1.

(a) T

(b) F. It does have assumption of no formal distributions.

(c) T

(d) F. Distance don't converge as $n \to \infty$

(e) F. Deep neural network is a non-convex optimization problem

(f) F. A well-defined kernel is a real symmetric matrix

(g) T

(h) F. PCA is not always better than random projection especially when data are uncorrelated.

(i) F. Models that are more easily interpretable doesn't mean models are more accurate

(j) T

(k) F. Each splitting feature is chosen completely uniformly at random.

(l) F. cluster center $C(x_i)$ is updated as: $m_k = \frac{1}{N_k} \sum_{C_i=k} x_i$, $N_k = \sum \mathbb{I}(C(x_i)=k)$. no guarantee of still be original points.

(m) T

(n) F. It is possible that it has 100% testing error if a classifier obtains 0% training error.

(o) T

2. A: k-means, single linkage clustering. Gaussian mixture

B: Gaussian mixture

C: Gaussian mixture, single linkage clustering

D: single linkage clustering

3. (a) Pre-pruning: generating a tree with fewer branches than would otherwise be the case, i.e set maximum depth

Post-pruning: generating a tree in full and then removing parts of it.

(b) positive effects: low bias since we make use of all data points

negative effects: high variation in the testing model

We could apply k-n when size of data set is small.

(c) i. no

ii. no

iii. Yes

iv. no

(d) • Decision tree supports automatic feature interaction. whereas KNN can't.

• Decision tree is faster due to KNN's expensive real time execution

(e) training error: 10%

(f) AdaBoost is sensitive to noisy data and outliers. Meaning that whenever the data are not easily amenable to a specific seperation plane (with acceptable performances based upon the model objective. For example. the weak learners have to be better than random guessing), it's harder that the meta-learner converges to a strong learner without overfitting.

(g) i. Through applying bagging algorithm to generate classifier $\hat{h}$ by $h_1 \ldots h_m$ classifiers on original dataset. And then getting the best classifier by voting procedure or taking the average.

ii. the vainanel V of classifier $\hat{h}$ tends to zero when $n \to \infty$.

iii. The bias of $\hat{h}$ and h are equal.

4. (a) SGD converges faster than GD since it only takes few instances at a time, but error function of GD gets more accurate minimization.

When size of data set is large, it's reasonable to use SGD for faster convergence. While precise result is needed, GD is preferable.

(b) It's not a good idea.

Because validation loss may decline with fluctuation. Since instances are chosen randomly, there is no guarantee of strictly declining of loss. We need to decide the maximum increase of loss.

(c) i. $L(\vec{x}_{1:N}, y_{1:N}) := \sum_{i=1}^{N} (y_i - \vec{w}^T \vec{x}_i)^2 + \frac{\lambda}{2} \vec{w}^T w$

$$= (Y - \vec{w}^T X)^T (Y - \vec{w}^T X) + \frac{\lambda}{2} \|\vec{w}\|_2^2$$

Setting $\nabla L(\vec{x}_{1:N}, y_{1:N})(\vec{w}) = 0$, we obtain

$$-2X^T(Y - \vec{w}^T X) + \lambda \vec{w} = 0$$

$$\vec{w} = (X^T X + \frac{\lambda}{2} I_N)^{-1} X^T Y$$

ii. formula derived in (i) is least square method who has result only when $X^T X$ is invertable. Second reason is that calculation of inverse matrix is costly.

iii. choose λ using cross validation. Since the ideal value of λ is data-dependent.

5

(a) Suppose A is a randomized algorithm such that $A(x) = f(x) + \eta$ and A is differentially private, i.e $P[A(x) \in S] \leq e^{\varepsilon} P[A(x') \in S]$ where x and x' are neighboring dataset.

