3. Выгодчикова И.Ю. Инструментарий принятия решений об инвестировании крупных российских компаний с использованием иерархической процедуры ранжирования и минимаксного подхода // Прикладная информатика. 2019. Том 14. № 6 (84). С. 123-137.

4. Markovitz H.M. Portfolio selection // J. of Finances. 1952. Vol. 7, №1.

Э.С. Кодиров, А.В. Махкамов, Ш.Ш. Абдувахобов

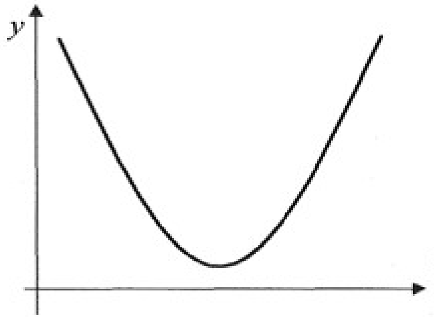
ОСНОВНЫЕ ИНСТРУМЕНТЫ МЯГКИХ ВЫЧИСЛЕНИЙ: НЕЧЕТКИЕ ЛОГИЧЕСКИЕ СИСТЕМЫ

(ТУИТ им. Мухаммада Аль-Хоразмий, Фергана, Узбекистан)

В последние годы нейронные сети, модели нечеткой логики и опорные векторные машины используются во многих различных областях. В этом разделе в первую очередь обсуждаются модели NN и FL. Однако из-за очень высокой степени сходства между NN и SVM почти все комментарии о репрезентативных свойствах NN также могут быть применены к SVM. Модели NN и FL являются инструментами моделирования. Они действуют таким же образом после завершения этапа обучения НС или внедрения человеческих знаний о какой-то конкретной задаче ПЛ. Это две стороны одной медали. Будет ли более подходящим инструментом для решения данной проблемы модель NN или FL, зависит от наличия предшествующих знаний о моделируемой системе и количества измеренных данных процесса. На полюсе НЕТ есть проектная ситуация черного ящика, в которой процесс полностью неизвестен, но есть примеры (измерения, записи, наблюдения, образцы, пары данных). На другом полюсе (модель FL) решение проблемы известно, то есть существует структурированное человеческое знание (опыт, знания, эвристика) о процессе. Тогда есть ситуация белого ящика. Короче говоря, чем меньше предшествующих знаний, тем больше вероятность того, что для попытки решения будет использован подход NN, а не FL. Чем больше знаний доступно, тем больше подходит проблема для применения моделирования нечеткой логики. В целом оба инструмента ориентированы на решение задач распознавания (классификации) и регрессии (приближение многомерной функции). Например, когда они применяются в области управления системой или в области обработки цифровых сигналов, нейронные сети можно рассматривать как инструмент нелинейной идентификации. Это наиболее тесная связь со стандартной и хорошо развитой областью оценки или идентификации линейных систем управления. Базовые зависимости (если они есть) обычно далеки от линейности, и линейные допущения больше не могут иметь места. Придется добавить новый скрытый слой нейронов. Таким образом, сеть может моделировать нелинейные функции. Этот этап проектирования приводит к огромному увеличению возможностей моделирования, но за это приходится платить: придется выполнять нелинейное обучение, а это, как правило, непростая задача. Однако именно здесь начинается мир нейронных сетей и опорных векторных машин.[1]

Основы нечеткого логического моделирования

Нечеткая логика находится на противоположном полюсе системного моделирования по сравнению с методами NN и SVM. Это подход белого ящика в том смысле, что предполагается, что человечество уже знает о решении. Следовательно, смоделированная система известна (т.е. белая). На уровне приложений FL можно рассматривать как эффективный инструмент для встраивания структурированных человеческих знаний в полезные алгоритмы. Это ценный инженерный инструмент, разработанный для того, чтобы хорошо сочетать точность и значимость. В этом отношении модели FL делают то, что люди делали очень давно. Как и в человеческих рассуждениях и умозаключениях, истинность любого утверждения, измерения или наблюдения зависит от степени. Эта степень выражается через функции принадлежности, которые количественно определяют (измеряют) степень принадлежности некоторых (четких) входных данных заданным нечетким подмножествам. Область нечеткой логики очень широка и охватывает множество математических и логических концепций, лежащих в основе приложений в различных областях. Основы этих концептуальных основ описаны в главе 6. В частности, в этой главе представлены фундаментальные концепции четких и нечетких множеств, вводятся основные логические операторы соединения (AND), дизъюнкции (OR) и импликации (IF-THEN) в области нечеткой логики (а именно, F-норм и F-конорм) и обсуждает эквивалентность NN и FL моделей. Кроме того, вводятся нечеткие аддитивные модели (FAM), которые являются универсальными приближениями в том смысле, что они могут аппроксимировать любую многомерную нелинейную функцию на компактной области с любой степенью точности. Это означает, что FAM плотны в пространстве непрерывных функций, и они разделяют это очень мощное свойство с NN и SVM.



