

Statistical Learning

- **Reading Assignments**

S. Gong et al. *Dynamic Vision: From Images to Face Recognition*, Imperial College Press, 2001 (Chapt. 3, hard copy).

T. Evgeniou, M. Pontil, and T. Poggio, "Statistical Learning Theory: A Primer", *International Journal of Computer Vision*, vol. 38, no. 1, pp. 9-13, 2000 (on-line).

Statistical Learning

- **Assumptions**

- We consider features extracted from images to form probabilistic observation feature vectors (e.g., features extracted from face images).
- Their semantic interpretations or *labels* (e.g., identity associated with a face) are also subject to uncertainty.
- The relationship between the observations and their labels can be modelled probabilistically.
- Such models can be estimated through statistical learning.

- **What is the goal of statistical learning?**

- To estimate an unknown *inference function* (i.e., model between observations and their labels) from a finite, often sparse, set of observations.

Example: suppose $x \in X$ is a random observation vector (e.g., face) and $y \in Y$ is the interpretation of x (e.g, identity).

- * The probability of observing the pair (x, y) is given by

$$P(x, y) = P(y/x)P(x)$$

- * If $P(y/x)$ is known (i.e., inference function), then an interpretation y can be inferred for any given obserbation x .

- * It is possible to estimate the inference function from a set of observations and their labels through *inductive learning*.

- Once learning has been performed, we can predict the interpretations of novel observations.

- **Learning as function approximation**

Learning machine: an algorithmic implementation of some family of functions $f(x, a)$ where a is a parameter vector.

Loss $L(y, f(x, a))$: a function measuring the discrepancy between y and the approximation $f(x, a)$ found by the learning machine.

Risk $R(a)$: the loss integrated over the joint probability $P(x, y)$

$$R(a) = \int L(y, f(x, a))dP(x, y)$$

- Learning aims to find a function $f(x, a_0)$ that minimizes the risk.
- In this context, inductive learning can be defined as a problem of *functional approximation by risk minimization*.

- **Examples of loss functions**

$$L(y, f(x, a)) = [y - f(x, a)]^2$$

- Corresponds to *supervised learning*
- The function to be approximated in this case is $E[y/x]$ (called *regression* if y is continuous or *classification* if y is discrete)

$$L(y, f(x, a)) = -\log p(x, a)$$

- Corresponds to *unsupervised learning*
- The function to be approximated in this case is the *pdf* of x

- **Empirical risk minimization**

- The expected risk cannot be minimized directly because $P(x, y)$ is not known.
- One approach is to approximate the risk R by the empirical risk R_{emp}

$$R_{emp}(a) = \frac{1}{M} \sum_{m=1}^M L(y_m, f(x_m, a))$$

- The least-squares method minimizes the following empirical risk:

$$R_{emp}(a) = \frac{1}{M} \sum_{m=1}^M [y_m - f(x_m, a)]^2$$

(i.e., $L(y_m, f(x_m, a)) = [y_m - f(x_m, a)]^2$)

- The maximum likelihood approach minimizes the following empirical risk:

$$R_{emp}(a) = - \frac{1}{M} \sum_{m=1}^M \log p(x_m, a)$$

(i.e., $L(y_m, f(x_m, a)) = - \log p(x_m, a)$)

- **Capacity of learned functions**

- Minimization of the empirical risk does not guarantee good generalization.
- To guarantee an "upper bound on generalization error", statistical learning theory says that the *capacity* of the learned functions must be controlled.
- Functions with large capacity are able to represent many dichotomies.
- Need functions whose capacity can be computed (a frequently used measure for the capacity is the Vapnik-Chervonekis (CV) dimensionality).
- *Support Vector Machines*: functions of appropriate capacity are chosen using *structural risk minimization*.

- **Constructing a learning machine**

(1) *Loss function*: regression, classification, or *pdf*.

(2) *Family of functions*: multi-layer neural nets, radial basis functions, linear discriminant functions, Gaussian density functions, mixtures of densities.

(3) *Learning algorithm*: minimize some approximation of the expected risk.

- **Bias-variance dilemma**

- Simple parametric functions with only a few parameters introduce a strong bias.

- Functions with many parameters are more powerful but introduce high variance.

Learning as density estimation

- Bayesian inference is based on estimating the density function.

$$P(y/x) = \frac{P(x/y)P(y)}{P(x)}$$

- This is the most general and difficult type of learning problem.

Linear classification and regression

- **Linear functions**

- Linear functions can be written in the form:

$$f(x) = (w \cdot x) + b$$

- When $N=2$, then the above equation defines a line (a plane or hyperplane if $N > 2$).

<figure3.4 Gong>

- The position and orientation of the line are changed by adjusting (w, b) where w is the normal to the line.

- For a two-class problem, a linear classification function has the form

$$f(x) = \text{sign}((w \cdot x) + b)$$

• Learning using linear functions

Using SVD

- The classical least-squares method can be used to determine the parameters (w, b) .
- This can be done by minimizing the empirical risk:

$$R_{emp}(a) = \frac{1}{M} \sum_{m=1}^M [y_m - f(x_m, a)]^2$$

- SVD can be used in this case to determine the parameters.

Using SVM (Support Vector Machines)

- There are usually many hyperplanes that result in low empirical risk using a given data set.
- We need a method that allows to choose a hyperplane that yields low generalization error.
- Statistical theory says that the optimal line/hyperplane is the one giving the largest margin of separation between the classes.
- SVM implements the optimal hyperplane.

Using NNs (Neural Networks)

- We will consider the multilayer network trained by the backpropagation rule. - This model minimizes the empirical risk.
- Can directly construct the decision boundary based on training data, without estimating the probability density of each class.
- The decision boundary can be highly non-linear.

Nonlinear classification and regression

- Successful inference is not always possible using only linear functions.
- We can obtain families of non-linear functions by transforming the input x by a set of non-linear functions $\phi_k(x)$ called *basis functions*:

$$f(x, a) = \sum_{k=1}^K w_k \phi_k(x) + b$$

(a are the parameters in the above expression)

- The basis functions together perform a non-linear mapping:

$$\Phi: R^N \rightarrow R^K$$

where $\Phi = [\phi_1(x), \phi_2(x), \dots, \phi_K(x)]^T$

- Certain forms of basis functions will allow any continuous function to be approximated to arbitrary accuracy using $f(x, a)$.
- Multilayer neural networks (e.g., back-propagation, radial basis functions) are examples of models for estimating the parameters a .
- Support Vector Machines can also be generalized to handling non-linear regression/classification.