**ТЗ на разработку адаптивного алгоритма тестирования**

**Вступление**

Адаптивное тестирование – это такой вид компьютерного тестирования человека, в котором тест с помощью адаптивного алгоритма подстраивается под ответы заполняющего. В случае тестов способностей, если человек ответил на вопрос правильно, то уровень следующего вопроса повышается, а если неправильно, то уровень понижается. Это позволяет существенно сократить время тестирования (в классическом неадаптивном тесте кандидат должен ответить на вопросы всех уровней сложности, в том числе и на слишком легкие для него, и на слишком сложные), а кроме того повысить точности оценки способности (опять же из-за того, что человек не решает слишком простые или слишком сложные задания, которые не дают нам никакой информации об уровне способности человека). Адаптивное тестирование стало возможным благодаря развитию IRT (см. основные понятия) и компьютерных технологий. Немного подробнее про адаптивное тестирование можно прочитать на вики:

<https://en.wikipedia.org/wiki/Computerized_adaptive_testing>

Примечание. У меня встречаются понятия «вопрос», «задание», «пункт» - это взаимозаменяемые синонимы.

**Основные понятия**

Адаптивное тестирование строится на такой теории психометрики, как **IRT** (Item Response Theory, Теория ответов на задания). IRT – это математическая модель, которая моделирует вероятность определенного ответа (в случае тестов способностей – правильного ответа) на конкретное задание, учитывая при этом как характеристики тестируемого, так и особенности задания. В основе IRT лежит предположение, что за наблюдаемым поведением человека стоит некая устойчивая скрытая характеристика, и, наблюдая ряд примеров однотипного поведения, можно делать выводы о степени выраженности у человека этой характеристики. Таким рядом однотипного поведения является тест, измеряющий, например, числовую или вербальную способность, а эти способности как раз и являются устойчивыми скрытыми характеристиками. Таким образом, наблюдая ответы кандидата на разные задания теста, мы можем вычислить уровень его способности.

Существует множество IRT-моделей, использующих различные параметры, наиболее применимы самые простые из них. В наших тестах мы будем использовать одну модель в двух ее подвидах – 2-хпараметрическую логистическую модель. Она берет на вход всего одну характеристику тестируемого человека – уровень его способности, а также 2 параметра задания, имеющая следующий вид:

, где

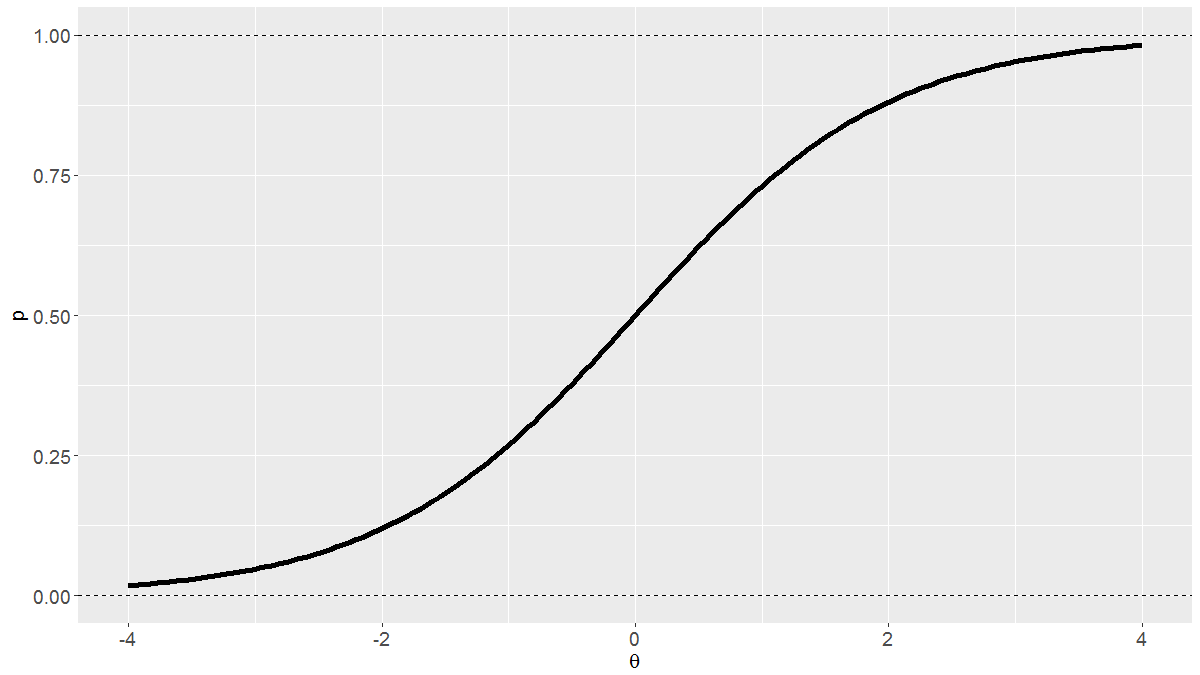
– ответ кандидата на вопрос *i* (правильный ответ кодируется 1)