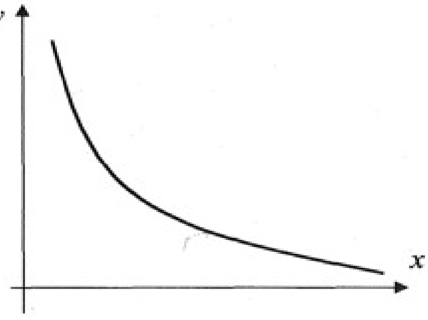


Рис. 1. Две разные нелинейные функции (отображения) 9t 'x 9t', моделируемые нечеткой аддитивной моделью.

В этом разделе обсуждается, как FAM аппроксимируют любую (не аналитически, а вербально или лингвистически) известную функциональную зависимость. FAM состоит из набора правил в форме операторов IF-THEN, которые выражают человеческие знания о функциональном поведении. Предположим, мы хотим смоделировать две функции, показанные на рисунке 1. Легко смоделировать вербально функциональные зависимости, показанные на рисунке 1. Обе модели будут содержать как минимум три правила IF-THEN. Использование меньшего количества правил снизит точность аппроксимации, а использование большего количества правил увеличит точность за счет большего требуемого времени вычислений. Это классическая дилемма мягких вычислений - компромисс между неточностью и неопределенностью, с одной стороны, и низкой стоимостью решения, управляемостью и надежностью, с другой. Соответствующие правила для функций на рисунке 1 следующие:

Левый график
IF х низкий, THEN у высокий.
IF х средний, THEN у низкий
IF х высокий, THEN у высокий

Правый график
IF х низкий, THEN у высокий.
IF х средний, THEN у средний.
IF х высокий, THEN у низкий.

Эти правила определяют три больших прямоугольных участка, закрывающих функции. Они показаны на рисунке 1 вместе с двумя возможными приближениями для каждой функции. Обратите внимание, что люди не думают (или очень редко) в терминах нелинейных функций. Мы не пытаемся «нарисовать эти функции в уме» или «увидеть» их как геометрические артефакты. Как правило, мы не обрабатываем геометрические фигуры, кривые, поверхности или гиперповерхности во время выполнения задач или выражения наших знаний.[2]

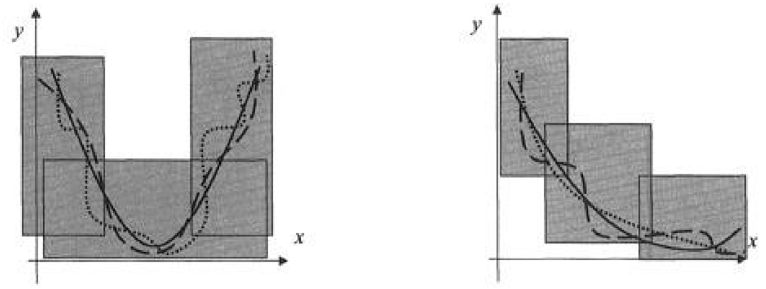


Рис.2. Две разные функции (сплошные линии на обоих графиках) покрыты тремя патчами, созданными по правилам IF-THEN и смоделированными двумя возможными приближениями (пунктирные и пунктирные кривые).

