of training examples and the i-th training example is typically indicated as $x^(i)$.
-

2.2 Linear Regression

1. Given a real world function y depending on variables $x = (x_1, \ldots, x_d)^T$ and given n training examples $(x^(j), y^(j))$ for $0 \leq i \leq n$ want to approximate y with a linear function $h(x) = w^T x$ where $w = (w_1, \ldots w_d)$ and b are fixed real numbers.
2. Why linear function? Because simplest kind of function ever and straight lines are the simplest kind of trend.
3. Most commonly used cost function to measure goodness-of-fit is the least squared loss function, i.e.
$$J(w) = \sum_{i=1}^{n} (y_i - w^T x - b)^2$$
. We want to find a w that minimizes this. How? Take the gradient, set $= 0$ and solve.
4. Set $Y = (y^{(1)} \ldots y^{(n)})$ and X be the matrix whose j-th row is the j-th training example $(x^{(1)}, \ldots x^{(n)})$ Get closed form for w and b values minimizing $J(w)$ as
$$(w, b)^T = (X^T X)^{-1} X^T y$$

[1] Assuming $P(y) \neq 0$

5. Also can use gradient descent given by (vectorized) update below, where α is the learning rate.

$$w := w + \alpha \left(\sum_{j=1}^{n} (y^{(j)} - h_w(x^{(j)})x^{(j)} \right)$$

.

6. Why least squares for cost function? This is what we get when we try to do maximum likelihood estimation of (w, b) assuming i.i.d normal residuals in

$$y_i = w^T x_i + b + \epsilon_i$$

with an independance assumption on the sampling. Build the likelihood function

$$\mathcal{L}(w, b, (x^{(j)}, y^{(j)})) = \prod P(y^{(i)}|x^{(i)}; w, b)$$

and take it's (w, b)-gradient and we basically get $J(w)$ modulo a constant term.

2.3 Logistic regression

1. Logit function

$$f(x) = \frac{1}{1 + e^{-t}}, \quad f(0) = \frac{1}{2}, \quad \lim_{t\to\infty} f(t) = 1, \quad \lim_{t\to-\infty} f(t) = 0.$$

Use the various features w_i

2. The assumptions of one variant of a Gaussian Naive Bayes classifier imply the parametric form of P(Y|X) used in Logistic Regression.

3. GNB and Logistic Regression converge toward their asymptotic accuracies at different rates. As Ng & Jordan (2002) show, GNB parameter estimates converge toward their asymptotic values in order $\log n$ examples, where n is the dimension of X. In contrast, Logistic Regression parameter estimates converge more slowly, requiring order n examples. The authors also show that in several data sets Logistic Regression outperforms GNB when many training examples are available, but GNB outperforms Logistic Regression when training data is scarce.

3 Miscellaneous

1. MLE vs MAP:

$$\text{argmax}_\theta P(D|\theta) \quad \text{versus} \quad \text{argmax}_\theta\, P(\theta|D) = \text{argmax}_\theta\, P(D|\theta)P(\theta),$$

where we assume a prior distribution shape in the latter.

2. Non-parametric vs parametric: Parametric just means that we are going to use the training data to train some *parameters* and use just those for prediction. No more reason to keep the training data. Eg: Linear, Logistic regressions and Naive Bayes. Non-parametric just means that we have to keep the entire training set around to predict the next test example. Eg: k-nearest neighbors

4 Useful links

1. Neural Network Notes (21 pages): Definitions of neural networks and derivation of backpropagation algorithms.

2. Naive Bayes vs. Logistic Regression

3. Andrew Ng's Machine Learning Notes

4. Numpy and Matplotlib primer

5. MLE vs MAP estimators