– уровень способности человека *j*

– дискриминативность задания *i*

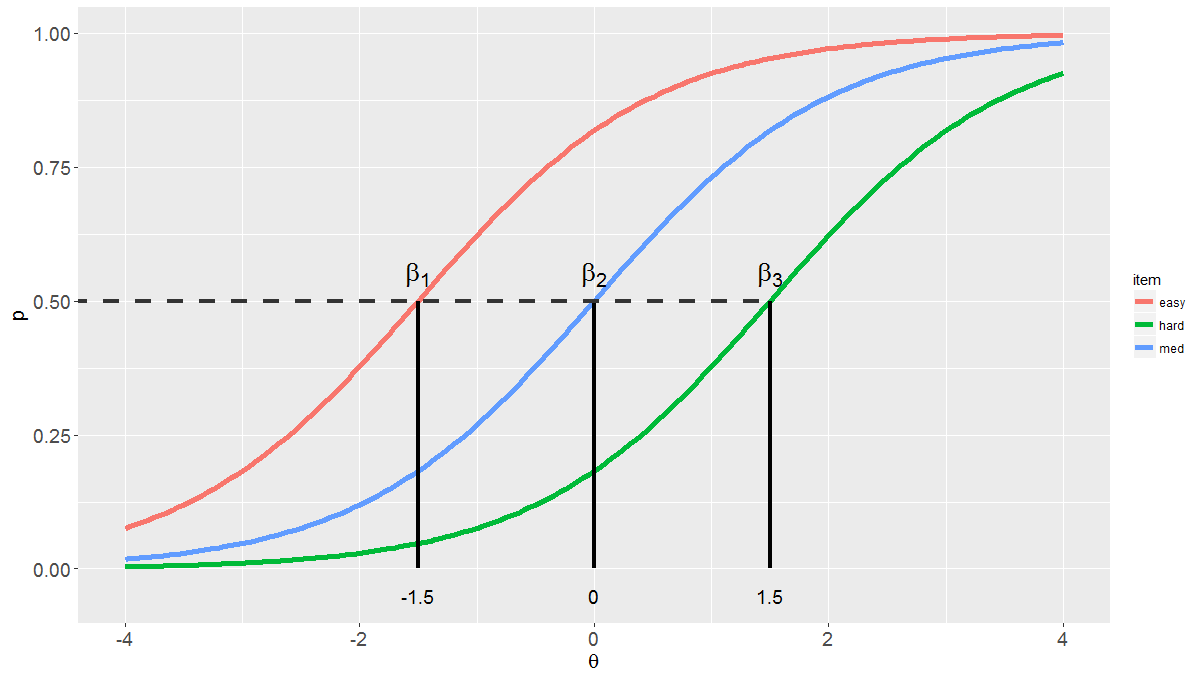
– сложность задания *i*

График изменения вероятности правильного ответа от уровня способности при 2-хпараметрической модели для задания со сложностью = 0 и дискриминативностью = 1:



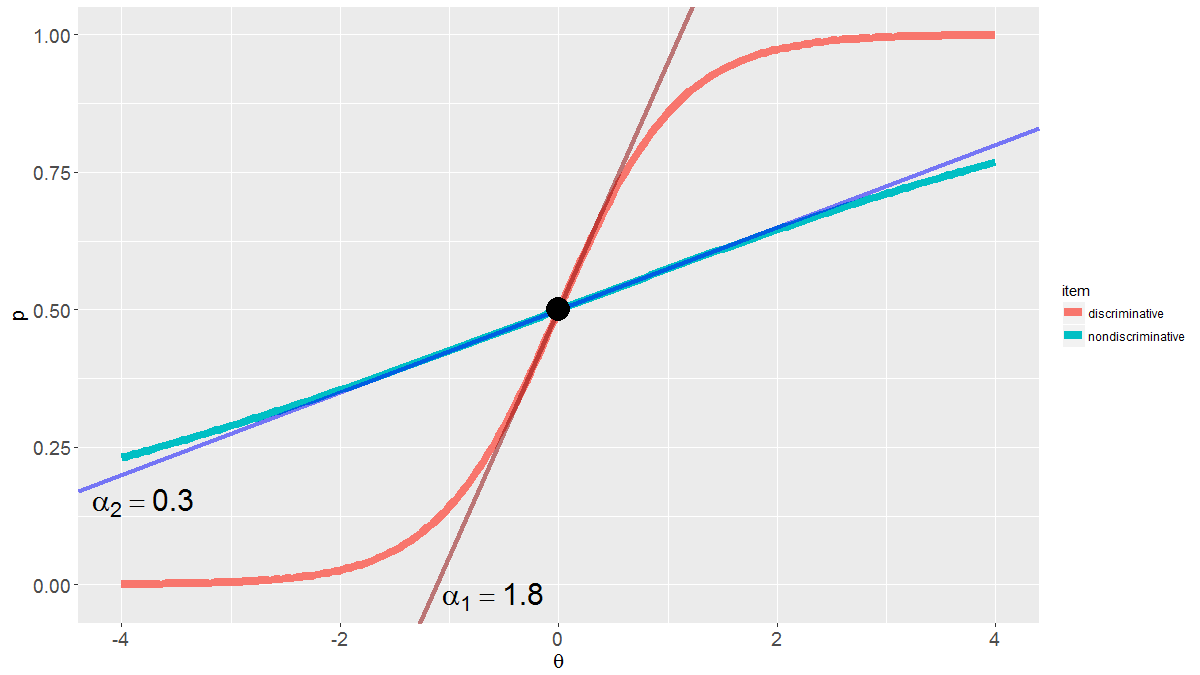
Рассмотрим параметры задания:

1. Сложность задания – это такой уровень способности кандидата, при котором вероятность правильного ответа на данное задание равна 0,5. Здесь интересно то, что в определении мы соотносим параметр задания (сложность) и уровень способности человека. В формуле эти два показателя также соотносятся – из одного вычитается другой. Если мы подставим в формулу значение способности человека равное сложности задания (например, и там, и там 0), то мы получим: . Отсюда мы видим, что, чтобы определить способность человека, нужно найти задание такого уровня сложности, которое он решит с вероятностью 0,5. График характеристических кривых заданий разных уровней сложности:



На графике обозначены простое задание (красное), задание среднего уровня (синее) и задание сложное (зеленое), а также отмечены их уровни сложности на шкале способностей. Графически сложность – это смещение функции вероятности по горизонтали: чем правее функция, тем сложнее задание, тем больший уровень способности (на оси Х) нужно иметь, чтобы выполнить задание с большой вероятностью.

1. Дискриминативность задания – это характеристика, которая показывает, насколько задание отделяет наиболее способных кандидатов от наименее способных. Дискриминативность может разниться от -∞ до +∞. Однако дискриминативность < 0 говорит о том, что менее способные кандидаты решают задание лучше, чем более способные кандидаты, поэтому практически применимы только те задания, в которых дискриминативность > 0. График вероятности решения заданий с одинаковой сложностью, но разной дискриминативностью:

