A short summary on data-driven computing

Jingwen Zhao 2021/7/6

# The data-driven computing paradigm

The data-driven computing is a method designed to tackle large amounts of data. The key difference between the data-driven computing and traditional method is that the data-driven paradigm is independent of any empirical material modelling, therefore bypassing the potential errors and uncertainty in modelling. Meanwhile, the data-driven computing still ensures the result follows the fundamental compatibility and conservation laws, in contrast to techniques such as machine learning.

Depending on the specific problem studied, the conservation law and compatibility law can be written in various forms. In the simplest case of potential field, denoted as scalar , the field describes the global state of the system. Here the localization law extracts from the local state at a given material point is , along with appropriate boundary conditions. The flux follows the conservation law where is the source density. The pair hence describes the local state of a system at a given material point. The collection of these state functions z defines the global phase space Z and from it the constraint set .

In the case of a simple linear elastic problem, the scalar field now represents the displacement field. And the strain tensor field is extracted as by compatibility. The stress tensor field is extracted from the body force as by equilibrium. Along with the boundary conditions and , where is boundary displacements, outer normal and applied tractions. The above four equations form the constraint set in this problem setup.

In the case of a truss, the simplified material behaviour can be characterized by the uniaxial strain and uniaxial stress in the bar. Same as above, the state is subjected to the compatibility and equilibrium constraints. The compatibility law , where u is the array of nodal displacements and the matrix describing the geometry and connectivity of the truss. The equilibrium law , where f is the array of applied nodal forces. The constraint set is thus constructed.

# Solving the data-driven problem

Two sets are set up and ready to be solved. Firstly, the constraint set C, which contains all the local points which satisfies the equilibrium and compatibility law, along its boundary conditions. This set is universal and material independent. Secondly, the material set E, which imperfectly characterises the material law by a set of local states.

It can be proved that an ideal noise-free material data set which satisfies the following conditions gives solutions that converges to the classical solution. A limiting material law is represented by a graph E in phase space, and that a sequence of material data set satisfies: i) there is a sequence such that , for all , and ii) there is a sequence such that , for all . This is a good sanity check which helps proving the validity of the algorithm, however not overly useful in solving engineering problems.

While trying to solve the problem, generally, it is nearly impossible to directly find the intersect of the constraint set and material data set, various relaxation is therefore needed. One of the simplest ways to do it is finding the closest point in the material dataset that is closest to the constraint set, through a properly defined distance d. However, this is prone to be affected by outliers caused by noise. The solver can be further updated to be probability driven, minimizing the weighed distance to a cluster of nearby material data points. The problem is then converted to minimizing the equivalent free energy over the constraint set, where and is given by the Boltzmann solution and a Pareto weight. This equation maximizes the Shannon’s information entropy while placing a lower weight for points distant from z.

A simulated annealing procedure is proposed to tackle the minimization as the free energy is strongly non-convex. However, at sufficiently small , F is convex. The initial condition can be , with . Then the value can be iteratively increased according to and . Here Pc is the closest-point projection to C. Furthermore, a control value on the annealing rate can be set as , where is the result of the above iteration value. The solver operates in this manner until the values for is large. Then it proceeds by distance minimization until convergence is reached.

As discussed in the last chapter, this algorithm can be further improved. The width of the distribution is of order , therefore it roughly only looks at data points within this range. In the early stage of the annealing, the algorithm looks at almost all the data points so it can be speed up by appropriate subsampling or summarizing. In the late stage of the algorithm only the data near the current point matters so the dataset can be truncated to reduce calculation as well.

# Few questions

1. In the truss example, they wrote out two equilibrium equations satisfied by the point in C closest to a given point in phase space.

How are these equations used in the algorithm? Is it connected to the Pc used to calculate in the iterative steps?

1. The function is achieved from theorem 4.1, which guarantees the contractiveness of the function in its neighbourhood, why do we still need to set to limit the annealing rate? (Admittedly, a smaller does gives better result)
2. In the truss example, the author assumes each bar might behave differently according to their different local data sets . Then what does it mean to superimpose the data points from the experiment/simulation result on one graph in figure 2b)?