Корректная селекция алгоритмов классификации

Хомич А.В.
BaseGroup Labs
ho76@mail.ru; supremum76@rambler.ru

Аннотация

Рассматривается проблема определения доверительных интервалов надежности классифицирующих алгоритмов. Показано, что при селекции классификаторов по результатам тестирования необходимо учитывать количество тестируемых классификаторов. Продемонстрировано смещение оценок надежности при селекции классификаторов по результатам теста. Предлагается отказаться от селекции только по результатам тестирования и ввести ограничение на количество параметров в алгоритмах классификации. Предложен метод поиска классификатора, не приводящий к увеличению длины доверительных интервалов и смещению оценок надежности. Часть полученных результатов может быть применена к задаче регрессии.

1. Введение

В настоящей работе рассматривается следующая задача. Пусть имеется конечный набор векторов. Каждому вектору неким «учителем» сопоставлен его класс. В качестве «учителя» может выступать эксперт, измерительный прибор и т. п. Вектор с указанным классом называют примером. Набор имеющихся примеров является выборкой из генеральной совокупности всех возможных в данной предметной области примеров. Требуется создать классификатор, правильно классифицирующий векторы, не участвующие в обучении, основываясь только на анализе имеющихся примеров. Под надежностью классификатора будем понимать вероятность правильной классификации на генеральной совокупности примеров. Все подходы к решению задачи получения надежного классификатора условно можно условно разделить на два случая:

- Подход, основанный на явном использовании сведений о распределении классов.
- Подход, основанный на представлении классификатора в виде «черного ящика» [1].

Первый подход позволяет провести статистически обоснованный анализ надежности классификатора. Но он применим только в относительно простых задачах, в которых наблюдаются хорошо изученные распределения классов и зависимости восстанавливаются в достаточно узком классе функций. Второй подход, напротив, позволяет решать сложные нелинейные задачи, восстанавливая зависимости широкого класса, но вызывает затруднение оценка надежности классификатора. В методе «черного ящика» при общепринятом подходе все примеры разделяются на непересекающиеся наборы обучающих и тестовых примеров. Параметры «черного ящика» настраиваются на правильную классификацию обучающих примеров. На тестовых примерах оценивается его надежность.

Дальнейшие рассуждения основываются на методе определения доверительного интервала надежности классификатора, предложенного в работе [2]. Новым результатом является то, что в данной работе продемонстрирована зависимость доверительного интервала от количества тестируемых вариантов классификатора (используется метод «черного ящика»).

2. Случай тестирования одного алгоритма классификации

Введем обозначения:

- P – значение некоторой вероятности;
- $\Pr(A)$ – безусловная вероятность наступления события A;
- $\Pr(A \mid B)$ – вероятность наступления события A при условии B;
- P_{est} - оценка на тестовых примерах вероятности правильной классификации (оценка надежности);
- $\forall P_{est}$ - читается, как «Все P_{est}, вычисленные в серии тестов, удовлетворяют условию…»;
- $\exists P_{est}$ - читается, как «Среди вычисленных оценок надежности найдется оценка равная P_{est}» или «Среди вычисленных оценок надежности найдется оценка P_{est}, удовлетворяющая условию…».

Пусть m – число бинарных позиций, принимающих значение 0 или 1; k – число позиций, принявших единичное значение ($m \geq k$). Число возможных вариантов размещения k единиц среди m бинарных позиций [3] равно

$$N(m,k) = C_m^k = \frac{m!}{(m-k)!\,k!}. \qquad (1)$$

Вероятность k из m правильных ответов при вероятности правильного ответа P_{true} определяется выражением [2]

$$\Pr(P_{est} \mid P_{true}, m) = N(m,k) P_{true}^k (1 - P_{true})^{m-k}, \qquad (2)$$

где $P_{est} = \frac{k}{m}$.

