1 Perceptron algorithm

1.1 Algorithm

Initialize $\theta^{(0)}$ to some value, and initialize the index k to 0. Iterate: Select the next data point $(\mathbf{x}_t, y_t)$ and check whether $\theta^{(k)}$ classifies it correctly. If it is incorrect (i.e.$y_t(\theta^{(k)})^T\mathbf{x}_t < 0$), set $\theta^{(k+1)} = \theta^{(k)} + y_t\mathbf{x}_t$ and increment $k \leftarrow k+1$.

1.2 Analysis of convergence and correctness

Assumption 1: There exists $R \in (0,\infty)$ such that every input $x_t \in \mathbb{D}$ satisfies $\|\mathbf{x}_t\| \le R$. Equivalently, the input vectors are bounded.
Assumption 2: There exists a parameter θ^* and positive constant $\gamma > 0$ such that $\min_{t=1,;n} y_t(\theta^*)^T\mathbf{x}_t \ge \gamma$

Theorem 1. Under the initial vector $\theta^0 = \mathbf{0}$, for any data set $\mathbb{D}$ satisfying the above assumptions, the perceptron algorithm produces a vector $\theta^{(k)}$ classifying every example correctly after at most $k_{\max} = \frac{R^2\|\theta^*\|^2}{\gamma^2}$ mistakes (and hence updates steps), where θ^*, γ, R are defined in the two assumptions.

1.3 Margin and geometry

Definition: Let $\gamma = \min_{t=1,\dots,n} y_t\theta^T\mathbf{x}_t$, we could define $\gamma_{geom} = \frac{\gamma}{\|\theta\|}$, which is the smallest distance from any data point $\mathbf{x_t}$ to the decision boundary.
Remark: We could see $\frac{1}{\gamma_{geom}}$ as a measure of difficulty. Also, we could rewrite $k_{\max} = \frac{R^2}{\gamma_{geom}^2}$, which is not directly dependent on dimension or number of samples.

2 Support vector machine

2.1 Formulation

We propose to find a linear classifier to maximize the margin, as larger margin leads to a more robust classifier. $\max_{\theta,\gamma} \frac{\gamma}{\|\theta\|}$ subject to $y_t\theta^T\mathbf{x}_t \ge \gamma\ \forall t$
By treating $\frac{\gamma}{\theta}$ as one entity, taking norm square and rewrite maximization into minimization, we could formulate as $\min_\theta \frac{1}{2}\|\theta\|^2$ subject to $y_t\theta^T\mathbf{x}_t \ge 1\ \forall t$
The optimal solution is unique.

2.2 Linear classifier with offset

We now add an offset into our linear classifier: $f(x) = \text{sign}(\theta^T\mathbf{x} + \theta_0)$, and consequently the optimization is reformulated as $\min_{\theta,\theta_0} \frac{1}{2}\|\theta\|^2$ subject to $y_t(\theta^T\mathbf{x}_t+\theta_0) \ge 1\ \forall t$ Offsets allow decision boundary not to pass through origin, thus giving greater flexibility. Note that the offset do not appear in the objective.

2.3 Soft margin

Since most datasets are not linearly separable in real life, we consider allowing misclassification, by penalize such mistakes in terms of additional cost in objective. The optimization problem is reformulated as $\min_{\theta,\theta_0,\varsigma} \frac{1}{2}\|\theta\|^2 + C\sum_{t=1}^n \varsigma_t$ subject to $y_t(\theta^T x_t + \theta_0) \ge 1 - \varsigma_t, \varsigma_t \ge 0\ \forall\ t$ In this case, $\varsigma = (\varsigma_1, \dots, \varsigma_t)$ is a set of slack variables. If $\varsigma_t = 0$, then we recover the original SVM problem. If $\varsigma_t \in (0,1)$, then it means that we still require correct classification, but we allow some points to lie within margin. If $\varsigma_t > 1$, then we allow misclassification. C is the magnitude of penalty, the larger we set C, the greater the price of violation, and thus the slack variables will be small for optimality to avoid large penalty.