Кроме того, наш опыт или понимание некоторых функциональных зависимостей очень часто вообще не является структурированной частью знаний. Обычно мы выполняем очень сложные задачи, не имея возможности объяснить, как мы их выполняем. Любопытный читатель должен попытаться, например, объяснить коллеге в виде правил IF-THEN, как он ездит на велосипеде, распознает числа или занимается серфингом. Между знаниями или опытом и окончательной нечеткой моделью есть много шагов, как эвристических, так и математических. После завершения всех этапов проектирования и вычислений эта окончательная модель представляет собой очень точно определенную нелинейную функцию. Выбирая сложность основы правил, можно управлять точностью нечеткой модели и получать компромисс с затратами на решение. Таким образом, сначала определяются наиболее важные входные и выходные переменные для проблемы. В терминах нечеткой логики необходимо определить вселенные дискурса, то есть области и диапазоны соответствующих переменных. Затем указывается, что является низким, средним, высоким, положительным, нулевым, горячим, холодным и т. Д. В данной задаче. В терминах нечеткой логики можно определить нечеткие функции принадлежности (нечеткие подмножества или атрибуты) для выбранных входных и выходных переменных. Затем структурируют знания в виде правил IF-THEN, то есть нечетких правил (устанавливаются основы правил). Заключительный этап - выполнить числовую часть - применить некоторый алгоритм вывода (например, SUM-PROD, MAX-MIN, SUM-MIN) - и дефаззифицировать результирующие (обычно ненормальные) нечеткие подмножества. Последние два шага представляют собой четкие и точные математические операции. Мягкой частью этих вычислений является выбор функций принадлежности, а также соответствующих механизмов вывода и дефаззификации.[3] Опять же, существует компромисс между простыми и быстрыми алгоритмами, имеющими низкие вычислительные затраты и желаемую точность. Проектные решения включают количество, формы и размещение входных и выходных функций принадлежности, применяемый механизм вывода и используемый метод дефаззификации. Продемонстрируем нечеткое моделирование простого одномерного отображения $y - x2, - 3 < x < 0$. Выберите четыре нечеткие функции принадлежности (нечеткие подмножества или атрибуты) для входных и выходных переменных следующим образом:

Входные переменные	**Выходные переменные**
For -3 < x < -2, x очень отрицательный	**For 4 < x < 10, y большой**
For -3 < x < -1, x немного отрицательный	**For -3 < x < -1, y средний**
For -2 < x < 0, x почти равен нулю	**For -2 < x < 0, y маленький**

Если кто-то не удовлетворен достигнутой точностью, следует определить больше правил. Это будет достигнуто за счет более тонкой грануляции (применения меньших патчей), которая может быть реализована путем определения большего количества функций принадлежности. Нечеткое приближение, которое следует из модели с семью правилами, показано на рисунке 3. Семь нечетких функций принадлежности (нечетких подмножеств или атрибутов) для входных и выходных данных, а также соответствующая основа правил определены следующим образом:[4]

Входные переменные	**Выходные переменные**
For -3.0 < x < -2.5 x чрезвычайно далеко от нуля	**For 6.25 < x < 9, y довольно большой**
For -3.0 < x < -2.0, x очень далеко от нуля	**For 4 < x < 9, y средний**
For -2.0 < x < 1.0, x далеко от нуля	**For 0 < x < 2.25, y маленький**

Нечеткая аппроксимирующая функция, полученная в результате нечеткой аддитивной модели с семью правилами, неотличима от исходной и известной функциональной зависимости. Напомним, однако, что структурированное человеческое знание обычно находится в форме (лингвистически выраженной) основы правил, а не в форме какого-либо математического выражения. Если бы кто-то знал математическое выражение, не было бы необходимости в разработке нечеткой модели.

Литература

1. Современные Цифровые Технологии В Образовании// Гончарова А.И., Ожерельева М.В. Брянский государственный технический университет,26-27 ноября, 2020
2. Fuzzy Logic, Soft Computing& Computational Intelligence: Eleventh International Fuzzy Systems Association World Congress//Ed. Y. Liu, G. Chen, M. Ying, ISBN 7-302-11377-7/TP-7494, Springer, 2005.
3. Theoretical Advances and Applications of Fuzzy Logic and Soft Computing // Eds O. Castillo, P. Melin, O. Montiel Ross, R. Cruz, W. Pedrycz, J. Kacprzyk, Avances in Soft Computing 42, Springer, 2007.
4. Zadeh, Lotfi A. (1965). Fuzzy sets. Information and Control 8, 378-53.

Д.В. Майков

ОСТРОВНОЙ КВАНТОВЫЙ АЛГОРИТМ ПРЕСНОВОДНЫХ ГИДР ДЛЯ РЕШЕНИЯ ЗАДАЧИ ОПТИМИЗАЦИИ

(Ижевский торгово-экономический техникум)

В некоторых задачах науки и техники необходимо выполнять поиск точки глобального экстремума функции многих переменных. В качестве примера подобных задач можно привести обучение нейронных сетей [1], построение оптимального управления для систем дифференциальных уравнений [2], оптимизация параметров технических систем и др.

При решении задачи условной многомерной оптимизации необходимо найти точку глобального экстремума (например, точку минимума) заданной целевой функции $f(\mathbf{x})$ внутри n-мерной области D: