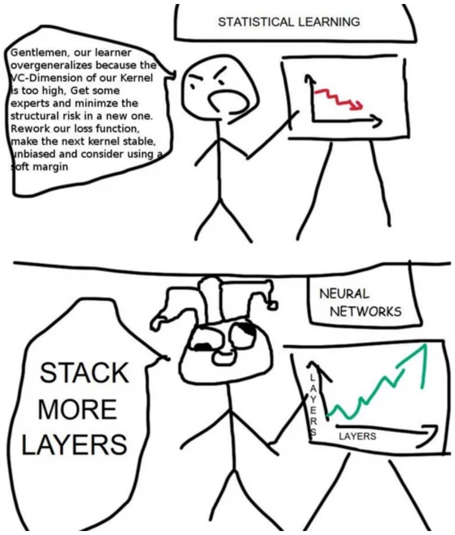
Chapter 3.4: Non-linear Transformations

Overview

In this section, we will see how to build non-linear models out of linear models. This will allow us to easily construct models with different VC dimensions that are appropriate for our datasets.


STATISTICAL LEARNING
Gentlemen, our learner overgeneralizes because the VC-Dimension of our Kernel is too high, Get some experts and minimze the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin
NEURAL NETWORKS
STACK MORE LAYERS
LAYERS
LAYERS

Definition 1. Define a *feature transform* as it relates to the perceptron hypothesis class.

Generic Feature Transforms

Problem 1. The *polynomial feature map* is a generalization of the examples above. It is one of the most popular feature maps due to its simplicity, and it is the main example in the textbook (see pages 99-104).

The polynomial map of degree Q (denoted Φ_Q in the book) has one feature for each of the Q degree monomials formed by the input data dimensions. For example, in 2 dimensions, the 2nd degree polynomial feature map is

$$\Phi_2(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, x_1x_2) \tag{1}$$

and the 3rd degree polynomial feature map is

$$\Phi_3(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_2^3, x_1^2x_2, x_1x_2^2). \tag{2}$$

1. How does the choice of Q affect E_{in}?

The following two links visualize this transformation. The first link is closely related to the examples in Figures 3.5 and 3.6; the second link is an entirely separate example.

(a) `https://www.youtube.com/watch?v=3liCbRZPrZA`

(b) `https://www.youtube.com/watch?v=ndNE8he7Nnk`

2. What is the *universal approximation property*?

3. The VC dimension of the hypothesis space $\mathcal{H}_\Phi$ equals $\tilde{d}$. What is a formula for $\tilde{d}$ (and thus the VC dimension) based on k and d?

4. When would we want to use the polynomial kernel?

Problem 2. The PCA feature map is an example of a *dimensionality reduction* technique. It is defined to be

$$\Phi(\mathbf{x}) = \mathbf{x}^T A \tag{3}$$

where A is a $d \times \tilde{d}$ matrix with the ith column equal to the ith eigenvector of $X^T X$, and $X : N \times d$ is the matrix of all data vectors. The A matrix is constructed this way because this selects a subspace with "maximum variance".

NOTE: For a cool example with "eigenfaces", see the scikit-learn documentation: `https://scikit-learn.org/stable/auto_examples/applications/plot_face_recognition.html`.

NOTE: Most explanations of PCA focus on why this choice of matrix maximizes the variance and ignore the statistical properties of classifiers created with PCA features. What actually matters in practice are these statistical properties, and so that's what we'll focus on here. Sometimes, machine learning technical interviewers will ask about why the eigenvalues maximize the variance, and so while you don't need to know that for this class (and it is not covered in the textbook), it is worth reviewing before interviewing. For an explanation, see stackoverflow: `https://stats.stackexchange.com/a/140579/16243`.

