Data Science для решения задач информационной безопасности

Лекция 3. RSA Risk Engine, RSA Rules Engine

Павел Владимирович Слипенчук

8 октября 2019

Москва, МГТУ им.Бауманка,
каф.ИУ-8, КИБ

План лекции

1. Обратная связь
2. Обратная связь в RSA
3. RSA Risk Engne
4. RSA Rules Engine

Обратная связь

Историческая заметка

1. В 1935 году советский физиолог Пётр Кузьмич Анохин формулирует понятие обратной связи.
2. Термин «перекочевал» в Кибернетику (1948) Н.Винера
3. В настоящее время термин «обратная связь» используется в медицине, в технике, в акустике, в биологии, в социальных науках.

Обратная связь в DS

Обратная связь (feedback) I – это любая информация, используемая для оценки качества ЭС и/или для получения новой *обучающей выборки* (или выборки *обучения с подкреплением*)

Обратная связь (feedback) II – это процесс, система, программный модуль, реализующий обратную связь I

Обучение с подкреплением, reinforcement learning (Дообучение, Refitting)[1]

Есть первоначальная обучающая выборка:

$$U_{fit} = \{y \mapsto \mathsf{x}\}$$

Первоначальное обучение:

$$score_1 := fit(U_{fit})$$

В дальнейшем при получении нового множества $\hat{U}_{fit}$ (возможно состоящее из одного элемента) мы дообучаем систему:

$$score_{i+1} := refit(\hat{U}_{fit}, score_i) \tag{1}$$

Таким образом функция $score_i$ заменяется на новую функцию $score_{i+1}$.

[1] слайд из лекции №2

Итеративное обучение

Есть первоначальная обучающая выборка:

$$\hat{U}_1 = U_{fit} = \{y \mapsto \mathbf{x}\}$$

Первоначальное обучение:

$$score_1 := fit(\hat{U}_1) = fit(U_{fit})$$

В дальнейшем при получении нового множества $\hat{U}_i$ (возможно состоящее из одного элемента) мы повторно обучаем систему:

$$score_{i+1} := fit(U_{fit} \cup \hat{U}_2 \cup ... \cup \hat{U}_i) \tag{2}$$

Таким образом функция $score_i$ заменяется на новую функцию $score_{i+1}$.

Верно ли утверждение что итеративное обучение позволяет для любого алгоритма осуществить обучение с подкреплением?

Стабильность. «Цепочка»

Есть итеративная система обучения и имеем скоринги:

$$score_{i-n} := fit(\hat{U}_1 \cup \hat{U}_2 \cup ... \cup \hat{U}_{i-n} \cup ... \cup \hat{U}_{i-1} \cup \hat{U}_i)$$

$$...$$

$$score_{i-1} := fit(\hat{U}_1 \cup \hat{U}_2 \cup ... \cup \hat{U}_{i-1} \cup \hat{U}_i)$$

$$score_i := fit(\hat{U}_1 \cup \hat{U}_2 \cup ... \cup \hat{U}_i)$$

Тогда задав веса $w_j > 0$ можно сделать простой ансамбль вида:

$$\mathbf{score_i}(\mathbf{x}) = \frac{w_1 \cdot score_{i-n}(\mathbf{x}) + ... + w_n \cdot score_i(\mathbf{x})}{\sum_{j=1}^{n} w_j}$$

Например:

$$\mathbf{score_i}(\mathbf{x}) = 0.2 \cdot score_{i-2}(\mathbf{x}) + 0.3 \cdot score_{i-1}(\mathbf{x}) + 0.5 \cdot score_i(\mathbf{x})$$

Возвращаемся к обратной связи

Зачем нужна обратная связь в DS?

