Theory Guided Data Science

Tyler Chang
December 10, 2018

Regularization

For an optimization problem of the form $\min_{x \in X} f(x)$, a *regularization term* is an additional term $g(x)$ that encodes some structural or physical requirement on solutions. I.e., the problem is replaced by $\min_{x \in X} f(x) + g(x)$, where $g(x)$ is large when x does not agree with physical intuition. For example, g is often taken to be the Hamming function ($\|.\|_0$) when x is expected to be sparse.

Dimension Reduction

When the dimensionality (d) of the data is greater than the physical dimensionality of the problem (k), several techniques can be used to reduce the dimension:

- *Principle Component Analysis* (PCA) is an unsupervised technique, in which input data is projected onto a subspace of dimension k in which it has maximal variance (or equivalently, in which the sum of squared residuals for the projections is minimized). Computationally, this is done by constructing the covariance matrix for the data (Z) then performing an Eigenvalue decomposition of Z. The first k Eigenvectors of Z (sorted by their corresponding $|\lambda_i|$) are then the basis vectors for the projected subspace.

- *Supervised PCA* is similar to PCA, but takes place in a supervised setting. Instead, a plane is fit to the response values, whose slope is determined by a matrix A. Then standard PCA is performed on the covariance matrix Z, rescaled by A to account for response values.

- *Linear Discriminant Analysis* (LDA) is a supervised technique, in which a linear separator is found between different classes, then the dimension is reduced to this separator's orthogonal complement.

- *Autoencoders* are neural networks that map input data directly into an intermediate layers with fewer neurons, forcing the network to collapse certain dimensions. This can be done in either a supervised or unsupervised setting. If linear activation functions are used with a single reduction layer, this is equivalent to PCA.

- *Correlation Analysis* is a supervised technique, where a model (such as a linear or quadratic least-squares fit) is fit to the input data, then the magnitude of each coefficient in the model is analyzed for statistical significance.

Interesting Problems

Inverse Problems

Given some physical model $f(x, z)$ (where $z \in \mathbb{R}^n$ denotes some vector of unknown hyperparameters) and a set of labeled data points X, the *inverse problem* is to find the hyperparameters $\hat{z}$ such that $f(\tilde{x}, \hat{z}) \approx \tilde{y}$ for all pairs $(\tilde{x}, \tilde{y}) \in X$. This problem is often solved using the *adjoint gradient* method, where (similarly as in gradient descent) the adjoint gradient operator is computed based on the current fit $f(x, z)$, and then is applied and updated iteratively, until $z \to \hat{z}$. This problem is much harder in the high-dimensional case (where $n >> |X|$) since the solution is underdetermined. Often, some kind of regularization can be used in this case to approximate a solution with good physical properties.

PDE Fitting

The general form for a partial differential equation is

$$F_t = N(F, F_{x_1}, F_{x_2}, \ldots, F_{x_1x_1}, F_{x_1x_2}, \ldots x, \mu)$$

where N is a nonlinear function of $F(x, t)$, its mixed partials, the spatial vector x, and some unknown hyperparameters μ. If U is a vector of observations of the value of $F(x, t)$ at different points in phase space, one could seek to solve for the underlying PDE by fitting a nonlinear function N. One solution is to build a data matrix

$$\Theta = \begin{bmatrix} 1 & U & U^2 & \ldots & U_x & U_{xx} & \ldots & Q \end{bmatrix}$$

where each column of Θ corresponds to a potential term in N. Then, the nonlinear function N can be approximated by solving the least squares problem

$$\min_{\xi} \|\Theta\xi - U_t\|.$$

Finding Graph Structure

Consider a general graph Γ, with nodes X but unknown edges, where time series data for some physical function $f(x, t)$ is known at each $x \in X$. Presumably if $x, y \in X$ share an edge, $f(x, t)$ will vary similarly through time as $f(y, t)$. Therefore, we can solve for graph structure by computing the correlation of $f(x, t)$ and $f(y, t)$ for all pairs of nodes $x, y \in X$.

Navier-Stokes Equations & the Closure Problem

The Navier-Stokes equations are a system of PDEs that govern viscous fluid flow. There are two types of flow, laminar and turbulent, the latter of which is typically of interest. To simulate turbulent fluid flow, one can solve the Navier-Stokes equations through time using direct numerical simulation (DNS), however this is typically too expensive. Alternatively, one could uniformly approximate the average fluid pressure in a turbulent region by its Reynolds number, without considering fluctuations. The solution to this problem is computationally feasible but much less accurate. The *closure problem* is the problem of approximating the error term in the Reynolds approximation.