Оценка P_{est} может оказаться как больше, так и меньше действительной надежности классификации P_{true}. Вероятность надежности классификации P_{true} при полученной оценке P_{est} по результатам m тестов можно выразить, используя формулу Байеса [4,5], как

$$\Pr\left(P_{true} \mid P_{est}, m\right) = \frac{\Pr\left(P_{est} \mid P_{true}, m\right)}{\sum_{P_{true} \in \Omega} \Pr\left(P_{est} \mid P_{true}, m\right)}, \quad (3)$$

где Ω – множество всех возможных значений P_{true}. Если количество возможных значений позиций вектора примера (факторов) конечно, то Ω конечно. В противном случае Ω должна быть заменена конечным множеством. Этого можно достичь, применив следующий прием. Предположим наличие некоторого большого, но конечного количества примеров и примем, что вероятность правильной классификации P_{true} приближенно равна отношению количества правильно решенных примеров к общему количеству примеров. Пусть количество примеров равно некоторому достаточно большому числу M. Тогда вместо $P_{true} \in [\alpha, \beta]$ можно записать

$$P_{true} \in \left\{\alpha, \frac{\alpha M + 1}{M}, \frac{\alpha M + 2}{M}, \cdots, \frac{\beta M - 1}{M}, \beta\right\}.$$

Значение M подбирают, соблюдая компромисс между точностью и сложностью вычислений. Будем обозначать множество, полученное таким способом, как $\Omega[\alpha, \beta]$.

Для более полного анализа надежности классификатора необходимо определить доверительный интервал P_{true}. Обозначим нижнюю границу доверительного интервала P_{true} как P_{below}, а верхнюю границу как P_{above} или $P_{below} \le P_{true} \le P_{above}$. Доверительные интервалы P_{true} можно определить по формуле

$$\Pr\left(P_{true} \in [P_{below}, P_{above}] \mid P_{est}, m\right) = \sum_{P_{true} \in \Omega[P_{below}, P_{above}]} \Pr\left(P_{true} \mid P_{est}, m\right) = \alpha, \quad (4)$$

где α – доверительная вероятность. При необходимости определить P_{above} и P_{below} можно воспользоваться формулой (4). Например, задавшись определенными значениями P_{est}, m, α, и перебрав различные варианты $[P_{below}, P_{above}]$ можно выбрать доверительный интервал наименьшей длины (интервал с наименьшим из рассмотренных вариантов значением $P_{above} - P_{below}$).

Наибольший интерес в большинстве приложений вызывает нижняя граница доверительного интервала. Она показывает, какую минимальную надежность классификации можно гарантировать. В таблицах 1 и 2 приведены значения P_{below}, вычисленные по формуле (4) (при расчетах использовалось значение $M = 100$).

Таблица 1 – Значения нижней границы доверительного интервала при $\alpha = 0.95$

P_{est} \ M	10	20	30	40	50
0.6	0.33	0.40	0.43	0.45	0.46
0.7	0.43	0.50	0.54	0.56	0.57
0.8	0.53	0.61	0.65	0.67	0.68
0.9	0.63	0.72	0.76	0.79	0.80
1.0	0.76	0.86	0.90	0.93	0.94

Таблица 2 – Значения нижней границы доверительного интервала при $\alpha = 0.9$

P_{est} \ M	10	20	30	40	50
0.6	0.37	0.43	0.45	0.47	0.48
0.7	0.48	0.53	0.56	0.58	0.59
0.8	0.58	0.65	0.68	0.69	0.70
0.9	0.69	0.76	0.79	0.81	0.82
1.0	0.81	0.89	0.92	0.94	0.95

3. Случай тестирования множества алгоритмов классификации

Для получения классификатора с удовлетворительной надежностью классификации чаще всего приходиться тестировать набор различных вариантов классификаторов. Перебор вариантов и их оценка может осуществляться автоматически. Исследователь тестирует набор различных классификаторов и, основываясь на результатах тестирования, выбирает самый надежный. Как и выше, считаем, что оценка надежности при тестировании определенного классификатора равна отношению числу правильных результатов классификации к общему числу тестов для данного классификатора $P_{est} = \frac{k}{m}$. Покажем, что в этом случае проблема расхождения оценки и истинной надежности классификации становится особенно острой.