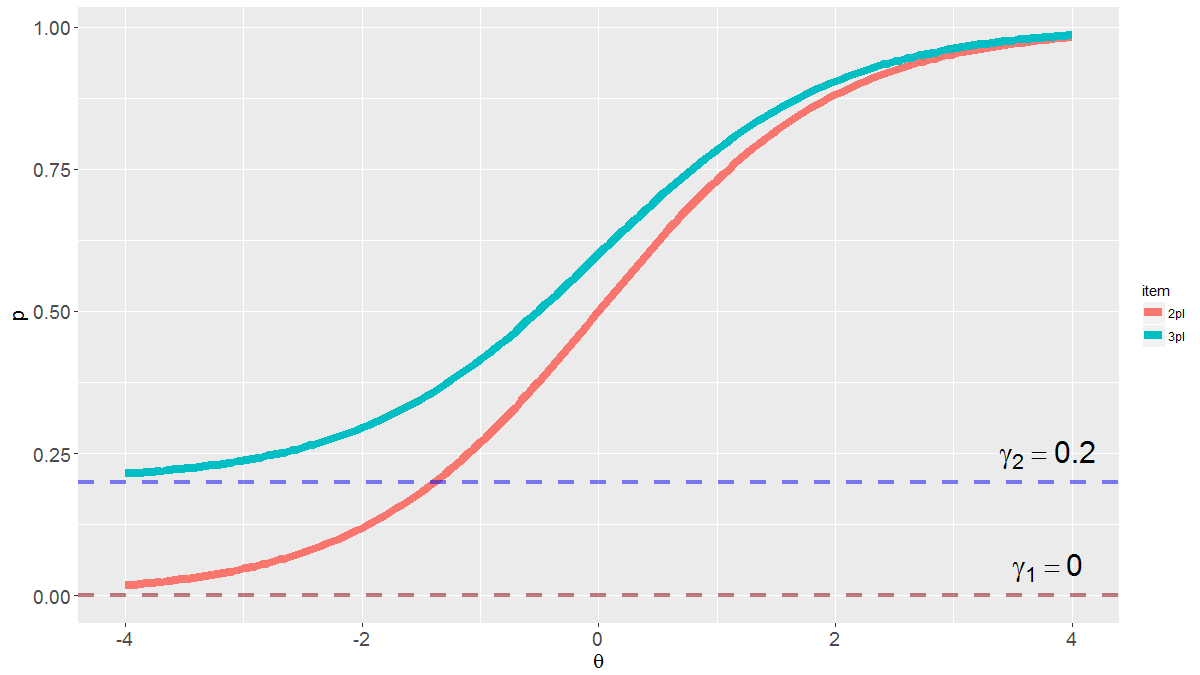


На графике нарисованы два задания одинаковой сложности (β = 0), но с разной дискриминативностью (α1 = 1,8; α2 = 0,3). Графически дискриминативность связана с углом наклона касательной к графику функции вероятности в точке сложности вопроса (на графике – черная точка, Х = 0, У = 0,5). Чем больше дискриминативность пункта, тем более «крутой» выглядит кривая. Концептуально, если кандидат успешно выполнил красненький (более дискриминативный) вопрос, то мы с большей уверенностью можем утверждать, что его способность больше 0, чем если он успешно выполнил синенький (менее дискриминативный) вопрос.

Кроме того, мы будем использовать 2хпараметрическую модель с фиксированным параметром случайного угадывания – вероятностью ответить на задание правильно путем угадывания. Данная модель имеет формулу:

,

где – параметр случайного угадывания. На графике такая кривая выглядит так:

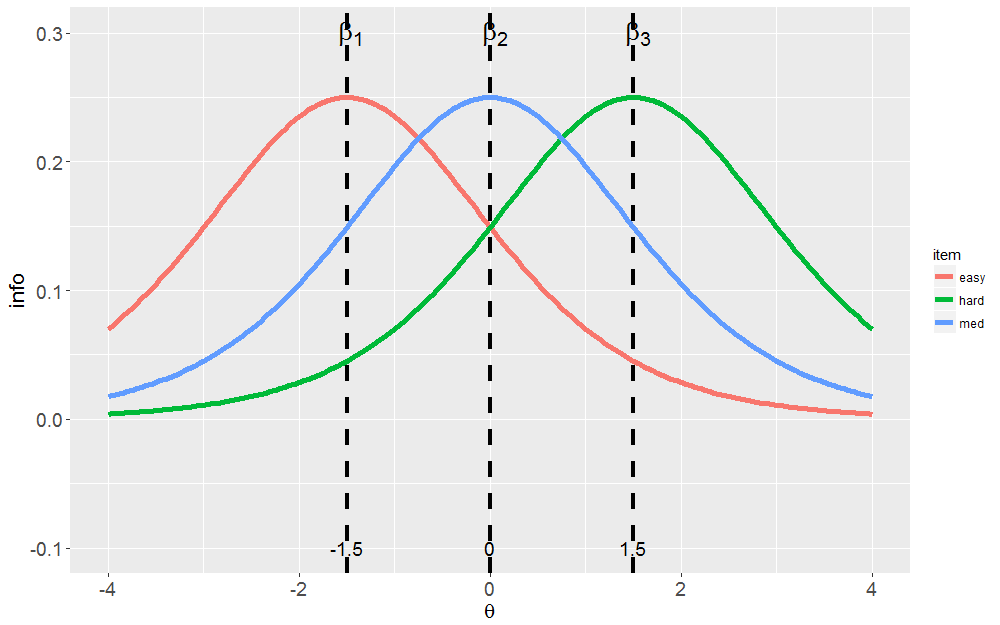


Параметр случайного угадывания поднимает нижнюю асимптоту графика на величину параметра, т.е. мы считаем, что вероятность правильного ответа на вопрос не опускается ниже вероятности случайного угадывания. Если кандидат обладает низким уровнем способности (левая часть оси Х), то ему проще угадывать ответ, чем пытаться отвечать «как положено».