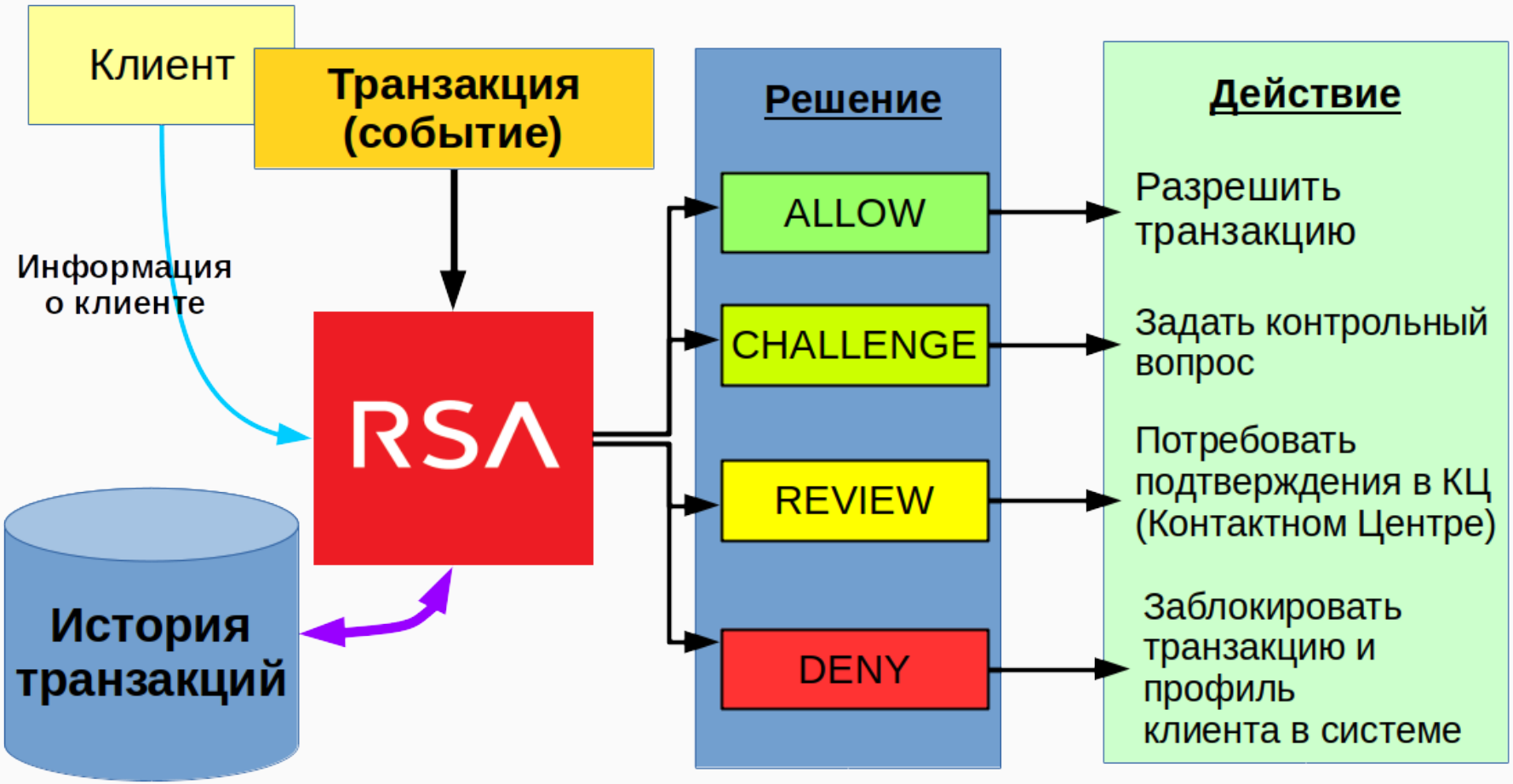
Feedback позволяет:

1. Изменить *score* на основании предыдущих *откликов* и размеченных данных. Например при каждой сработке можно с вероятностью p проводить расследование и дообучать систему.
2. генетические алгоритмы: изменение параметров
3. позволяет выявлять аномалии[2], корректировать мат.модель, выбирать из ML моделей лучшую.

