**Lecture 1:**

Classification deals with the problem of partitioning the data into pre-defined classes by a **decision boundary** or **decision surface**

The **input space** can also be called **instance space**

Empirical risk is the approximation of the error of the model:

True risk (generalization error):

Consistent hypothesis is a hypothesis that correctly classifies all training examples:

Version space

Most general hypothesis

Most specific hypothesis

Margin: The minimum distance between the decision boundary and a training point.

Confusion matrix:

False negatives: h(x) = 0 while y = 1

False positives: h(x) = 1 while y = 0

More specific: higher false negatives

More general: higher false positives

Some scores:

Precision:

Recall:

F1 score:

**Lecture 2:**

Probably approximate correct learning framework formalizes the notion of generalizations in machine learning

Ingredients (see the slides)

Goal: learn a hypothesis with

A class is **PAC-learnable** if there exist an algorithm that gives a hypothesis has a generalization error satisfies: for some chosen epsilon, delta, for any distribution D, and any concept C. Also, the sample size must grow polynomial in 1\epsilon, 1\delta.

Finite hypothesis classes arise when:

Input variables have finite domains or converted to such in preprocessing.

The representations of hypotheses have finite size.

Dealing with subclasses of Bool formulae.

For these classes, assuming there exist a consistent hypothesis. Pr(R(hs ) ≤ ǫ) ≥ 1 – δ if: m ≥ 1/ǫ(log(|H|) + log(1/δ))

An equivalent generalization error bound: R(h) ≤ 1/m (log(|H|) + log(1/δ))

For arbitrary Boolean formulae:

There are 2^d possible input vectors for a formula of d variables. |X| = 2^d

For a subset of input space X, there are 2^|X| ways of picking a subset of X

Thus, the number of required training examples is: m ≥ 1/ǫ ((2^d) log 2 + log(1/δ))

For inconsistent hypothesis, the general error bound is: R(h) ≤ Rˆhat(h) + sqrt(log(|H|) + log(2/δ) ) / 2m). This can be achieved with probability of at least 1 – δ.

**Lecture 3:**

Vapnik-Chervonenkis dimension, or VC dimensions, can be understood as measuring the capacity of a hypothesis class to adapt to different concepts. Or the size of the largest training set that we can find a consistent classifier for all labelings.

The underlying concept of VC dimension is shattering. A hypothesis class H is said to shatter datapoint S if for any possible partition of S into positive and negative subset we can find a hypothesis that correctly classifies all datapoints (h(x) = 1 if and only if x is in the positive subset).

To show that VC(H) = d:

There exist a set of input of size d that can be shattered by H (VC(H) >= d)

There does not exist a set of input of size d+1 that can be shattered by H (VC(H) < d+1)

VC dimension of finite hypothesis class: VC(H) ≤ log\_2(|H|)

Some examples of classes with a finite VC dimension (see the slide)

Generalization bound base on VC dimension d (see the slide)

Rademacher complexity: Use to measure how well does the hypothesis fit random noises.

Generalization bound with Rademacher complexity for a probability of at least 1 - delta: see the slide.

Rademacher vs VC dimension. One important characteristic is that VC dimension does not depend on training sample while Rademacher complexity does.

**Lecture 4:**

Stochastic scenario: Assume that outputs are probabilistic function of the inputs. This means that there might not always exist a target concept that has zero generalization error.

Underfit and overfit

Two general approaches to control the complexity of the model:

Select a hypothesis class.

Regularization: punish hypothesis that use too many parameters (too complex) by bounding the norm of the weights.

Bayes error: the minimal non-zero error for any hypothesis (the minimum possible error).

Bayes classifier: the hypothesis that achieve the Bayes error

The average error made by the Bayes classifier is called the noise

Excess error of choosing the hypothesis h instead of the Bayes classifier can be decomposed into the estimation error (the excess error of the hypothesis h over the best hypothesis in the class) (also called the bias), and the approximation error (the excess error of choosing the wrong hypothesis class instead of the best one) (also called the variance).

Structural risk minimization (see the slides, and note the generalization bound for SRM).

Regularization-based algorithms: an extension to uncountable union of hypothesis classes?

Bound of Rademacher complexity of L^2 norm case.

Evaluate the performance of the model on the test set

If the dataset is small, the training set is as big as possible, otherwise, it will be as big as the computational resources allow.

Stratification

Performance in cross validation is the average performance on all folds.

Hyperparameters are chosen such that they have the best average performance over the n validation test.

**Lecture 5:**

Linear classifiers have some attractive properties:

Fast to evaluate and takes small space (O(d) time and space)

Easy to understand

Low complexity: VC dimension = d + 1

With g(x) = w^T x

Geometric margin: yg(x) / ||w||

Functional margin: yg(x)

Perceptron:

For each iteration, find an incorrectly classified example (x\_i. y\_i)

Add that example to the weight vector: w(t+1) = w(t) + y\_i \* x\_i

Problem with zero-one loss:

Non-differentiable: cannot use gradient approaches

Non-convex: optimizer susceptible to fall into local minima

The logistic loss: see the slide

Minimizing the logistic loss is maximizing the probability of observing the data.

The loss changes fast when the margin is negative. Thus penalizing examples that are far in the incorrect halfspace. However, it does not give bonus for example that are far in the correct halfspace (not change much when margin is much positive)

Stochastic gradient descent

**Lecture 6:**

Support vector machine is used to find a hyperplane that have the largest minimum margin from all training example.

Support vectors are vectors that have functional margin of exactly 1

Hard margin support vector machine assumes linearly separable data.

Hinge loss: Comes from the constraint of the relaxed soft margin SVM.

On small data, we can use quadratic program solver to solve for the optimization of SVM.

On big data, we can use stochastic gradient to find the minimum.

Hinge loss is not differentiable at 1, but we can choose the gradient such that it is 0.

SVM always shrink the weight vector by some amount to regularize.

The weight vector is a linear combination of the training examples that have been updated thus far.

Dual representation of the soft margin SVM

**Lecture 7:**

Key characteristic of kernel methods:

Embedding

Linear models

Kernel trick

Regularized learning

Formally, kernel is an inner product in a feature space F.

Linear kernel: correspond to the dot product in R^n

Geometric interpretation of the linear kernel: the cosine of the angle between two feature vectors.

Kernel matrix: the matrix that record the kernel of two vectors. This matrix is symmetric.

The kernel matrix is positive semidefinite

The positive semi definite of the kernel matrix ensures that we can find a global optimum through gradient descent.

The dimension of the polynomial feature space is d+q choose q. Uwith d is the dimension of the input space and q is the degree of the polynomial.

The Rademacher complexity of the kernel methods are relates to the norm of the weight vector and the trace of the kernel matrix.

Some methods to design a kernel

**Lecture 8:**

Activation function should be non-linear function, else we would only output a linear model

The multilayer perceptron can represent all three basic logical operation AND, OR, and NOT. Thus, all Bool functions can be represent by the MLP.

Neural networks can fit any function, but this may require a large network.

Learning the optimal weights for MPs is computationally hard.

SGD in MLPs might result in local optimum. Thus, need to run many time with different initialization to find a good local optimum.

The update of the weight vector in the hidden layer is through backpropagation form the output layer to the hidden unit.

Uwe should first update the weight of the hidden layer using the old values of the weight for the output layer, and then update the output layer weights afterward.

**Lecture 9:**

Using majority vote as the prediction of ensemble, i.e., the result that matches the majority of the base classifiers.

The probability that majority vote is incorrect given the same empirical risk for each base classifier and the probability that they predict wrong is independent.

Diverse base classifier = classifiers that have different error pattern (make errors on different training examples)

Compare the loss of the average model with the average loss of small models

For averaging model, diversity always help

For majority voting model, diversity will help depends on the context.

The definition of weakly PAC learnable

Logic of AdaBoost: each iteration, the weight of examples that are classified correctly will be down weight, while incorrectly classified examples are up-weight (so that models that classified those correct are easier to be choose)

Note the weight for the example D and the weight for the model alpha. (the weight for the model is fixed after set, while the weight for the example can be changed on the run)

If the examples are noisy, AdaBoost will give high weight for misclassified examples and easily overfitting

Even if the AdaBoost model is complex, empirically the testing error still go down (no overfitting observed)

**Lecture 10:**

Example and reasoning of variable transformation

Variable selection

Variable ranking

Procedures for generating subset:

Forward selection

Backward elimination

Wrapper approach might be computationally heavier but can also find better subset than the filter approach.

Backward elimination might be better at finding the suitable subset

Sparse matrix: matrices with mostly zeros.

Sparse model: use p-norm

If p < 1: Sparse but hard to optimize

If p > 1: Not sparse

P = 1 is the solution that give the sparsest while easy to optimize

Use: combine the l1 regularization with the loss function

Note: Sparse regression (Squared loss + sparse regularization) is also called the LASSO

Problem with variable subset selection and sparse modelling is their sensitivity to small perturbations

**Lecture 11:**

One versus all approach: train a model for each class

Easy to implement and relatively efficient, however suffer from class imbalance and calibration problem.

One versus one approach: train a model for each pair of two classes (1 if x is in the first class, -1 if x is in the other class)

Need to train more classifiers, but the training set are smaller since they only contain examples from the two sets at a time (example are not in either of the set are excluded). Less likely to suffer from imbalance

Each column of the ECOC code book has a classifier

Multiclass SVM and multiclass AdaBoost: mostly is that now the problem has k weight vectors, or the labels are represented as matrix.

**Lecture 12:**

Types of preferences: absolute preference (object score) and relative preference (pairwise comparisons)

Ranking function is a bijective function (give rank to object and retrieve the object of the rank)

Kendall’s distance: how many pair that are ranked incorrectly

Note the ranking scoring (agreement) and the ranking extraction