Другим важным понятием является понятие информации пункта – о каком уровне способности задание дает больше информации. Информация рассчитывается по формуле:

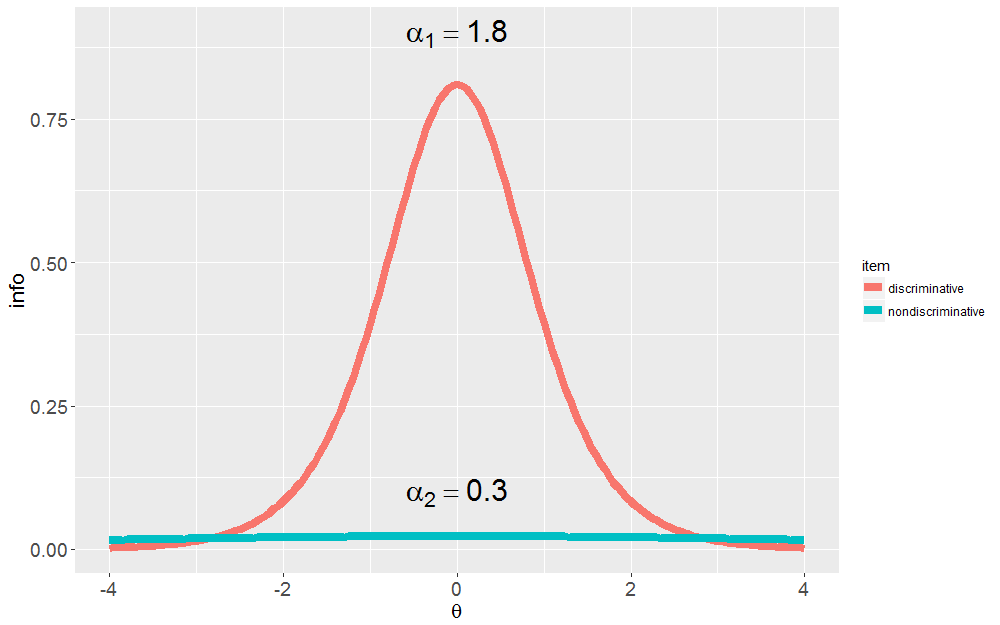
,

где p – это вероятность правильного ответа на задание (см. выше), а p’ – это первая производная этой вероятности (см. ниже). Примеры графиков информации – графики построены по тем же параметрам, что все графики выше и сгруппированы соответственно. Информация по заданиям разной сложности:



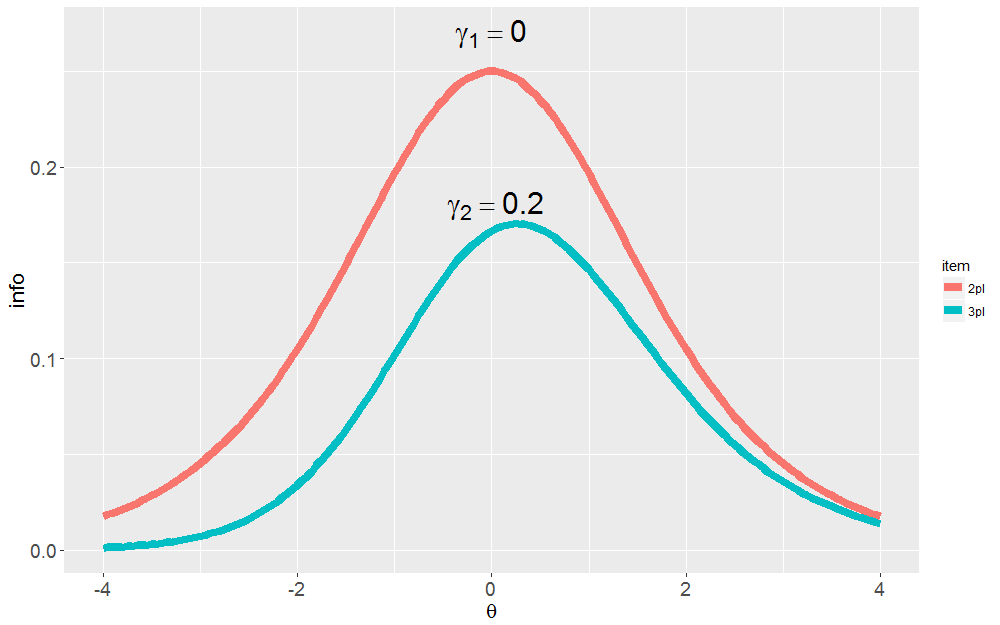
Как видно из графика простой вопрос дает больше информации по кандидатам на низком уровне способности, сложный – на высоком, средний – на среднем. Кроме того, мы видим, что максимум информации мы получаем в точке уровня сложности задания (вертикальные линии).