[2] «аномалитика» :)

Обратная связь в RSA

Общая схема


Клиент
Транзакция (событие)
Информация о клиенте
RSA
История транзакций
Решение
ALLOW
CHALLENGE
REVIEW
DENY
Действие
Разрешить транзакцию
Задать контрольный вопрос
Потребовать подтверждения в КЦ (Контактном Центре)
Заблокировать транзакцию и профиль клиента в системе

Последовательность

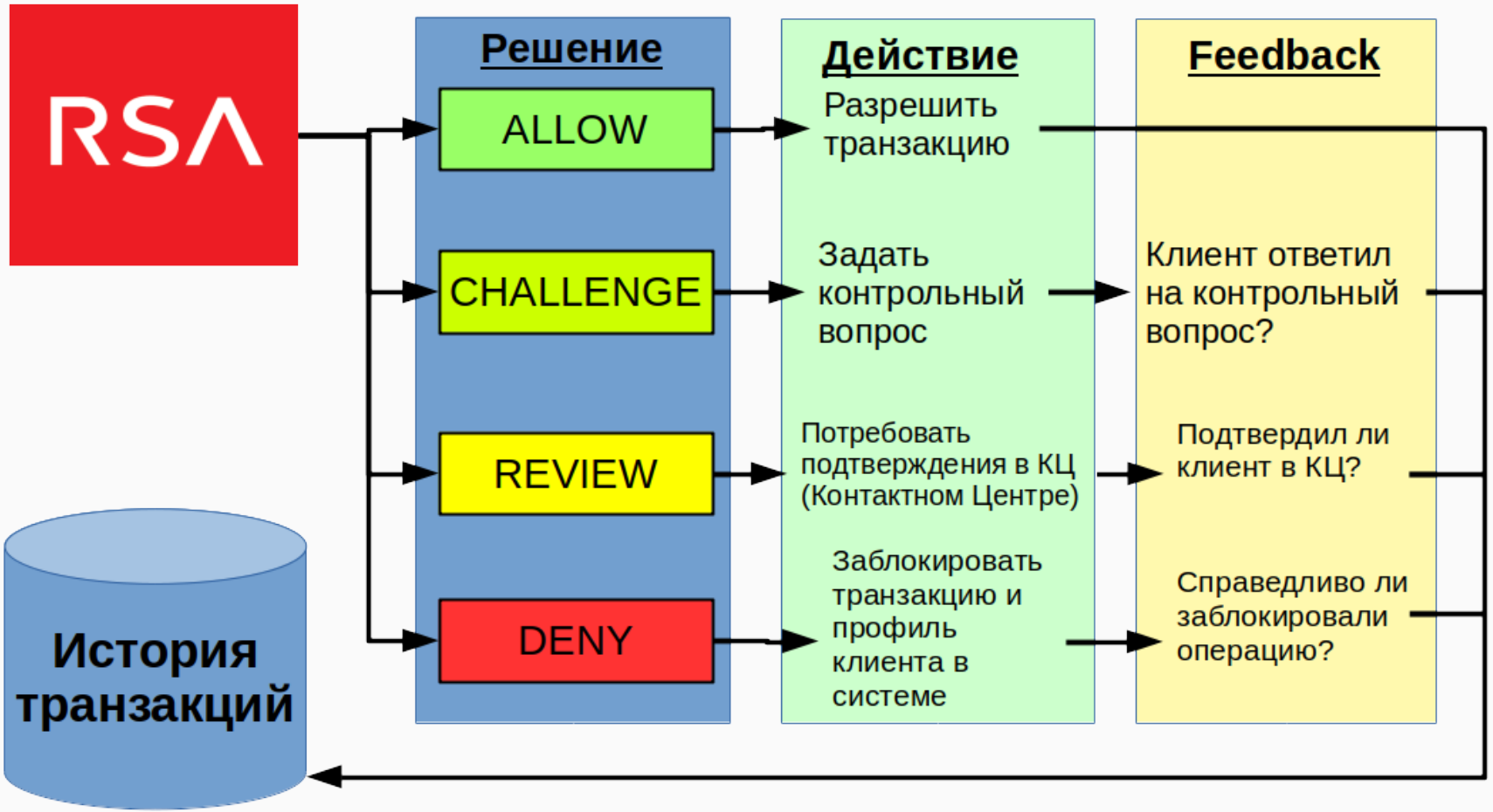
Построение:

1. Выбираются данные за $T = 90$ дней.
2. Строится оценщик (Risk Engine)
3. Анализируется мошенничество заказчика и создаются правила

Эксплуатация:

1. Расчёт вектора признаков $\mathbf{x}$
2. Risk Engine
3.
4. Rules Engine
5. Процессинг системы

Обратная связь. Сработки системы



Обратная связь. Жалобы

Если у клиента украли деньги и он это заметил, последовательность следующая:

1. клиент пишет жалобу/звонит/иным способом сообщает это
2. специальный человек проверяет корректность жалобы. Если она корректна, то данные помещаются как F.

