Differentiable Kernel Regression via Convolutional Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Kernel methods have relied on parametric kernels and cross validation to recover the kernel family and optimal parameters. This severely limits the power of these methods in kernel density estimation, regression, and any consequent tasks. In this paper, we provide a simple formulation of kernel regression based on convolution, which enables us to learn the kernel function and parameters directly from temporal data. We show how learning kernels improves the quality of density estimation and regression, compared to traditional cross validation within parametric families of kernels. Moreover, we show our method can easily extend to multivariate input with structural and temporal dependencies, and is able to recover dependency structure on the input time series automatically.

1 Introduction

In several areas of data science and machine learning, tractability of computations has often been traded off with simplifying assumption on the dependencies and distribution of the random variables of the model.[] [] [] [] As the size of datasets are growing, these assumptions begin to hurt more than they help [] [] []. Kernel density estimation[] is one of the methods which does not impose most of the simplifying assumptions on the distribution of the data[], and has been previously studied in the context of density estimation, regression, causal inference, conditional distribution modeling, among others. Kernel regression[] in particular, is a general method for estimating any function, given samples from the true distribution and a kernel similarity function.

While the theory of kernel regression allows the use of any positive semi definite kernel, in practice, methods only cross-validate over a few well-known kernel functions such as Radial basis(Gaussian,) Laplace, etc. and fail to consider the entire space of legal kernels. Of course when one leaves this step to cross-validation nothing more is reasonable to expect. In this paper, we try to liberate the kernel regression method from searching within a set of pre-defined kernel families.

In particular, we focus on irregularly measured temporal time series, and show how to construct a differentiable formulation of kernel density estimation for single time series as well as multiple dependent time series. Our method is limited to discrete time series, and we use a differentiable formalization and gradient descent to estimate the kernel function over its domain. As we show, not only this method is easily able to recover multivariate kernels, it can also recover spatio-temporal dependencies between the random variables.(maybe! we'll see)

2 Kernel Regression

Imagine the input to be samples from D time series, each sampled irregularly. An example context would be different types of lab measurements for a patient at different time points across their life.

We denote the samples as $x^1_{t^1_1}$, $x^1_{t^1_2}$, ..., $x^1_{t^1_{n_1}}$, ...,$x^D_{t^D_1}$, $x^D_{t^D_2}$, ..., $x^D_{t^D_{n_D}}$, where x^d refers to time series d and t^d_1,... $t^d_{n_d}$ refer to the time points over which time series d is sampled.

Kernel regression provides a general formalism for estimating any function with additive noise. Let's start from a single time series, $x(t)$. Kernel density estimation assumes the following.

$$x = f(t) + \epsilon$$
$$\epsilon \sim N(0, \sigma^2)$$

Given samples x_{t_1},...x_{t_n} from a time series, general function regression with additive noise lets us estimate the value of x at a new time point t_{new} as follows.

$$x(t_{new}) = \mathbf{E}_{x \sim P(x|t=t_{new})}[x]$$
$$\mathbf{E}_{x \sim P(x|t=t_{new})}[x] = \int_x xP(x|t=t_{new})dx = \int_x x\frac{P(x, t=t_{new})}{P(t_{new})}dx$$

At this point, one can use kernel density estimation to estimate the probabilities $P(x, t = t_{new})$ and $P(t_{new})$ from the training data. Nadaraya[] and Watson[]showed that using a positive semidefinite kernel function $K(t, t')$, the nonparametric regression formulation is reduced to the following.

$$\mathbf{E}_{x \sim P(x|t=t_{new})}[x] = \frac{\sum_{i=1}^n x_{t_i} K(t_{new}, t_i)}{\sum_{i=1}^n K(t_{new}, t_i)}$$

We can now rewrite the nonparametric regression using convolution operator. Denoting convolution operator as $*$, i.e. $(k*f)(t) = \int_\tau K(t-\tau)f(\tau)d\tau$, and having the observed nonzero samples $\bar{X}_{train}$ = x_{t_1},...,x_{t_n}, the numerator of the kernel regression is equal to the following.

$$\sum_{i=1}^n x_{t_i} K(t_i - t_{new}) = (K * \bar{X}_{train})(t_{new})$$

The numerator $P(t_{new})$ can similarly be written as a convolution of the kernel function with a sequence of 1s at each point at which we have a sample.

$$\sum_{i=1}^n K(t_{new}, t_i) = (K * (\bar{X}_{train} \neq 0))(t_{new})$$

So the kernel regression formulation of Nadaraya and Watson reduces to the following formulation.

$$\mathbf{E}_{x \sim P(x|t=t_{new})}[x] = \frac{(K * \bar{X}_{train})(t_{new})}{(K * (\bar{X}_{train} \neq 0))(t_{new})}$$

The benefit of this formulation, is that we can now differentiate this ratio with respect to each $K(\tau)$ using function composition, and estimate the optimal value of the kernel at each position τ within the kernel domain. We can also plug in the kernel regression module within any subsequent differentiable operators and perform multiple tasks such as classification, reconstruction or beyond via gradient descent.

This formulation easily extends beyond single time series.

3 Related Work

4 References