Какими средствами наиболее удобно пользоваться при прогнозировании страховых взносов

В.В. Колмыков

Мордовский государственный университет имени Н.П.Огарева,

e-mail: v.v_k@bk.ru

В данной статье мне бы хотелось рассмотреть, возможно ли осуществить предсказание страховых взносов средствами 1С:предприятия или же лучше воспользоваться другими средствами например нейронными сетями. Под страховым взносом будем понимать продажи.

Механизм анализа данных и прогнозирования в системе 1С:предприятие позволяет реализовать в прикладных решениях различные средства для выявления закономерностей, которые обычно скрываются за большими объемами информации.

В механизме анализа данных и прогнозирования реализовано несколько типов анализа данных.

Общая статистика

Тип анализа общая статистика предназначен для сбора информации о данных. Этот тип анализа необходим для предварительного исследования анализируемой информации.

Анализ показывает ряд характеристик дискретных и непрерывных полей. При выводе отчета в табличный документ заполняются круговые диаграммы для отображения состава полей.

Анализ данных «*Общая статистика*» не используется для прогнозирования.

Поиск ассоциаций.

Данный тип анализа осуществляет поиск часто встречаемых вместе групп объектов или значений характеристик, а также производит поиск правил ассоциаций.

На мой взгляд, данный тип анализа неудобен, поскольку собирает часто встречаемые вместе группы объектов. А данные, которые будут использоваться при прогнозировании, изначально будут разбиты на группы, например по виду страхования.

Поиск последовательностей

Поиск последовательностей позволяет выявлять в источнике данных последовательные цепочки событий.

Этот тип анализа позволяет осуществлять поиск по иерархии, что дает возможность отслеживать не только последовательности конкретных событий, но и последовательности родительских групп.

Он может использоваться тогда, когда одним из важных анализируемых показателей является последовательность продаж, которые продаются друг за другом в течение какого-либо определенного промежутка времени и т.п.

Данный тип также не целесообразно использовать в прогнозировании. Поскольку нужно отслеживать последовательность конкретных событий.

Кластерный анализ

Кластерный анализ позволяет разделить исходный набор исследуемых объектов на группы объектов таким образом, чтобы каждый объект был более схож с объектами из своей группы, чем с объектами других групп. Анализируя в дальнейшем полученные группы, называемые кластерами, можно определить, чем характеризуется та или иная группа, принять решение о методах работы с объектами различных групп.

Кластерный анализ – математическая процедура многомерного анализа, позволяющая на основе множества показателей, характеризующих ряд объектов, сгруппировать их в группы (кластеры) таким образом, чтобы объекты, входящие и один кластер, были более однородными, сходными по сравнению с объектами, входящими в другие кластеры.

В основе данного анализа лежит вычисление расстояния между объектами.

Дерево решений.

Тип анализа дерево решений позволяет построить иерархическую структуру классифицирующих правил, представленную в виде дерева.

Для построения дерева решений необходимо выбрать целевой атрибут, по которому будет строиться классификатор и ряд входных атрибутов, которые будут использоваться для создания правил.

Рассмотрев данные методы анализов можно сделать вывод, что наиболее подходящими методами являются кластеризация и дерево решений. Но метод кластеризация используется в нейронных сетях. И чаще всего он используется для распознавания образов. Средствами 1С:Предприятие можно осуществить прогнозирование но важными

На мой взгляд, для прогнозирования продаж в страховании лучше использовать нейронные сети. Поскольку нейронные сети строятся непосредственно под решаемую задачу, где учитываются все необходимые факторы. Изначально задача упрощается под сеть. И надо помнить, что лучше решить. И какая именно проблема стоит при прогнозировании.

Нейронные сети - исключительно мощный метод моделирования, позволяющий воспроизводить чрезвычайно сложные зависимости. На данный момент существует большое множество алгоритмов для обучения нейронных сетей. Это алгоритм обратного распространения, сеть встречного распространения, двунаправленная ассоциативная память,

сети хопфилда. Рассмотрим алгоритм обратного распространения, и сеть встречного распространения для применения к прогнозированию финансовых рядов.

Алгоритм обратного распространения.

В алгоритме *обратного распространения* вычисляется вектор градиента поверхности ошибок. Этот вектор указывает направление кратчайшего спуска по поверхности из данной точки, поэтому если мы "немного" продвинемся по нему, ошибка уменьшится. Последовательность таких шагов в конце концов приведет к минимуму того или иного типа. Определенную трудность здесь представляет вопрос о том, какую нужно брать длину шагов.