Информация заданий с разной дискриминативностью:



Как видно из графика более дискриминативный пункт дает гораздо больше информации об определенному уровне способности, чем менее дискриминативный пункт (хотя они и одного уровня сложности). Интересно также заметить, что менее дискриминативный пункт дает больше информации о способностях выше 3 и ниже -3, чем дискриминативный.

Информация пунктов с разным параметром случайного угадывания:



Как мы видим, пункт без параметра случайного угадывания дает нам больше информации почти на всех уровнях способности (это связано с тем, что кривая пункта со случайным угадыванием более пологая). График информации по пункту с параметром угадывания несимметричный, кроме того, пик информации смещен вправо относительно уровня сложности этого пункта (потому что точка перегиба на кривой пункта тоже правее – см. график 4).

Информация о вопросе – это важный концепт для адаптивного теста, поскольку на ее основании идет отбор следующего задания – об этом далее. Подробнее об IRT – например, <https://en.wikipedia.org/wiki/Item_response_theory>

**Концептуальный процесс прохождения адаптивного теста**

Цель адаптивного теста – найти такое задание, уровень сложности которого равен уровню способности кандидата (т.е. которое кандидат решит с вероятностью 0,5). Это связано с тем, что в зоне сложности задание дает наибольшую информацию о способности. Как это решается?

Есть банк заданий с параметрами и кандидат, который заполняет тест. Первая задача – это предъявить ему первое задание. Обычно берется задание среднего уровня сложности, поскольку больше всего народу со средними способностями, соответственно, оттуда скорее всего быстрее пройдет адаптирование под уровень кандидата. Как только кандидат решил первое задание, мы определяем его предварительный уровень способности и подбираем задание под этот уровень. Такая последовательность сохраняется для всех последующих заданий (определяем уровень способности, затем подбираем задание). Как мы подбираем задание? Метод называется определение максимальной апостериорной информации Фишера:

1. Для всех заданий мы знаем их информацию на некотором диапазоне значений способности (я использовал 33 значения от -4 до 4) – т.е. для каждого задания у нас есть вычисленная функция информации (график ее см. выше).
2. У нас есть паттерн ответов кандидата, т.е. последовательность 0 и 1, где 1 обозначает правильно выполненное задание, а 0 – неправильно выполненное задание. Исходя из этого паттерна, мы можем вычислить функцию правдоподобия паттерна относительно предъявленных заданий по формуле:

где – это вероятность правильного ответа на задание i,

*u* – это ответ кандидата на задание I (1 или 0),

*i = 1,…,k* – это последовательность вопросов, уже предъявленных кандидату.

Мы можем рассчитать значения функции правдоподобия на том же диапазоне способностей (мои 33 значения от -4 до 4) для испытуемого. Получится тоже некая кривая с единичным максимумом.

1. Кроме того, у нас есть априорное представление о распределении способностей (включаем Байесовскую статистику). Это просто стандартное нормальное распределение (среднее 0, стандартное отклонение 1), значения которого мы рассчитаем на том же диапазоне способностей (33 значения от -4 до 4).
2. Итого: для каждого значения способности из выбранного диапазона (который всё те же 33 числа от -4 до 4) у нас посчитано 3 величины: информация по каждому заданию из банка; значение функции правдоподобия паттерна ответов; значение стандартного нормального распределения. Перемножаем эти 3 величины для каждой способности в диапазоне для каждого задания из банка.
3. После перемножения у нас есть 33 числа для каждого задания из банка. Чтобы получить одно число для каждого пункта вместо 33, нам нужно проинтегрировать полученную информацию. Я использовал (вроде бы) метод прямоугольника, хотя другие методы тоже сгодятся (метод Симпсона, например). Интегрирование я не могу объяснить письменно, могу показать и объяснить код.
4. В результате у каждого непредъявленного задания есть финальная апостериорная информация, которую мы получим, если предъявим это задание кандидату. Очевидно, что мы выбираем пункт с максимальной ожидаемой информацией.

