Работая с МНК будет готовы:

1. МНК не всегда уменьшит ошибку на данных KEEL.
2. МНК на больших матрицах может не изменить консеквенты
3. МНК работает долго, через него не стоит прогонять все базы правил, ограничьтесь несколькими лучшими на итерации.
4. Не пытайтесь оптимизировать туже базу правил над которой работает МНК.

Для работы с МНК в своем методе:

1. Добавьте в свой проект, ссылку на проект DevariationsMethods
2. Как и любой метод оптимизации МНК требует всю нечеткую систему, нулевой базой правил должна быть та, которую он оптимизирует.

Это выливается примерно в такой код:

public a\_Fuzzy\_System TuneUpFuzzySystem(a\_Fuzzy\_System Approx, ILearnAlgorithmConf conf)

{

int best; // индекс лучше базы правил ей и будем оптимизировать консеквенты. То есть лучшая база правил находиться в Approx.Rulles\_Database\_Set[best];

//вы что то делает

double best\_before = Approx.approx\_Learn\_Samples(best);//Запомните ошибку до оптимизации

Knowlege\_base\_ARules zeroSolution = Approx.Rulles\_Database\_Set[0]; // Сохраните бывшую нулевую базу правил куда-нибудь чтобы не исчезла.

Approx.Rulles\_Database\_Set[0]= Approx.Rulles\_Database\_Set[best]; //запихиваем свою лучшую базу правил в нулевой элемент. Т.к. МНК оптимизирует только нулевую базу правил;

Knowlege\_base\_ARules CloneBestSolution = new Knowlege\_base\_ARules( Approx.Rulles\_Database\_Set[best]); // Склонируем лучшую базу правил если вдруг после МНК ошибка увеличиться.

Adaptive\_LSM LSM = new Adaptive\_LSM(); // Создаем алгоритм МНК.

Approx =LSM.TuneUpFuzzySystem(Approx, new Null\_conf\_for\_all()); //Запускаем МНК на нечеткой системе. Чтобы не было ошибок проверьте есть ли ссылка в проекте на FuzzyCore и должен быть using Fuzzy\_system.Fuzzy\_Abstract; Нужно чтобы передать МНК пустые параметры.

double best\_after = Approx.approx\_Learn\_Samples(0); // Вычисляем ошибку после МНК.

if (best\_after> best\_before) // все плохо МНК сделал хуже

{

Approx.Rulles\_Database\_Set[0]= CloneBestSolution; // поэтому вспомнинаем клона

}

Approx.Rulles\_Database\_Set[best] = Approx.Rulles\_Database\_Set[0];// Ставим лучшую на свое законное место;

Approx.Rulles\_Database\_Set[0] = zeroSolution; // а бывшую нулевую на свое;

// вы что-то делаете

}