Контактный центр принимает звонки при всех REVIEW операциях. Если клиент не позвонил, какие выводы?

При DENY операциях проводятся специальные расследования с помощью особого подразделения внутри самого банка.

CUSTOM_MARK

Маркировка проходит вручную.

- CUSTOM_MARK=U (Uknoun) – неизвестная транзакция
- CUSTOM_MARK=G (Genuine[3]) – подлинная
- CUSTOM_MARK=F (Fraud) – фродовая
- CUSTOM_MARK=S (Suspicious) – скорее всего фродовая
- CUSTOM_MARK=A (Authentic) – скорее всего подлинная
- CUSTOM_MARK=Null – не промаркировано.

[3] RSA олдскульная система. Вместо флага L(legitim) в ней используется флаг G

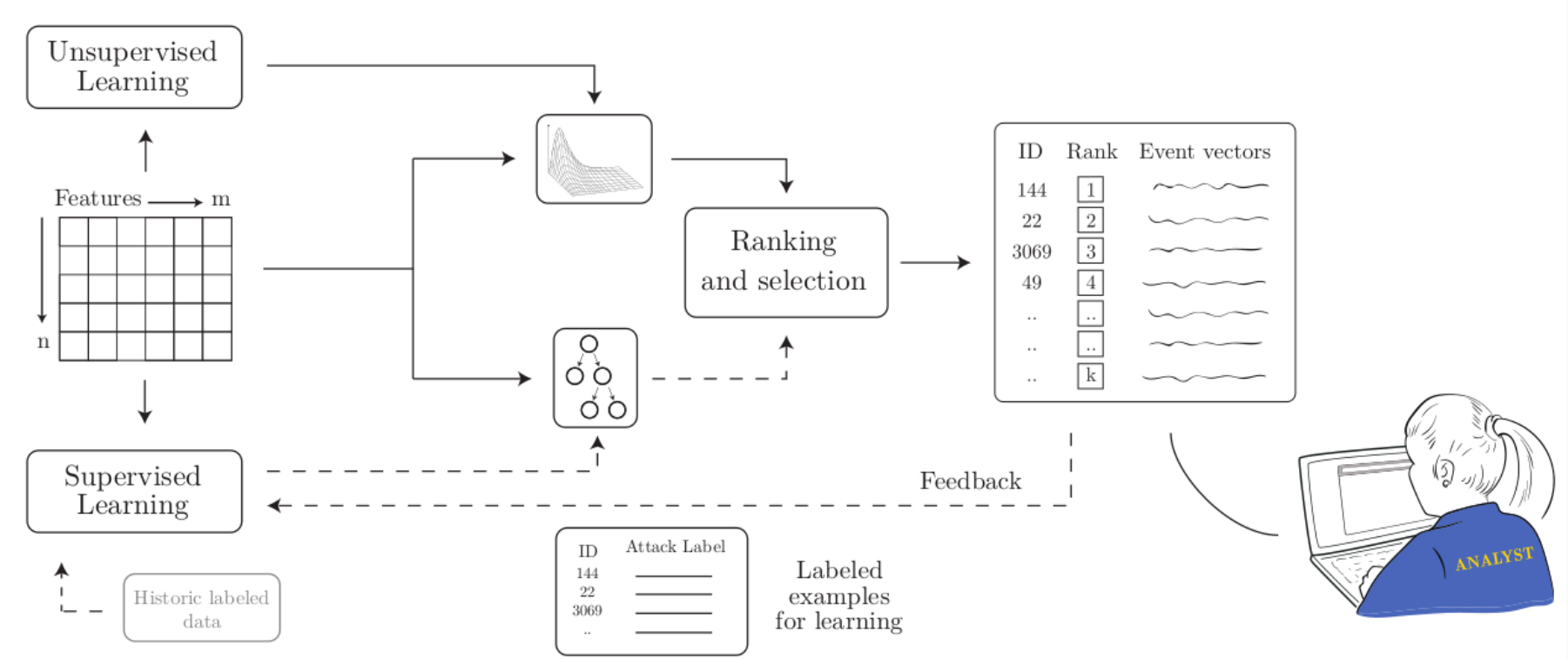
Определение класса

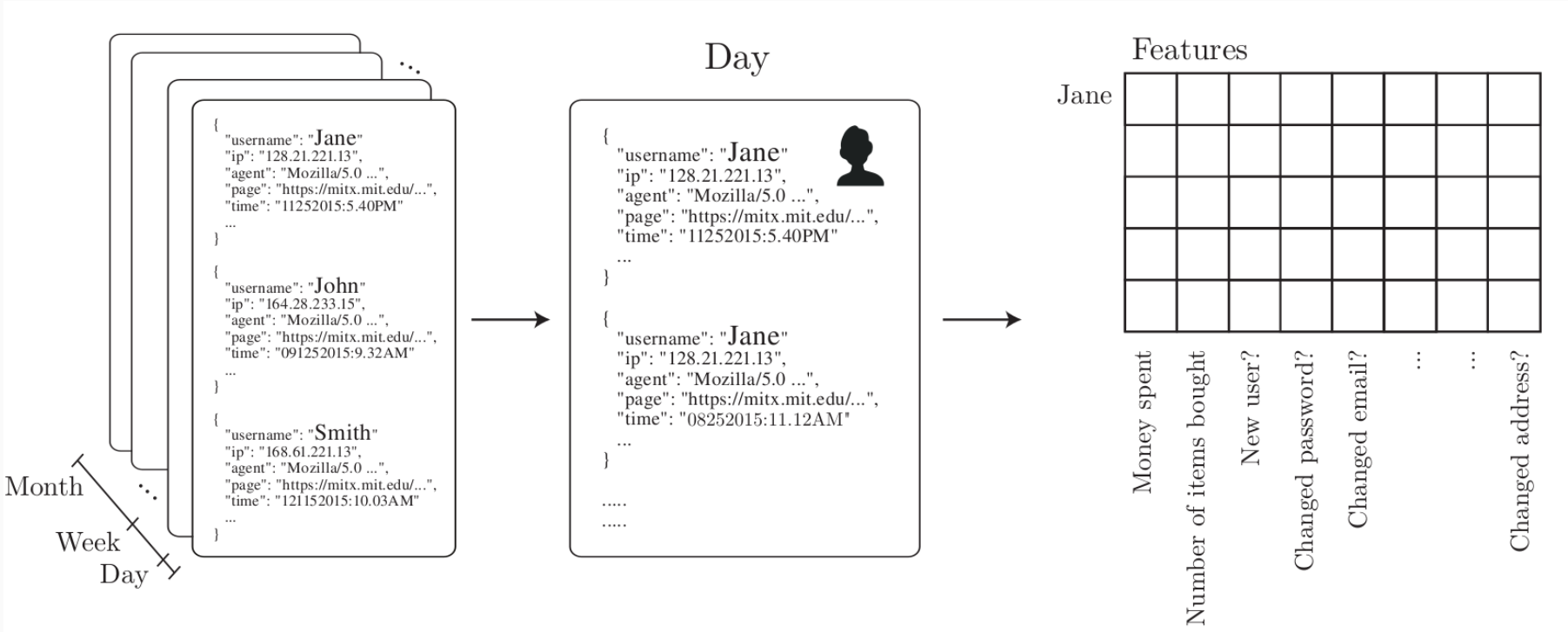
По CUSTOM_MARK определяется класс транзакции:

- 0 – легитимная
- 1 – мошенническая
- x – не используется

CUSTOM_MARK	класс	Примечание
U	x	
G	0	помечаются транзакции всей сессии
F	1	помечаются транзакции всей сессии
S	1	
A	0	
Null	0 или x	Спустя много дней считается легитимной

Статья «AI^2: Training a big data machine to defined» Kalyan Veeramachaneni, Ignacio Arnaldo и др.