Далее мы предъявляем пункт с максимальной ожидаемой информацией, смотрим ответ кандидата, рассчитываем предварительный уровень его способности и опять выбираем новый пункт по тому же методу. Заканчиваем, когда достигнуто определенное число предъявленных пунктов (пока у нас это 15 – здесь нужно это прописать в интерфейсе). После окончания предъявления заданий рассчитываем окончательный уровень способности и стандартную ошибку измерения кандидата.

У этого метода подбора пунктов есть одна проблема – он будет предпочитать более дискриминативные пункты менее дискриминативным. Почему – см. график функции информации для пунктов с разной дискриминативностью. Следовательно, пункты будут предъявляться неравномерно – дискриминативные часто, а менее дискриминативные редко. Чтобы этого избежать, нужно использовать методы контроля предъявлений. Мы используем 2 метода.

Первый метод - Ограничивающий. Он заключается в том, что система не предъявляет задания, которые до этого предъявлялись слишком часто. Он простой:

1. Ведется запись количества человек, прошедших тест в системе.
2. Ведется запись количества предъявлений каждого пункта.
3. Если отношение количества предъявлений к количеству человек больше 0,4, то такой пункт изымается из банка до тех пор, пока это число не упадет меньше 0,4.

Второй метод – Прогрессивный. Он заключается в том, что мы отбираем задания не только по их ожидаемой информации, но и немного случайным образом. Дело в том, что в начале сессии тестирования мы знаем о способности кандидата очень мало, и он может решить задания не так, «как надо» (например, вдруг решить сложное для него или не решить простое), соответственно, система предъявит не тот пункт. А если он будет очень дискриминативным (поскольку они дают очень много информации), то мы можем застрять на уровне способности, который не соответствует способности нашего кандидата. Для этого и для увеличения частоты предъявления всех пунктов мы делаем следующее:

1. После того, как мы нашли ожидаемую апостериорную информацию Фишера для каждого пункта, мы находим пункт с максимальной информацией (назовем это **H**)
2. Далее мы набираем случайных чисел из равномерного распределения от 0 до H по количеству оставшихся в банке пунктов (назовем это **R**)
3. Затем мы вычисляем отношение порядкового номера предъявляемого пункта в сессии (например, если это второй пункт из 15, то 2/15). Назовем это число **S**
4. Затем к полученной информации Фишера по каждому пункту добавляем случайную информацию по формуле:

где – это информация Фишера (как она рассчитывается по алгоритму на предыдущей странице) для задания i,

- это случайный компонент задания i,

- это относительная позиция задания в тестовой сессии,

– это число для изменения доли информации и рандомного компонента по отношению к позиции задания в сессии (чаще всего 1),

- это финальная информация, которая используется для отбора пунктов.

Соответственно, мы выбираем пункт с максимальной полученной величиной. Стоит заметить, что чем дальше задание расположено в сессии, тем больший вес в итоговой информации имеет информация Фишера – дальше середины сессии фактически отбор пунктов идет по информации Фишера с небольшим случайным компонентом. И наоборот, вначале превалирует случайный компонент.

**Описание работы алгоритма**

Алгоритм можно разбить на отдельные модули, каждый из которых принимает на вход некоторые данные, что-то вычисляет и отправляет дальше в другие модули. Каждого модуля может быть несколько вариантов (взаимозаменяемых), поэтому, мне кажется, нужно в системе предусмотреть возможное добавление этих модулей. Модули следующие:

**Банк заданий**

Банк заданий – все задания, из которых алгоритм делает выбор тех, что предъявляются кандидату. Банк заданий для алгоритма – это таблица, где каждая строчка – это задание, а столбцы – это параметры заданий. Параметры – это параметры заданий (дискриминативность, сложность, вероятность случайного угадывания – см. формулу IRT), а также параметр для контроля предъявлений – обновляемое число предъявлений данного задания всем кандидатам (см. блок контроля предъявлений) и параметры содержательного контроля (будут актуальны только для вербального теста) – параметр типа вопроса (в вербальном тесте 4 типа вопросов – например, на нахождение синонимов или определения значения, - нужен для выравнивания количества предъявлений заданий каждого типа одному кандидату) и еще один параметр (не знаю, как его определить, поэтому опишу). В вербальном тесте у каждого стимульного есть вопросы, у которых один и тот же правильный ответ (например, есть вопрос – какой основной вывод следует из текста – понятно, что вывод один на текст, а вопросов к тексту два), следовательно, этот параметр контролирует, если один из таких вопросов был предъявлен кандидату, то второй не должен быть предъявлен. Для числового теста предлагаю эти последние параметры сделать NULL, и их игнорировать.