Пусть *n* различных классификаторов тестируются на *m* примерах. Обычно значения надежности классификаторов различны. Под ошибочным выбором классификатора будем понимать выбор классификатора с наименьшей ошибкой тестирования, но не наименьшей реальной ошибкой обобщения. Вероятность ошибочного выбора классификатора зависит от сочетания значений надежности тестируемых классификаторов. Наилучшим случаем является ситуация, когда часть тестируемых классификаторов реально обладают нулевой вероятностью ошибки классификации, а оставшаяся часть обладает единичной вероятностью ошибки классификации. В этом случае ошибка тестирования будет полностью соответствовать реальной ошибке классификации. Следовательно, ошибка тестирования позволит однозначно выявить лучшие и худшие классификаторы.

Наихудшим случаем является ситуация, когда истинная надежность всех тестируемых классификаторов одинакова и недостаточна для практического приложения. В этом случае разброс значений ошибок тестирования отражает лишь наличие неопределенности в оценке ошибки классификации. Случайно полученное малое значение ошибки тестирования может быть ошибочно расценено как признак достаточно малой для практического приложения реальной ошибки классификации. Трудно оценить сочетание реальных значений надежности тестируемых классификаторов без привлечения дополнительных сведений. В отсутствии дополнительных сведений приходиться исходить из наихудшего случая.

Допустим, истинная надежность всех классификаторов одинакова и равна P_{true}. Это предположение соответствует наихудшему случаю и используется во всех дальнейших рассуждениях. Если заменить его более слабым ограничением на равенство средней надежности классификаторов некоторому значению, то очевидно с максимальной вероятностью будет выбран классификатор с максимальной надежностью, и его надежность будет не ниже средней надежности по всем классификаторам. Только при условии равенства реальной надежности классификаторов максимальное и среднее значения надежностей совпадают.

Пусть все классификаторы тестируются независимо друг от друга и пусть исследователем задано значение удовлетворительной оценки надежности P_{well}. Вероятность случайно получить, в ходе исследования, удовлетворительную оценку надежности P_{well} определяется по формуле (2), то есть

$$\Pr\left(P_{est} = P_{well} \mid P_{true}, m\right) = \Pr\left(P_{well} \mid P_{true}, m\right). \tag{5}$$

Но надо учитывать и случаи, когда получена оценка больше P_{well}. То есть надо учитывать все случаи, когда может быть получено правильных ответов $mP_{well}, mP_{well}+1, mP_{well}+2, \ldots, m$. Согласно формуле, описывающей наступление любого несовместного события [4,5], вероятность получения оценки, не меньшей P_{well}, определяется выражением:

$$\Pr\left(P_{est} \geq P_{well} \mid P_{true}, m\right) = \sum_{i=0}^{m\left(1-P_{well}\right)} \Pr\left(P_{est} = \frac{mP_{well}+i}{m} \mid P_{true}, m\right). \tag{6}$$

Вероятность того, что такая оценка при *n* испытаниях ни разу не будет получена, подчиняется закону совместного наступления независимых событий

$$\Pr\left(\forall P_{est} < P_{well} \mid P_{true}, n, m\right) = \left(1 - \Pr\left(P_{est} \geq P_{well} \mid P_{true}, m\right)\right)^n. \tag{7}$$

Вероятность того, что хотя бы одному классификатору удастся при тестировании показать оценку надежности не меньше P_{well}, равна

$$P_{\text{deception}} = \Pr\left(\exists P_{est} \geq P_{well} \mid P_{true}, n, m\right) = \\ = 1 - \left(1 - \Pr\left(P_{est} \geq P_{well} \mid P_{true}, m\right)\right)^n \quad . \tag{8}$$

В случае, если $P_{true} < P_{well}$ величина $P_{\text{deception}}$ характеризует вероятность получить завышенную оценку надежности.