При большой длине шага сходимость будет более быстрой, но имеется опасность перепрыгнуть через решение или уйти в неправильном направлении. Выбор же скорости обучения зависит от конкретной задачи и обычно осуществляется опытным путем; эта константа может также зависеть от времени, уменьшаясь по мере продвижения алгоритма.

Формула обратного распространения (back-propagation formu1a) для локального градиента $\delta_j(n)$ скрытого нейрона j:

$$d_j(n) = j'_j(u_j(n))\sum_k d_k(n)w_{kj}(n) \qquad (1)$$

Множитель $\varphi'_j(\upsilon_j(n))$, использованный в формуле для вычисления локального градиента $\delta_j(n)$, зависит исключительно от функции активации, связанной со скрытым нейроном j. Второй множитель формулы (сумма по k) зависит от двух множеств параметров. Первое из них – $\delta_k(n)$ – требует знания сигналов ошибки $e_k(n)$. Сигнал ошибки выходного нейрона j на итерации n (соответствующей n-му примеру обучения) определяется соотношением

$$e_j(n) = d_j(n) - y_j(n) \qquad (2)$$

Для всех нейронов слоя, находящегося правее скрытого нейрона j, которые непосредственно связаны с нейроном j. Второе множество – $w_{kj}(n)$ – состоит из синаптических весов этих связей.

Теперь можно свести воедино все соотношения для алгоритма обратного распространения. Во-первых, коррекция $\Delta w_{kj}(n)$, применяемая к синаптическому весу, соединяющему нейроны i и j, определяется следующим дельта-правилам:

$$\begin{pmatrix} Коррекция \\ веса \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} Параметр\,ско- \\ рости\,обучения \\ h \end{pmatrix} \begin{pmatrix} Локальный \\ градиент \\ d_j(n) \end{pmatrix} \begin{pmatrix} Входной\,сиг- \\ нал\,нейрона\ j \\ y_j(n) \end{pmatrix} \quad (3)$$

Во-вторых, значение локального градиента $\delta_j(n)$ зависит от положения нейрона в сети.

1. Если нейрон j выходной, то градиент $\delta_j(n)$ равен произведению производной $\varphi_j'(\upsilon_j(n))$ на сигнал ошибки $e_j(n)$ для нейрона j (см. выражение (4)).

$$d_j(n) = -\frac{\partial E(n)}{\partial u_j(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial \upsilon_j(n)} = e_j(n) j_j'(u_j(n)) \qquad (4)$$

2. Если нейрон j – скрытый, то градиент $\delta_j(n)$ равен произведению производной $\varphi_j'(\upsilon_j(n))$ на взвешенную сумму градиентов, вычисленных для нейронов следующего скрытого или выходного слоя, которые непосредственно связаны с данным нейроном j (см. выражение (5)). [2]

$$d_j(n) = j_j'(u_j(n)) \sum_k d_k(n) w_{kj}(n) \qquad (5)$$

Сеть встречного распространения

Данная сеть имеет два слоя с последовательными связями.

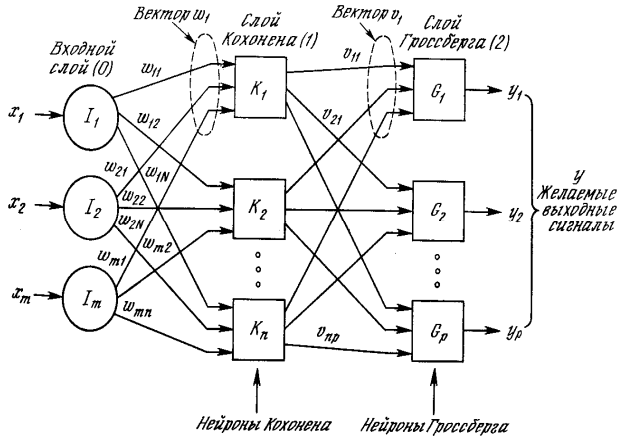

Рисунок 1 – Сеть с встречного распространения

Слои Кохоненна

В своей простейшей форме слой Кохонена функционирует в духе «победитель забирает все», т. е. для данного входного вектора один и только один нейрон Кохонена выдает на выходе логическую единицу, все остальные выдают ноль. Нейроны Кохонена можно воспринимать как набор электрических лампочек, так что для любого входного вектора загорается одна из них.