Month
Week
Day
{
"username": "Jane"
"ip": "128.21.221.13",
"agent": "Mozilla/5.0 ...",
"page": "https://mitx.mit.edu/...",
"time": "11252015:5.40PM"
...
}
{
"username": "John"
"ip": "164.28.233.15",
"agent": "Mozilla/5.0 ...",
"page": "https://mitx.mit.edu/...",
"time": "091252015:9.32AM"
...
}
{
"username": "Smith"
"ip": "168.61.221.13",
"agent": "Mozilla/5.0 ...",
"page": "https://mitx.mit.edu/...",
"time": "121152015:10.03AM"
...
}
Day
{
"username": "Jane"
"ip": "128.21.221.13",
"agent": "Mozilla/5.0 ...",
"page": "https://mitx.mit.edu/...",
"time": "11252015:5.40PM"
...
}
{
"username": "Jane"
"ip": "128.21.221.13",
"agent": "Mozilla/5.0 ...",
"page": "https://mitx.mit.edu/...",
"time": "08252015:11.12AM"
...
}
.....
.....
Features
Jane
Money spent
Number of items bought
New user?
Changed password?
Changed email?
...
...
Changed address?

RSA Rules Engine

Risk Engine & Rules Engine

- ***Risk Engine*** – это классификатор, возвращающий значения RISK_SCORE от 0 до 1000. Чем выше это значение, тем более вероятность мошенничества.
- ***Rules Engine*** – это ЭС, принимающая на вход значения *Risk Engine* и другие признаки транзакции, и выдающая решения ALLOW, REVIEW, CHALLANGE или DENY.

Risk Engine проектируется компанией DELL, Rules Engine поставляется как движок правил.

Офицеры безопасности заказчика, используя движок Rules Engine разрабатывают правила фрод мониторинга.

Замечание

Для заказчика Risk Engine – это чёрный ящик, он не вмешивается в его работу. Risk Engine даётся «as is», или для крупных заказчиков настраивается отдельно.

Маркетинговая информация:
`https://www.emc.com/collateral/hardware/h9096-rsa-risk-engine-sb-11-2.pdf`

RSA Risk Engne

Замечание

Описание Risk Engne актуально на 2015 год. Возможно многие моменты изменились (вряд ли, т.к. это жёстокий энтерпрайз)

Контрибуторы

Все признаки разбиваются на ***контрибуторы***[4] по 1, 2, 3 или 4 признака, но не более.

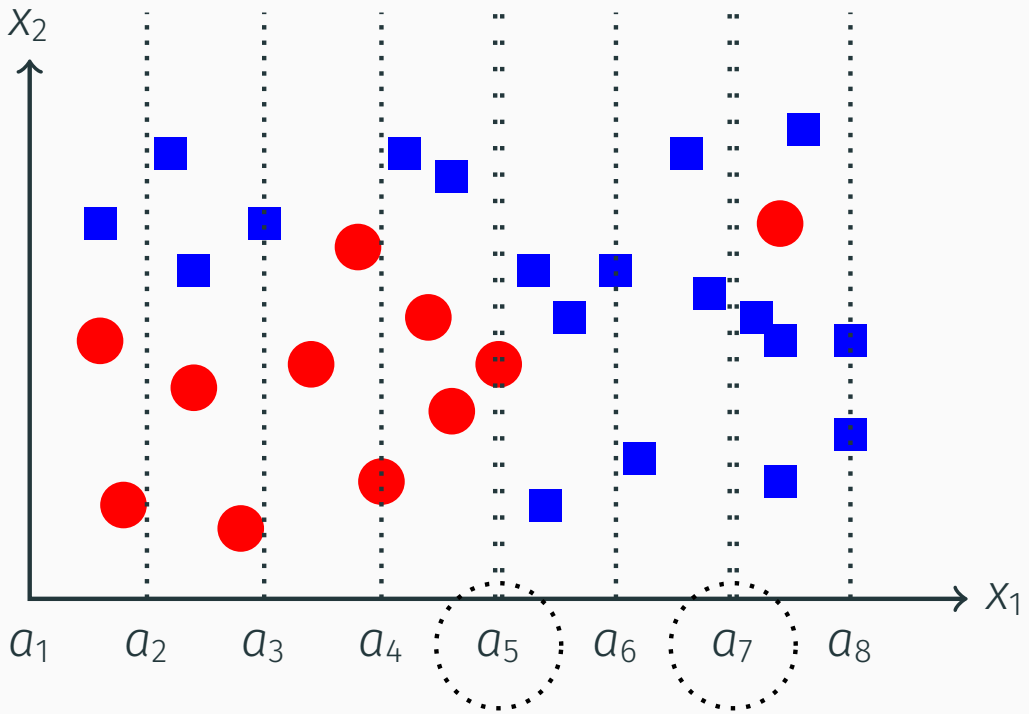
Таким образом имеем R^1, R^2, R^3 или R^4 *пространство признаков.*