**Паттерн ответов кандидата**

Список ответов кандидата на предъявленные ему задания. Кодировка – 1 (правильно), 0 (неправильно).

**Список предъявленных кандидату вопросов**

Список заданий, которые были уже предъявлены кандидату в тестовую сессию. Он нужен, чтобы не предъявлять кандидату эти вопросы по второму разу и для контроля количества предъявлений.

**IRT-модель**

IRT-модель – модель, которая сопоставляет характеристики кандидата и характеристики задания с вероятностью ответа кандидата на задание (см. описание IRT выше). Для алгоритма это просто функция вычисления вероятности ответа на задание кандидатом и ее производная для вычисления функции информации. Общая формула, которая будет использована в адаптивных тестах – это 2хпараметрическая модель с фиксированным третьим параметром – вероятностью случайного угадывания. Формула следующего вида:

,

где – это параметр случайного угадывания для задания *i*. Для числового теста этот параметр будет равен 0 (полностью двухпараметрическая модель, как описано во введении). В случае числового теста используются вопросы с выбором из выпадающего списка, соответственно, там очень много вариантов, и вероятность случайного угадывания не сильно отличается от нуля. Для вербального теста этот параметр будет ненулевым. Для вычисления информации пункта нужно вычислить производную вероятности правильного ответа на вопрос по формуле:

Здесь же вычисляется и функция информации. Если – это вероятность неправильно ответить на вопрос, то информация рассчитывается по формуле:

**Метод измерения способности человека**

Метод измерения способности человека – как мы на основании завершенных заданий оценивается уровень способности кандидата. Этот метод применяется для предварительной оценки способности при выборе следующего задания, для финальной оценки после прохождения теста, кроме того, он используется для измерения ошибки измерения теста после его прохождения. Мы используем метод EAP (Expected A Posteriori). Из названия следует, что это тоже Байесовский метод. Он заключается в том, что мы берем возможный диапазон способностей кандидата (как при определении информации Фишера – см. Концептуальный процесс прохождения теста) – я брал все те же 33 числа от -4 до 4 (назовем из X) и считаем с ним следующее:

1. Для этих 33 чисел у нас есть априорное распределение способности – все то же стандартное нормальное (назовем N)
2. Для этих 33 чисел мы можем посчитать функцию правдоподобия паттерна (так же, как при расчете информации Фишера – назовем L)
3. Для каждого из этих 33 чисел мы перемножаем это число, значение для него по нормальному распределению и значение для него функции правдоподобия и интегрируем: X\*N\*L
4. Затем перемножаем попарно значения нормального распределения этих чисел и функции правдоподобия (то же самое, что в пункте 3, но без самих чисел) и интегрируем: N\*L
5. Делим 3 пункт на 4: integrate(X\*N\*L) / integrate(N\*L)

В результате получаем уровень способности кандидата (назовем его T). Не могу объяснить письменно, почему так ☺

Стандартная ошибка измерения теперь. Вычисляется похоже:

1. Вычитаем из каждого X вычисленный уровень способности T и возводим в квадрат: (X-T)^2
2. Домножаем на N и L и интегрируем
3. Делим полученное на тот же знаменатель, что и выше: integrate((X-T)^2\*N\*L) / integrate(N\*L)

Результат – стандартная ошибка измерения.

**Метод выбора стартового пункта**

Первый пункт выбирается случайно из 5 пунктов, чья сложность ближе всего к 0.

**Метод выбора следующего пункта**

Их много. Мы пока используем Максимальную апостериорную информацию Фишера – см. выше.

**Методы контроля предъявлений**

Их может быть несколько. Мы используем пока 2 – прогрессивный и ограничительный. См. выше.

**Что на выходе алгоритма?**

1. Финальная информация по кандидату – уровень способности и стандартная ошибка измерения
2. Информация по работе алгоритма – предварительные уровни способности кандидата (в сырую выгрузку) – нужно для оценки качества работы алгоритма
3. По каждому кандидату в сырую выгрузку – паттерн ответов на вопросы – матрица (количество кандидатов х количество вопросов в банке), где 1 – кандидат правильно ответил на данный вопрос, 0 – кандидат неправильно ответил на данный вопрос, пустота – кандидату не был предъявлен данный вопрос. Данную матрицу можно использовать и для подсчета количества предъявлений каждого пункта из банка – для ограничительного метода контроля предъявлений.
4. Общее число кандидатов на настоящий момент, т.е. после каждого кандидата +1 – тоже для ограничительного метода.