Для демонстрации значимости проблемы завышения оценки надежности классификаторов приведем следующий пример. Пусть тестируется 1000 классификаторов на 100 тестовых примерах. Пусть все классификаторы обладают надежностью 0.5. Вероятность получить оценку надежности, не меньше 0.65, при однократном тестировании приближенно равна 0.0015. Вероятность получить оценку надежности не меньше 0.65, хотя бы при одном испытании из 1000, приближенно составляет 0.79. Как видно, вероятность получения существенно разнящейся оценки и истинной надежности при многократных испытаниях может быть большой даже при малой вероятности при однократном испытании.

Напомним, было принято ограничение, что реальная надежность всех тестируемых классификаторов равна P_{true}. Вероятность того, что в n тестах на m примерах только одна оценка надежности классификации будет равна P_{est}, определяется как

$$n \Pr\left(P_{est} \mid P_{true}, m\right)\left(1 - \Pr\left(P_{est} \mid P_{true}, m\right)\right)^{n-1}.$$

Вероятность того, что в n тестах на m примерах ровно две оценки надежности классификации будут равны P_{est}, определяется, как

$$N\left(n,2\right)\Pr\left(P_{est} \mid P_{true}, m\right)^2\left(1 - \Pr\left(P_{est} \mid P_{true}, m\right)\right)^{n-2}.$$

Рассуждая далее аналогично, получим ряд несовместных событий. Для того чтобы получить вероятность того, что в n тестах на m примерах хотя бы одна оценка надежности классификации будет равна P_{est}, достаточно сложить вероятности этих событий или

$$\Pr\left(\exists P_{est} \mid P_{true}, n, m\right) = \\ = \sum_{i=1}^{n} N\left(n, i\right)\Pr\left(P_{est} \mid P_{true}, m\right)^i\left(1 - \Pr\left(P_{est} \mid P_{true}, m\right)\right)^{n-i} \quad . \tag{9}$$

Имея формулу вероятности получения конкретного значения оценки надежности, можем определить вероятность истинной надежности. Вероятность надежности классификации P_{true} при полученной оценке P_{est} по результатам тестирования n классификаторов на m тестах можно выразить, используя формулу Байеса [4,5]

$$\Pr(B_i \mid A) = \frac{P(B_i)P(A \mid B_i)}{\sum_{j \in [1,i) \cup (i,k]} P(B_j)P(A \mid B_j)},$$

где k – количество учитываемых несовместных событий, влияющих на вероятность реализации события A. Задача состоит в получении формулы для определения $\Pr(P_{true} \mid P_{est}, n, m)$. Формально для этого необходимо знать все условные вероятности $\Pr(P_{est} \mid P_{true}, n, m)$ и безусловные вероятности $\Pr(P_{est})$. Условие несовместности событий выполняется, так как, очевидно, что в одном тесте невозможно одновременно получить две разные ошибки тестирования. $\Pr(P_{est} \mid P_{true}, n, m)$ интерпретируется, как условная вероятность получить при тестировании ошибку тестирования, равную P_{est}. Исходя из этой интерпретации, $\Pr(P_{est} \mid P_{true}, n, m)$ можно заменить на $\Pr(\exists P_{est} \mid P_{true}, n, m)$. $\Pr(P_{est})$ интерпретируется, как вероятность получения при произвольном тестировании произвольного классификатора ошибку тестирования, равную P_{est}. Сложность определения $\Pr(P_{est})$ заключается в том, что ни тестовый набор примеров, ни тестируемый классификатор явно ничем не ограничен. Конечно, можно рассмотреть все множество возможных пар <тестовых примеры, тестируемый классификатор>. Но не факт, что на практике каждая такая пара имеет одинаковую вероятность реализации. Примем как гипотезу, что $\Pr(P_{est})$ равномерно распределено на отрезке [0,1]. В этом случае $\Pr(P_{est})$ в формуле Байеса сокращается.

$$\Pr(P_{true} \mid P_{est}, n, m) = \frac{\Pr(\exists P_{est} \mid P_{true}, n, m)}{\sum_{P_{true} \in \Omega[0,1]} \Pr(\exists P_{est} \mid P_{true}, n, m)}. \qquad (10)$$

Формула (10) выводится аналогично формуле (3) [2], но с учетом множественности проводимых тестов и применения процедуры выбора классификатора с максимальной оценкой P_{est}.