2.4 Support vector

The support vectors are data points: lie exactly on the margin, lie within the margin but still classified correctly, misclassified. These are the vectors that determine the optimal model, i.e.if we apply SVM on a reduced dataset with only those support vectors, we get back the same classifier.

2.5 Hinge loss

We define hinge loss as $[z]_+ = \max(0, z)$. We could reformulate soft margin problem as

$$\min_{\theta,\theta_0,\varsigma} \frac{1}{2}\|\theta\|^2 + C\sum_{t=1}^n [1 - y_t(\theta^T x_t + \theta_0)]_+$$

Since if we consider the slack constraint $y_t(\theta^T x_t + \theta_0) \ge 1 - \varsigma_t \implies \varsigma_t \ge 1 - y_t(\theta^T x_t + \theta_0) \implies \varsigma_t \ge \max(0, 1 - y_t(\theta^T x_t + \theta_0)) \implies \varsigma_t \ge [1 - y_t(\theta^T x_t + \theta_0)]_+$. When optimality is reached, the slack variables will converge to hinge loss.

3 Logistic regression

3.1 Formulation

We consider 'soft prediction', which is to output probability(confidence) of predicted label rather than making an assertion. We consider the logistic likelihood model: $P(y=1|x) = \frac{1}{1+e^{-z}}$, $z = \theta^T x + \theta_0$.

3.2 Property of sigmoid function

Let g denote sigmoid function, we have $P(y = -1|x) = 1 - P(y=1|x) \implies \log\frac{P(y=1|x)}{P(y=-1|x)} = \theta^T x + \theta_0 \iff$ positive result leads to probability of positive label greater than $\frac{1}{2}$ and vice versa. We have $P(y|x) = g(y(\theta^T x + \theta_0))$.

3.3 Regularization and logistic loss

Note that by scaling up θ, θ_0, the decision bound remains but the calculated probability rapidly grows to 100%/0%, therefore we can make arbitrarily confident predictions, prone to errors.
Note that logistic regression reports the predicted likelihood of label given data point, thus assuming independence, the total likelihood of reporting a dataset is the product of all such likelihoods, namely $L(\theta,\theta_0) = \prod_{t=1}^n P(y_t|x_t)$, we aim to maximize such likelihood, which is equivalent to max log-likelihood, which simplifies to $(\theta,\theta_0) = \arg\min_{\theta,\theta_0} \sum_{t=1}^n \log(1 + \exp(-y_t(\theta^T x_t + \theta_0)))$, which is called logistic loss. Note that if the dataset is linearly separable, once we find such a classifier, we could arbitrary scale θ, θ_0 to make such loss vanish. Therefore, we add regularization term $\frac{\lambda}{2}\|\theta\|^2$ to control the norm of parameter vector.

4 Linear regression

4.1 Formulation

We propose to find a linear regression model $y(x) = \theta^T x + \theta_0$ We optimize the model by minimizing mean square loss(visually, square of difference between true values and predicted values), therefore we are dealing with optimization problem $\arg\min_{\theta,\theta_0} \sum_{t=1}^n (y_t - \theta^T x_t - \theta_0)^2$

4.2 Bayesian perspective

4.2.1 Model and noise distribution

We consider from the perspective that the dataset is generated from a fixed unknown linear relation(assume the true data distribution is somehow linear) and perturbed by random noise $y_t = (\theta^*)^T x_t + \theta_0^* + z_t$ where θ^*, θ_0^* are fixed and unknown. We consider that the random noise is Gaussian noise. $z_t \sim N(0, \sigma^2)$. We see that for a given classifier with parameter θ, θ_0, we have $P(y|x,\theta,\theta_0) = N(y; \theta^T x + \theta_0, \sigma^2)$

4.2.2 Maximum likelihood estimation

By assumption of independence between samples, the likelihood function(intuitively, probability that the dataset is described by the classifier with assumption of Gaussian noise) is the product of likelihood for each data point:

$$L(\theta,\theta_0,\sigma^2; D) = \prod_{t=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y_t - \theta^T x_t - \theta_0)^2}{2\sigma^2})$$