For Laplace mechanism, the global sensitivity of A is:

$$\Delta' A := \max_{x, x' \text{ s.t. } x, x' \text{ diff by one row}} \|A(x) - A(x')\|_1$$

Also

$$\eta \sim Lap(\eta \mid b) = Lap(\eta \mid \Delta' A / \varepsilon)$$

$$b = \frac{\Delta' A}{\varepsilon} = \frac{\max \| A(x) - A(x') \|_1}{\varepsilon}$$

$$= \frac{\max \| (f(x) + \eta) - (f(x') + \eta) \|_1}{\varepsilon}$$

$$= \frac{\max \| f(x) - f(x') \|}{\varepsilon}$$

$$= \frac{\max \| \frac{x_1 + \dots + x_n}{n} - \frac{x'_1 + \dots + x'_n}{n} \|}{\varepsilon}$$

$$= \frac{1}{n\varepsilon}$$

Since the element of x and x' are both 0 or 1 and x and x' only have one different element.

(b) For the same reason. set median of x as m_1 and median of x' is m_2
we have $\max \| m_1 - m_2 \| = 1$.

$$\Rightarrow b = \frac{\Delta' A}{\varepsilon} = \frac{\max \| m_1 - m_2 \|}{\varepsilon} = \frac{1}{\varepsilon}$$

(c) Median:

the case given $S = \{x_1, \dots x_n\}$ containing $\frac{n}{3}$ x's are ones and the rest are 0.

$\Rightarrow$ median of $S \triangleq$ median of $S' = 0$

i.e we don't need to add noise

Mean:

As the same description about b in (a),

we can have $b = \frac{\Delta' A}{\varepsilon} = \frac{\max \| \frac{n}{3}/n - (\frac{n}{3} \pm 1)/n \|}{\varepsilon} = \frac{1}{n\varepsilon}$

6. (a) matrix K to be postive semi-definate:

- $\forall\, q \in \mathbb{R}^n$, $q^T K q \geq 0$
- All eigenvalues of matrix K are non-negative

(b) Linear regression and Gaussian Processes.

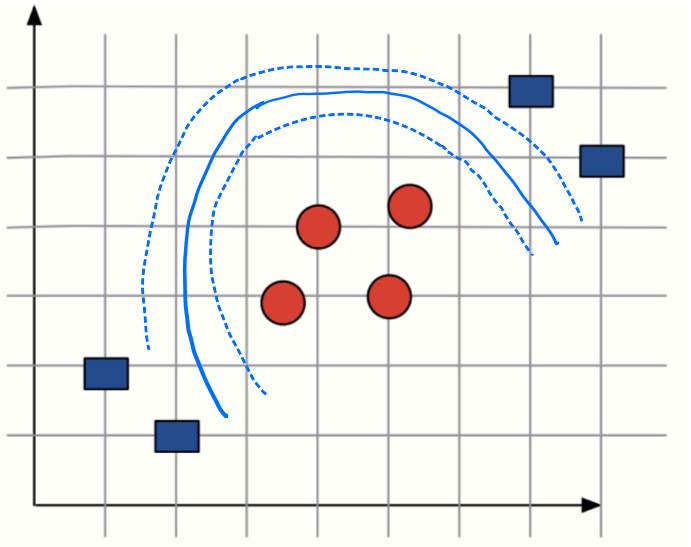
Kernelized linear regression:

$h(x) = w^T x$ where $w = X\vec{\alpha}$, $\vec{\alpha} = K^{-1}\vec{y}$ and $K_{ij} = k(\vec{x}_i, \vec{x}_j)$

Kernelized Gaussian process:

$k(x, z) = \exp\left[-\frac{(x-z)^2}{\sigma^2}\right]$

(c)



(d) as dimensionality d of data increases, i.e add more features of data which giving us more info. Hence, we could determine more support vectors to define the margin. So the ratio $\frac{m}{n}$ will increase as $m \uparrow$ and fixed n.

(e) When data is linear seperable with high dimensionality. For example, in handwritten recognition, we can rewrite objective using inner product and replace with $K(x, z) = x^T z$ that allow us to represent function in high dim. space with much lower computational cost.

(f) Gaussian Process.

When given a non-linear regression set, we couldn't apply linear regression model even with RBF kernel. First, linear regression is not suitable for non-linear data set. Second, RBF kernel is mainly used in calculation in high dim space which is not aim for regression.

And Gaussian processes are useful as a powerful non-linear multivariate interpolation tool.