Контрибуторы R^n, где $n > 4$ не используются.

[4] ***Контрибутор*** (контрибьютер) – это совокупность признаков (возможно один), который вносит определённый вклад в скоринговую модель. См. лекцию №2

Скользящее окно. Разбиновка.

Определяем окно Δ_i для каждого x_i признака *контрибутора* $(x_1, .., x_n)$



Определяем оптимальные бины (с помощью Индекса Джини). Они определяются либо в рамках отрасли (все банки), либо в рамках заказчика. Это – экспертное решение DS-специалиста в рамках каждого *контрибутера*.

Какова сложность алгоритма для R^n? Почему в RSA не используют контрибутеры сложности более чем 4?

Категория

Процедура разбиновки осуществляется либо единожды, либо при каждом повторном обучении, в зависимости от настроек, вручную заданные экспертом.

Категория – это функция скоринга контрибьютера в диапазоне $[-C_{MAX}, +C_{MAX}]$. Эта величина рассчитывается в каждом бине (bin) через *Индексы Джини*:

$$C(bin) \stackrel{def}{=} log_{coef} \frac{q \cdot \Delta_1(bin)}{\Delta_0(bin)} \tag{3}$$

Где коэффициент $coef$ задаётся экспертом вручную.

Величина q – это величина, показывающая во сколько раз **во всех бинах** данных легитимных транзакций больше данных мошеннических транзакций:

$$q \stackrel{def}{=} \frac{\sum_{\forall bin} \Delta_0(bin)}{\sum_{\forall bin} \Delta_1(bin)} \tag{4}$$

Категория

Физический смысл категории в том, что если $C(bin) = 0$, то данный бин не репрезентативный.

Если $C(a) > 0$, то a скорее более мошеннический бин, чем легитимный.

Если $C(b) < 0$, то b скорее более легитимный бин, чем мошеннический.

Формула (3) дополняется следующими условиями:

1. Если данных в бине мало, то $C(bin) := 0$.
2. Если $\Delta_0(bin) = 0$, то $C(bin) := C_{MAX}$,
3. Если $\Delta_1(bin) = 0$, то $C(bin) := -C_{MAX}$
4. Если значение выходит за рамки $[-C_{MAX}, +C_{MAX}]$ то присваивается минимально возможное / максимально возможное значение: $-C_{MAX}$, C_{MAX}.

Проблема «скачков» у функции логарифма

Так как у функции логарифма есть скачки возле нуля, то вводят *монотонно не убывающую* функцию F.
Переопределим рассчёт категории:

$$C(bin) \stackrel{def}{=} F\left(log_{coef}\frac{q \cdot \Delta_1(bin)}{\Delta_0(bin)}\right) \tag{5}$$

Что из себя представляет функция F является «коммерческой тайной RSA», но во-первых можно догадатся, во-вторых можно проверить свою догадку, имея доступ к СУБД.

Ваши предположения?