We aim to find a model that is most likely to describe the dataset, which is to maximize L. Note that to maximize L is equivalent to

$$\max_{\theta,\theta_0,\sigma^2} \log L = K - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{t=1}^n (y_t - \theta^T x_t - \theta_0)^2$$

We could observe that to optimize (θ,θ_0) is not dependent on σ^2, hence we look at optimization problem $\arg\max_{\theta,\theta_0} -\frac{1}{2\sigma^2}\sum_{t=1}^n (y_t - \theta^T x_t - \theta_0)^2$
which is equivalent to

$$\arg\min_{\theta,\theta_0} \sum_{t=1}^n (y_t - \theta^T x_t - \theta_0)^2$$

We recover least square estimator! In other words, to find least square estimator is equivalent to find maximum likelihood estimator assuming Gaussian noise.

4.3 Analytic solution of least square estimator

Switch to matrix notation:

$$\sum_{t=1}^n (y_t - \theta^T x_t - \theta_0)^2 = \|y - X\vartheta\|^2$$

where

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} X = \begin{bmatrix} x_1^T & 1 \\ \vdots & \vdots \\ x_n^T & 1 \end{bmatrix} \vartheta = \begin{bmatrix} \theta \\ \theta_0 \end{bmatrix}$$

Differentiate the loss with respect to ϑ, we have the derivative $-2X^T(y - X\vartheta)$, set derivative to zero and solve for ϑ, and we get

$$\vartheta_{opt} = (X^T X)^{-1} X^T y$$

The matrix $(X^T X)^{-1}X^T$ is known as *pseudo-inverse* of X.

4.4 Bias and variance

The true data y is generated by $y = X\vartheta^* + z, z \sim N(\mathbf{0}, \sigma^2\mathbf{I})$, therefore from result of ϑ_{opt} we have

$$\vartheta_{opt} = \vartheta^* + (X^T X)^{-1} X^T z$$

Intuitively, suppose noise is not significantly high, then ϑ_{opt} is close to true parameter ϑ^*. Since $\mathbb{E}[\mathbf{z}] = \mathbf{0}$, we have $\mathbb{E}[\vartheta_{opt}] = \vartheta^*$, which means we are correct on average.
However, note that $\text{cov}[\vartheta^*, \vartheta_{opt}] = \sigma^2(X^T X)^{-1}$, hence if $(X^T X)^{-1}$ has large entries, the corresponding entries of ϑ_{opt} will have high variance.

4.4.1 Trade-off

The minimum square error could be broken down as

$$\mathbb{E}[\|\vartheta_{opt} - \vartheta^*\|^2] = \|\mathbb{E}[\vartheta_{opt}] - \vartheta^*\|^2 + \mathbb{E}[\|\vartheta_{opt} - \mathbb{E}[\vartheta_{opt}]\|^2]$$

It is the bias-variance decomposition of total loss. Usually, increasing model complexity makes model more flexible and sensitive to data, which reduces bias but increases variance.

4.5 Regularization and ridge regression

We penalize large entries of θ by adding regularization term:

$$\arg\min_{\theta,\theta_0} \sum_{t=1}^n (y_t - \theta^T x_t - \theta_0)^2 + \lambda\sum_{j=1}^d \theta_j^2$$

In matrix form, this gives

$$\vartheta_{opt} = \arg\min_{\vartheta} \|y - X\vartheta\|^2 + \lambda\|\vartheta\|^2$$

Now closed form solution is

$$\vartheta_{opt} = (X^T X + \lambda I)^{-1} X^T y$$

Similarly, we could consider expectation of estimator, in this case

$$\mathbb{E}[\vartheta_{opt}] = (I - \lambda(X^T X + \lambda I)^{-1})\vartheta^*$$

On average, the estimator now shrinks the ground truth, which is expected given that we penalize large parameters.

5 Useful references

Gaussian distribution: $N(z;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(z-\mu)^2}{2\sigma^2})$
Hessian matrix of f: matrix of $\frac{\delta f}{x_i x_j}$
Positive semidefinite matrix: Symmetric matrix M such that $z^T M z \ge 0 \forall\ z \iff$ all eigenvalues ≥ 0
Cauchy-Schwarz inequality: $< u, v > \le \|u\|\|v\|$