Доверительные интервалы P_{true} в случае тестирования n классификаторов на m тестах можно определить по формуле

$$\Pr(P_{true} \in [P_{below}, P_{above}] \mid P_{est}, n, m) = \\ = \sum_{P_{true} \in \Omega[P_{below}, P_{above}]} \Pr(P_{true} \mid P_{est}, n, m) = \alpha, \qquad (11)$$

где α – доверительная вероятность, $\Omega[P_{below}, P_{above}]$ – конечное перечислимое множество, определяемое так же, как и в формуле (4).

На практике формулу (11) можно использовать для вычисления довери-

тельных интервалов P_{true}. Например, задавшись определенными значениями P_{est}, m, n, α, и перебрав различные варианты $[P_{below}, P_{above}]$, можно выбрать доверительный интервал наименьшей длины (интервал с наименьшим из рассмотренных вариантов значением $P_{above} - P_{below}$).

Представленная методика позволяет определить доверительный интервал надежности классификатора и обоснованно выбирать количество тестовых примеров. Как видно из таблиц 1 и 2, для оценки надежности классификатора рекомендуется использовать не менее 50 тестовых примеров. При использовании 50 или менее тестовых примеров следует наряду с полученной оценкой указывать и нижнюю границу доверительного интервала надежности классификатора. При тестировании набора конкурирующих классификаторов следует увеличить число тестовых примеров. Также можно дать рекомендацию по возможности избегать многократных тестов. В следующем разделе приводится алгоритм, позволяющий выбрать наиболее надежный классификатор без необходимости большого количества повторений процедуры тестирования.

4. Эффект структурного переобучения

В качестве цели оптимизации структуры и параметров классификатора часто рассматривается [6,13,15] увеличение надежности классификации. На практике используются и другие цели оптимизации структур и параметров, но в данной статье внимание сконцентрировано именно на надежности классификации. При решении задачи регрессии критерием оптимальности может служить среднеквадратичное отклонение по всей области возможных входных значений. Часто эти критерии объединяют общим термином – ошибка обобщения. Непосредственно измерить ошибку обобщения невозможно. Прибегают к оценкам, выводимым из ошибок на тестовых примерах (ошибка тестирования). Параметры алгоритма настраиваются с целью минимизации ошибки на обучающих примерах (ошибка обучения). Надежность алгоритма оценивают по ошибке тестирования. Только при неограниченном увеличении количества тестовых примеров разница между ошибкой тестирования и ошибкой обобщения стремится к нулю. При оптимизации структуры алгоритма приходится перебирать множество вариантов структур [6]. В качестве критерия селекции структур может служить ошибка тестирования.

Практика показывает [7,8,9], что селекция только по ошибке тестирования, в случае перебора большого количества алгоритмов, не гарантирует получение алгоритма с удовлетворительной ошибкой обобщения. Одной из причин этого служит возможность случайного получения малой ошибки тестирования. Если при оценивании 1-5 вариантов структур вероятность такого мала, то при переборе 100 и более структур вероятность случайного результата резко возрастает. Для устранения этого недостатка необходимо увеличивать количество тестов.

Другим источником проблем служит эффект “структурного переобучения”. Эффект “переобучения” при настройке параметров хорошо известен [10]. Эффект переобучения проявляется в том, что, начиная с некоторого порога,

уменьшение ошибки обучения сопровождается увеличением ошибки тестирования. Такое поведение обусловлено тем, что параметры классификатора начинают настраиваться на шумы и закономерности, присущие только обучающему набору примеров. При оптимизации структуры классификаторов также возникает подобный эффект. Алгоритм с выбранной структурой показывает малую ошибку тестирования, но ошибка обобщения остается высокой. Особенно явно этот эффект проявляется в условиях малого количества доступных примеров (50-200). Подобный эффект наблюдался и другими исследователями [7,8,9].