Ассоциированное с каждым нейроном Кохонена множество весов соединяет его с каждым входом. Например, на рисунке 1 нейрон Кохонена K_1 имеет веса w_{11}, w_{21}, …, w_{m1}, составляющие весовой вектор $\mathbf{W}_1$. Они соединяются через входной слой с входными сигналами x_1, x_2, …, x_m, составляющими входной вектор **X**. Подобно нейронам большинства сетей выход NET каждого нейрона Кохонена является просто суммой взвешенных входов. Это может быть выражено следующим образом:

$$\mathrm{NET}_j = w_{1j}x_1 + w_{2j}x_2 + \ldots + w_{mj}x_m \qquad (6)$$

где NET_j – это выход NET нейрона Кохонена *j*,

$$\mathrm{NET}_j = \sum_i x_i w_{ij} \quad (7)$$

или в векторной записи

$$\mathbf{N} = \mathbf{XW}, \quad (8)$$

где **N** – вектор выходов NET слоя Кохонена.

Нейрон Кохонена с максимальным значением NET является «победителем». Его выход равен единице, у остальных он равен нулю.

Слой Гроссберга

Слой Гроссберга функционирует в сходной манере. Его выход NET является взвешенной суммой выходов k_1,k_2, ..., k_n слоя Кохонена, образующих вектор **К**. Вектор соединяющих весов, обозначенный через **V**, состоит из весов v_{11}, v_{21}, ..., v_{np}. Тогда выход NET каждого нейрона Гроссберга есть

$$\mathrm{NET}_j = \sum_i k_i w_{ij}, \quad (9)$$

где NET_j – выход *j*-го нейрона Гроссберга, или в векторной форме

$$\mathrm{Y} = \mathrm{KV}, \quad (10)$$

где **Y** – выходной вектор слоя Гроссберга, **К** – выходной вектор слоя Кохонена, **V** – матрица весов слоя Гроссберга.

Если слой Кохонена функционирует таким образом, что лишь у одного нейрона величина NET равна единице, а у остальных равна нулю, то лишь один элемент вектора **К** отличен от нуля, и вычисления очень просты. Фактически каждый нейрон слоя Гроссберга лишь выдает величину веса, который связывает этот нейрон с единственным ненулевым нейроном Кохонена.

Обучение слоя Гроссберга – это обучение с учителем, алгоритм использует заданные желаемые выходы.

В полной модели сети встречного распространения имеется возможность получать выходные сигналы по входным и наоборот. Этим двум действиям соответствуют прямое и обратное распространение сигналов.[1]

Основная область применения распознавание образов, восстановление образов, сжатие данных.

Исходя из выше сказано, что для применения нейронных сетей в страховании наиболее предпочтительный вариант использовать алгоритм обратного распространения.

Основная специфика предсказания лежит в области предобработки данных. Процедура обучения отдельных нейросетей стандартна. Как всегда, имеющиеся примеры разбиваются на три выборки: обучающая, валидационная и тестовая. Первая используется для обучения, вторая - для выбора оптимальной архитектуры сети и/или для выбора момента остановки обучения. Наконец, третья, которая вообще не использовалась в обучении, служит для контроля качества прогноза обученной нейросети.

Однако для сильно зашумленных финансовых рядов существенный выигрыш в надежности предсказаний способно дать использование комитетов сетей

В литературе имеются свидетельства улучшения качества предсказаний за счет использования нейросетей с обратными связями. Такие сети могут обладать локальной памятью, сохраняющей информацию о более далеком прошлом, чем то, что в явном виде присутствует во входах.

Нейронные сети имеют свои как недостатки также как и преимущества. Основными преимуществами являются удобные способы модифицировать модель по мере того как появляются новые наблюдения. Модель хорошо работает с временными последовательностями, в которых мал интервал наблюдений, т.е. может быть получена относительно длительная временная последовательность. Что благоприятствует для прогнозирования страховых продаж.

Список использованной литературы.

1. Ф., Уоссермен. Нейрокомпьютерная техника: Теория и практика 118 с.
2. С., Хайкин. Перевод с английского. д.т.н. н.н. Кусеуль, к.т.н. А.Ю. Шелестова, под редакцией д.т.н. н.н. Кусеуль. Нейронные сети: полный курс, 2-е издание. : Пер. с англ. М. Издательский дом "Вильямс", 2006. 1104 с. : ил. Парал. тит. англ.