1. How does $\tilde{d}$ affect E_{in}?

2. How does $\tilde{d}$ affect the VC dimension (and thus the generalization error)?

Popular Models as "Just" Feature Maps

Problem 3. The random feature map is defined as

$$\Phi(\mathbf{x}) = \mathbf{x}^T A \tag{4}$$

where A is a random $d \times \tilde{d}$ matrix. Any distribution can be used to select the entries of the A matrix, but some common choices are the uniform distribution over $(-1, 1)$ or the standard gaussian distribution with mean 0 and variance 1.

The hypothesis class $\mathcal{H}_\Phi$ (i.e. random features with the perceptron hypothesis class) is often called the *random kitchen sink*. It was first introduced in 2006, and in 2017 it received the "Test of Time" award at the NIPS conference (now called NeurIPS). This random kitchen sink won this award because it remains state-of-the-art for MANY applications. You can watch the award presentation at `https://www.youtube.com/watch?v=ORHFOnaEzPc`.

1. How does E_{in} vary with $\tilde{d}$?

2. How does E_{in} for random features compare to the E_{in} for PCA?

3. What is the VC dimension if $\tilde{d} \leq d$?

4. What is the VC dimension if $\tilde{d} > d$?

5. Under what conditions would you want to use the random feature map or PCA?

Problem 4. (See Example 3.15 in the textbook.) Define the feature map

$$\Phi_{(k)}(\mathbf{x}) = (1, x_k) \tag{5}$$

to extract the kth coordinate from the input data point and let $\mathcal{H}_{\Phi_{(k)}}$ be the hypothesis class of perceptrons with the above feature map. The *decision stump* hypothesis class is defined to be

$$\mathcal{H}_{\text{stump}} = \cup_{k=1}^{d} \mathcal{H}_{\Phi_{(k)}}. \tag{6}$$

1. State an upper bound on the VC dimension of $\mathcal{H}_{\text{stump}}$ in terms of d using big-O notation. (Hint: This bound is given in the textbook in a form that doesn't use big-O notation.)

2. The decision stump is a popular model for datasets with many dimensions but few data points. Explain why using VC theory.

Single Feature Transformations

Problem 5. Many features are "discrete". That is, they have a fixed number of values where the value has little semantic meaning. For example, marital status in the famous German Credit dataset[1] is encoded as a column with the following semantics:

column value	semantic meaning
1	single - never married
2	married - first marriage
3	single - divorced
4	single - widowed
5	married - remarried (female)
6	married - remarried (male)

The *one-hot encoding* converts these discrete columns into multiple boolean columns, one for each value in the original column. The one-hot encoding for marital status above would convert a single column into 5 separate columns in the feature space. This conversion often greatly improves E_{in} because a linear separation of the original discrete values often has no semantic meaning.

1. If the discrete column has c possible values, then the feature space has $\tilde{d} = d + c - 1$ columns. What is the VC dimension of $\mathcal{H}_\Phi$?

2. The number of discrete values can sometimes be extremely large. For example, c might be the city that a person was born in, and there are millions of cities in the world.

   (a) From a VC perspective, why is this a problem?

   (b) How can we fix this problem?

[1] http://archive.ics.uci.edu/dataset/144/statlog+german+credit+data

Other Operations

Problem 6. The *mean centering* feature transformation is defined to be

$$\Phi(\mathbf{x}) = \mathbf{x} - \mu, \tag{7}$$

where

$$\mu = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i. \tag{8}$$

1. What is the VC dimension of $\mathcal{H}_\Phi$?

2. When should you use the mean centering feature transformation?

Problem 7. The *range normalization* feature transformation rescales all of the features to between $[-1, 1]$. It is defined to be

$$\phi(\mathbf{x})_i = x_i/\alpha_i \tag{9}$$

where α_i is the maximum absolute value of the x_i component of all the training datapoints.

NOTE: The lowercase ϕ instead of the uppercase Φ above is intentional. It is the book's notation to let you know that the index refers to a specific column of the output of the Φ function.

1. What is the VC dimension of $\mathcal{H}_\Phi$?

2. When should you use the range normalization transformation?