При малом количестве доступных примеров применяются алгоритмы перекрестного тестирования с множеством вариантов разбиений на обучающую и тестовую группы примеров. В исследованиях автора статьи, в которых наблюдался эффект "структурного переобучения", использовалась bootstrap-оценка [11,12] (один из вариантов метода скользящего контроля). При вычислении bootstrap-оценки генерируется множество случайных разбиений примеров на тестовую и обучающую группы. Тестовая группа содержит фиксированное количество примеров. В данной работе использовались тестовые группы из 10 примеров. Итоговая ошибка тестирования определяется суммированием ошибок тестирования по всем тестовым группам. Эффект "структурного переобучения" уже нельзя устранить одним увеличением количества тестов. В данной работе общее количество тестов доводилось до 500, но эффект продолжал проявляться. По всей видимости, эффект "структурного переобучения" является следствием наличия в выборках ограниченного размера зависимостей, свойственных только этим выборкам (не наблюдаемых на генеральной совокупности примеров). Если при обучении одного алгоритма происходит подстройка под обучающие примеры, то при селекции по результатам тестирования множества алгоритмов происходит подстройка под тестовые примеры. На первый взгляд можно из множества перебираемых алгоритмов выбирать тот, который после обучения демонстрирует наименьшую ошибку тестирования. Но вероятность ошибки на тестовых примерах вовсе не обязательно равна вероятности ошибки на произвольных примерах. При отборе может быть выбран алгоритм, который демонстрирует малую ошибку тестирования не в силу высоких способностей к обобщению, а в силу «удачного» сочетания структуры, начального приближения параметров и свойств алгоритма оптимизации параметров. Если для поиска алгоритма используются генетические алгоритмы, известные своей способностью решать сложные задачи, а поиск такого «удачного» сочетания очевидно сложная задача, то такая ситуация весьма вероятна.

Для устранения эффекта "структурного переобучения" в работе [8] предлагается выбирать среди множества конкурирующих алгоритмов, алгоритм не с самой лучшей оценкой точности. В данной статье предлагается использовать иной подход, заключающийся во введении явных ограничений на максимальное количество параметров (МКП) в алгоритме. Зависимости, свойственные только обучающей выборке, сформированы случайно, следовательно, сложнее общих зависимостей (наблюдаемых на генеральной совокупности примеров). Поэтому структура малой сложности не позволит их выявить и будет вынуждена настраивать свои параметры на общие зависимости. Аналогично устраняют эффект "параметрического переобучения", когда ограничивают количество параметров и разброс их значений [13]. В обоснование введения МКП можно также

сослаться на принцип минимальной длины описания [14].

Оптимальное значение МКП можно приближенно определять методом, предложенным в работе [15]. В этой работе предложено выбирать количество настроечных параметров равным $\sqrt{K * N}$, где K – размерность вектора примера, N – количество обучающих примеров. При необходимости, МКП можно уточнять методами одномерной оптимизации (например, методом золотого сечения). На минимальную сложность алгоритма ограничения не накладываются. Селекция алгоритмов, удовлетворяющих требованию МКП, выполняется по ошибке обучения. После завершения оптимизации структуры и параметров алгоритма выполняется проверка на тестовом наборе примеров. Запоминается алгоритм с наименьшей ошибкой тестирования. Затем можно скорректировать МКП и повторить поиск. При оптимизации МКП количество пробуемых вариантов следует ограничивать 3-10 значениями. Во-первых, вычисление критерия оптимальности требует много вычислительных затрат. Во-вторых, при увеличении числа проб растет вероятность случайно получить хороший результат.

Опишем использованный в данной работе алгоритм поиска надежных алгоритмов классификации или регрессии. В процессе поиска ищется алгоритм ALG с наименьшей ошибкой обобщения. Количество итераций алгоритма ограничено величиной T, задаваемой пользователем.

