MSystems Litepaper

Background: The problem of building AI models

Increased demand for AI models, spurred by deep learning, has created the need to rapidly develop and iterate on machine learning models and deploy them at scale - both large and small. A contemporary, 'big data', approach taken by big-tech companies is to combine large datasets to train deep learning systems on huge computing clusters, resulting in a spiral of increasing costs to train models like ChatGPT have reached tens of millions of dollars, with maintenance costs in the >$100k/day. These huge computing requirements, alongside the limited interpretability and lack of robustness of deep learning systems has limited their uptake in many areas of science and industry where simpler, interpretable models based on statistical machine learning are the dominant.

Such models support the human analyst, providing insights into domain specific applications and are built, deployed and maintained by comparatively smaller teams. These systems are much more numerous and span a wide variety sectors - often being custom-build a specific application domain from the bottom up. Similarly to deep learning systems it has remained challenging to design, develop and deploy such systems - with companies requiring teams with diverse skill sets ranging from application-specific domain expertise through to mathematicians and data scientists, DevOps engineers and hardware engineers, to successfully design, develop, deploy and maintain them. This is partly due to the wide range of algorithms available to tackle a given problem and a lack of unified frameworks that can address many analysis tasks in a general way.

Vision: A platform for Interpretable AI models, designed by and for humans, assisted by machines

MSystems aspires to become a platform for the design and deployment of AI models by establishing itself as a leading provider to model design, construction and deployment tools for statistical machine learning enabling individuals, small teams and large corporations to accelerate development of AI models in a range of domains - from scientific R&D to safety-critical infrastructure. We will achieve this by providing an integrated toolchain taking a high level machine learning model definition, is interpretable and developed by domain experts and synthesizing from it a set of machine learning algorithms and low-level code that can be subsequently compiled and deployed onto a range of different compute targets - from low-power embedded devices through to high performance compute clusters. In this way we aim to

drastically accelerate the process of model design and deployment, minimizing the time and resource investment required to field advanced analytics systems.

Challenge: Rapid acceleration of model development and deployment

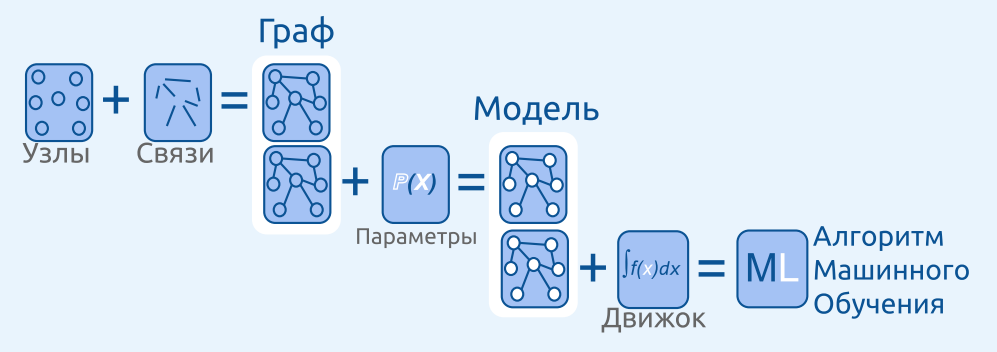
While the increased demand for AI models is undeniable, practically developing AI models for an application requires expertise in a range of disciplines ranging from application domain experts to statisticians and data scientists, DevOps engineers and in some instances FPGA and electronics engineers. In order to be successful these teams must successfully communicate across the boundaries of their disciplines to identify system requirements and, followed by a lengthy component design process, integrate the separate
solution stack to form the final product with a tailored algorithm and infrastructure to support it. This process is lengthy and error prone which introduces significant risks of the developer.

Solution

To enable rapid design and deployment of AI models we need to decouple model specification from the algorithms needs to run inferences. In doing so we will allow designers to rapidly iterate on models without worrying about low-level details. At the same time - the work of engineers on improving and testing general purpose inference engines will simultaneously benefit all models not just one.