1. $t := 1$;
2. Выбирается МКП$_t$;
3. Создается множество A_t алгоритмов с количеством свободных параметров в каждом, не превышающим МКП$_t$;
4. Каждому алгоритму из A_t назначаются значения параметров, доставляющие минимум ошибки обучения (эмпирической ошибки);
5. Из A_t выбирается алгоритм ALG_t с наименьшей ошибкой обучения (эмпирической ошибкой);
6. Для алгоритма ALG_t вычисляется ошибка тестирования (оценка ошибки обобщения) $E(ALG_t)$;
7. $t := t + 1$;
8. Если $t > T$, то переход ша шаг 9, иначе переход на шаг 2;
9. $ALG = \arg\min_{ALG_t} E\left(ALG_t\right)$, $t = 1,\ldots,T$.

Описанный алгоритм не ограничивает методы создания множества алгоритмов, настройки параметров алгоритмов, разделения примеров на обучающие и тестовые наборы, выбора МКП. Алгоритм служит для поиска других алгоритмов с заданными характеристиками. С учетом всего этого представленный алгоритм может считаться метаалгоритмом.

Для проверки гипотезы наличия эффекта структурного переобучения был выполнен ряд экспериментов по восстановлению модельных функций (таб. 3). Результаты модельных экспериментов приведены в таблице 4. Эксперименты проводились с помощью симулятора искусственных нейронных сетей NeuroGenesis [16]. В каждом эксперименте создавалось 1000 нейронных сетей с различной структурой. На структуры нейронных сетей накладывались только общие ограничения. Ограничивалось количество скрытых слоев. Разрешались структуры нейронных сетей содержащие от 1 до 3 скрытых слоев. В части экс-

периментов ограничивалось максимальное суммарное количество синапсов в нейронной сети. Количество нейронов явно не ограничивалось. Каждый вариант структуры нейронной сети оценивался перекрестным тестированием (bootstrap-оценка). В качестве решения выбиралась нейронная сеть с наименьшей ошибкой тестирования. Затем нейронная сеть тестировалась на дополнительном наборе из 1000 примеров. Вероятность ошибки на дополнительных примерах принималась за приближенное значение вероятности ошибки на произвольных примерах.

Таблица 3. Модельные функции

Обозначение	Определение
F_1	$F_1(x_1, x_2, x_3, x_4) = \begin{cases} 1, \text{если } x_1 x_2 > x_3 x_4 \\ 0, \text{в противном случае} \end{cases}$
F_2	$F_2(x_1, x_2, x_3, x_4) = \begin{cases} 1, \text{если } \max\{x_1, x_2\} > \max\{x_3, x_4\} \\ 0, \text{в противном случае} \end{cases}$
F_3	$F_3(x_1, x_2, x_3, x_4, x_5, x_6) = \begin{cases} sign(x_5), \text{если} \\ \min\{x_1, x_2, x_1 x_2\} > \min\{x_3, x_4, x_3 x_4\} \\ sign(x_6), \text{в противном случае} \end{cases}$ $sign(x) = \begin{cases} 1, \text{если } x > 0 \\ 0, \text{если } x \le 0 \end{cases}$
F_4	$F_4(x_1, x_2, x_3, x_4) = \begin{cases} sign(x_1 x_2), \text{если } x_1 x_2 x_3 x_4 > 0 \\ sign(x_3 x_4), \text{в противном случае} \end{cases}$

Таблица 4. Результаты экспериментов по восстановлению модельных функций

Восстанавливаемая функция	Кол-во примеров	Отсутствие ограничений на кол-во синапсов		Наличие ограничений на максимальное кол-во синапсов	
		Минимальная ошибка тестирования	Ошибка на произвольных примерах	Минимальная ошибка тестирования	Ошибка на произвольных примерах
F_1	200	19%	60%	30%	32%
F_2	300	18%	43%	19%	19%
F_3	500	22%	53%	32%	33%
F_4	500	21%	54%	22%	26%

Из таблицы 4 видно, что оптимизация структур нейронных сетей в отсутствии ограничений на максимальное количество синапсов приводит к большому

отличию вероятности ошибки на тестовых примерах от вероятности ошибки на произвольных примерах. При введении ограничений на максимальное количество параметров вероятность ошибки на тестовых примерах намного точнее приближает вероятность ошибки на произвольных примерах. Эффект "структурного переобучения" ярко выражен во всех 4-х модельных экспериментах.

5. Заключение

Выбор классификатора из множества вариантов по результатам тестирования ведет к увеличению длины доверительных интервалов надежности. В статье показаны комбинаторно-вероятностные причины данного явления. Также установлено наличие эффекта подстройки под тестовые примеры, что приводит к занижению оценок ошибки обобщения. Все это затрудняет селекцию алгоритмов классификации или регрессии по результатам тестирования. В качестве альтернативы предлагается ввести дополнительный уровень настройки алгоритма поиска классификатора или регрессора. Предлагается метод максимизации надежности алгоритмов классификации и регрессии, использующий ограничения на количество настроечных параметров в алгоритмах. В предложенном методе поиск ведется среди алгоритмов, удовлетворяющих ограничению числа параметров, селекция выполняется по ошибке обучения, по окончании поиска выполняется тестирование выбранного алгоритма. При необходимости ограничение на максимальное количество параметров меняется, и поиск повторяется снова. При таком подходе можно ограничиться вычислением 3-10 оценок ошибки обобщения по ошибке тестирования, что практически не приведет к увеличению длин доверительных интервалов или смещению оценок.

Литература

1. Эшби У.Р. Введение в кибернетику —М.: УРСС, 2005. — 432 с. —ISBN: 5-484-00031-9
2. Highlayman W. H. The design and analysis of pattern recognition experiments // Bell System Technical Journal. — 1962. — vol. 41. — p. 723—744.
3. Нефедов В. Н., Осипова В. А. Курс дискретной математики — М.: МАИ, 1992. — 263 с.
4. Гмурман В. Е. Теория вероятностей и математическая статистика — М.: Высш. шк., 2003. — 479 с.
5. Кендалл М., Стьюарт А. Статистические выводы и связи — М.: Наука, 1973. — 900 с.
6. Хомич, А. В., Жуков Л. А. Метод эволюционной оптимизации и его приложение к задаче синтеза искусственных нейронных сетей // Нейрокомпьютеры: разработка, применение. — 2004. — № 12. — С. 3—15.
7. Tukey J. W. Comparing individual means in the analysis of variance // Biometrics. — 1949. — № 9. — p. 99—114.
8. Andrew Y. Ng. Preventing overfitting of cross-validation data // Proc. 14th International Conference on Machine Learning. — San Mateo, CA, USA: Morgan Kauffman, 1997. — p. 245—253.
9. Klockars A. J., G. Sax G. Multiple Comparisons — Sage Publications, 1986. — 88 p.
10. Lawrence S., Giles C. L. Overfitting and neural networks: Conjugate gradient and backpropagation // Proceedings of the IEEE International Conference on Neural Net-

works (IJCNN'2000). — IEEE Press, 2000. — p. 114—119.
11. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection // IJCAI. — San Francisco, California, USA: Morgan Kaufmann, 1995. — p. 1137—1145.
12. Efron B. The Jackknife, the Bootstrap, and other resampling plans — Philadelphia, USA: SIAM, 1982. — 92 p. — ISBN 0-89871-179-7.
13. Bartlett P. L. For valid generalization, the size of the weights is more important than the size of the network // Advances in Neural Information Processing Systems 9 (1996). — USA: MIT Press, 1997. — p. 134—140.
14. Rissanen J. Modeling by shortest data description // Automatica. — 1978. — №14. — p. 465—471.
15. Ежов А. А., Шумский С. А. Нейрокомпьютинг и его применения в экономике и бизнесе. Сер. учебники экономико-аналитического института МИФИ. Под ред. проф. В. В. Харитонова. М.: МИФИ, 1998. — 220 с.
16. Хомич А. В. А. с. 2005611168 РФ. Программа для ЭВМ “Neurogenesis” №2005611168; Опубл. 24.02.2005.

Статья поступила 14 мая 2006 г.
После доработки 20 сентября 2006 г.