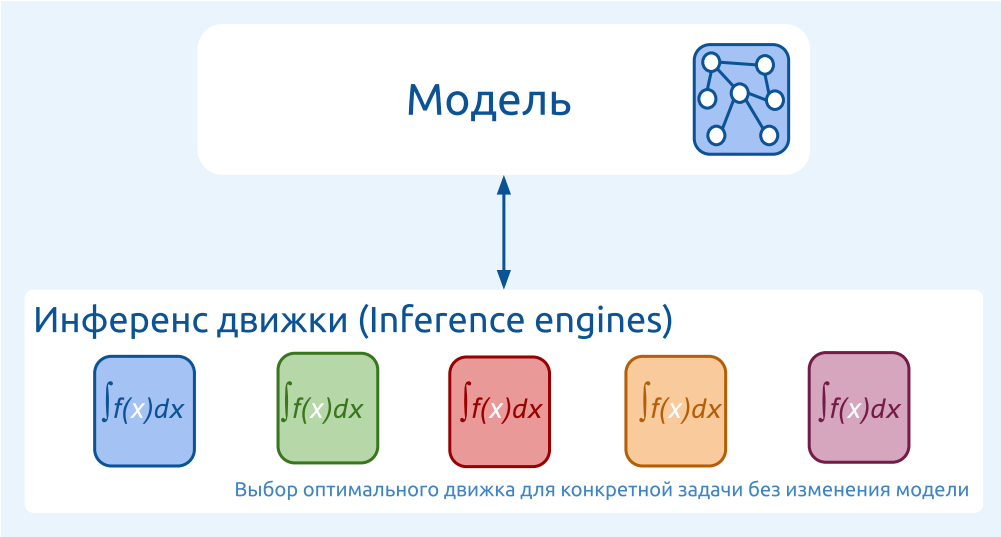
Probabilistic Graphical Modelling

We can achieve this using the language of probabilistic graphical models, which can represent complex, multidimensional probability distributions using a graph structure - where nodes represent the model variables, while edges encode the interdependencies between them. By processing a given probabilistic graphical model with an inference engine we can synthesize a machine learning algorithm.



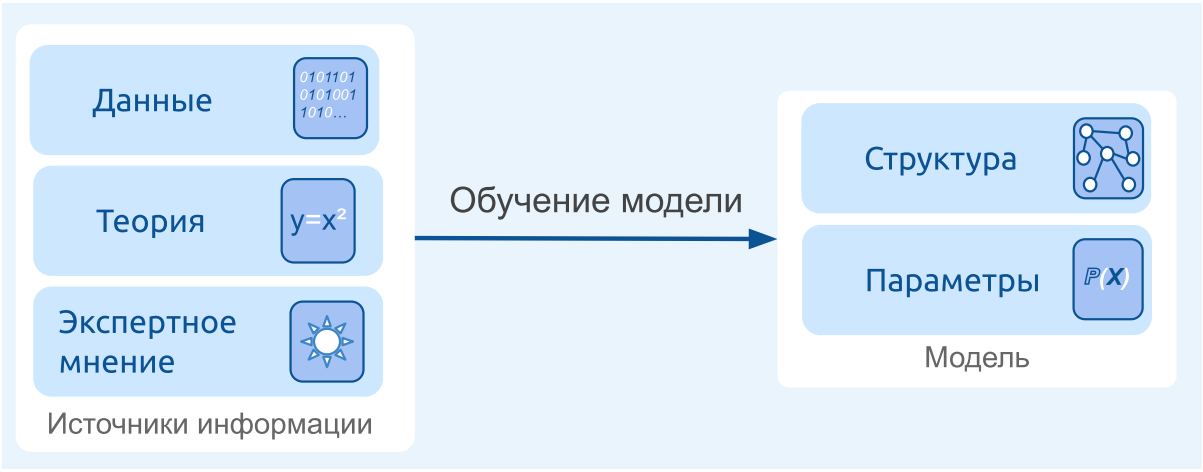
Синтез алгоритма машинного обучения из вероятностной графовой модели (модель).

The structure of the model in the form of a graph allows users to visualize the internals of the models, as well as shows the assumptions the model makes. If a given assumption is not suitable it can be modified by updating the graph, which then generates a new, updated machine learning algorithm.



Разделение модели и движка позволяет выбрать движок под конкретную задачу

As well as modifying the model we can also optimize inference engine to optimize the resulting machine learning algorithm to a given application and query type. This allows the same model to be deployed onto different compute modules and operating conditions from low-power IoT devices to real-time and high-reliability systems.



Модель можно обучить как из данных так и из других источников знаний

Unlike conventional deep learning and machine learning systems, probabilistic graphical models are interpretable. In addition a probabilistic graphical model can be trained by a combination of methods that are not possible with conventional machine learning and deep learning techniques - including the use of subjective expert judgement, theoretical results in the form of invariants and constraints, as well as both 'big' and 'small'. The latter point is of particular interest to 'data-sparse' domains where datasets difficult to produce or are not sufficiently big for adequate training of deep learning systems.

Model Designer - mBayes

The first part of achieving this decoupling is to provide users with low-code tools that allow creation and editing of models and inference them during the exploratory and prototyping phases of a project. By being low-code this systems allows domain experts to work directly with the model. At the same time, the ability to launch different inference algorithms and generate code from the model will accelerate subsequent development stages - allowing developers to experiment with different configurations before investing significant resources into implementing a solution.

Редактора позволит быстро разработать и запустить прототип

Probabilistic Modelling Software Suite - mCore

To support the low-code editor as well as provide advanced users and developers with tools for model design, inference and we have developed a software suite *mCore* that supports users in all stages of the development of probabilistic models, including structure learning and parameter calibration modulus as well as a set of inference engines and code generation toolchains.

Система позволит разделить задачи по созданию и реализации модели между командами

Combined with the low-code editor *mBayes*, the software suite *mCore* allows teams to decouple model prototyping and implementation by letting different teams focus on tasks falling withing their expertise - allowing for rapid iteration and accelerated model development.

This is achieved by using the probabilistic model as a unified representation that can be utilised both by domain experts when performing exploratory analysis, model tuning and refinement and by developers who can programatically manipulate the graph data structure, work with exported code or raw probability distributions to transfer the developed model onto a computer architecture suitable for the given application.

Запуск модели в облачном сервисе поможет быстро разработать прототип модели, а синтез кода ускорит реализацию в железе.