Группа

Группа – это ***max-ансамбль*** *категорий* вида:

$$G_i(bin) \stackrel{def}{=} max\big(C_{i_1}(bin), ..., C_{i_m}(bin)\big) \tag{6}$$

Группа может состоять из одной категории:

$$G_i(bin) = max\big(C_i(bin)\big) \equiv C_i(bin)$$

Группы собираются из категорий вручную на основе того или иного кейса мошенничества и на основании здравого смысла.

Появляется новый кейс мошенничества – следовательно появтяется одна (или более) новая группа.

Статичные категории

Существуют кейсы мошенничества, которые происходят крайне редко. Либо которые происходят в одном банке и ещё «не дошли» до банков поменьше.

Таким образом обучающая выборка мошенничества может выйти за пределы окны $T = 90$.

Что делать?

Статичные категории

Существуют кейсы мошенничества, которые происходят крайне редко. Либо которые происходят в одном банке и ещё «не дошли» до банков поменьше.

Таким образом обучающая выборка мошенничества может выйти за пределы окны $T = 90$.

Для решения этой проблемы задаются ***статичные и полустатичные категории***.

Статичная категория рассчитывается на определённой выборке и фиксируется в системе (не пересчитывается при повторном обучении системы).

Полустатичная категория использует статичные данные мошенничества, но свежие данные легитимных операций при расчёте категории.

Обнуляющая категория

Если мы хотим взвесить определённый риск и добавить группу, которая не уменьшит скоринг системы, мы можем в группу добавить обнуляющую категорию:

$$C_0(bin) \equiv 0 \tag{7}$$

Тогда группа никогда не будет уменьшать скоринг. Она либо его увеличит, либо оставит без изменений.

Приведите пример кейса мошенничества, в котором разумно использовать обнуляющую категорию?

Как обнуляющая категория связана со статическими и полустатическими категориями?

Неприведённый скоринг (Preliminary score)

Неприведённый скоринг (предварительный/ая скоринг/оценка, preliminary score) вектора признаков $\mathbf{x}$ – это функция, возвращающая число в диапазоне $(-\infty, +\infty)$, которое с помощью какой-либо *монотонно не убывающей* функции M можно привести к *отклику* (*априорной вероятности* мошенничества) p

$$preliminary\_score(\mathbf{x}) \in (-\infty, +\infty)$$

$$score(\mathbf{x}) = M\big(preliminary\_score(\mathbf{x})\big) = p \in [0, 1] \tag{8}$$

Иногда для удобства скоринг вычисляют умножением на 1000:

$$score(\mathbf{x}) = int\Big(1000 \cdot M\big(preliminary\_score(\mathbf{x})\big)\Big) = s \in [0, 1000] \tag{9}$$

Неприведённый скоринг (Preliminary score)

Неприведённый скоринг очень часто используют в ***суммирующих ансамблях***:

$$preliminary\_score(\mathbf{x}) \stackrel{def}{=} \sum_{i=1}^{m} w_i \cdot score_i(\mathbf{x}) \tag{10}$$

где функция $score_i(\mathbf{x})$ возвращает значение не в диапазоне $[0, 1]$, а в диапазоне $[-1, 1]$.

w_i – весовые коэфициенты.

Иногда вместо весовых коэфициентов используют монотонно-неубывающие функции W_i:

$$preliminary\_score(\mathbf{x}) \stackrel{def}{=} \sum_{i=1}^{m} W_i\big(score_i(\mathbf{x})\big) \tag{11}$$

Неприведённый скоринг в RSA

После расчёта всех групп $G_i(\mathbf{x})$:

$$preliminary\_score(\mathbf{x}) \stackrel{def}{=} \sum_{i=1}^{m} G_i(\mathbf{x}) \tag{12}$$

Нормализация

Нижний диапазон	Верхний диапазон	Процент	Суммарный процент
900	1000	0.25%	0.25%
800	900	0.25%	0.50%
700	800	0.50%	1.00%
600	700	2.00%	3.00%
500	600	2.00%	5.00%
400	500	5.00%	10.00%
300	400	10.00%	20.00%
200	300	10.00%	30.00%
100	200	20.00%	50.00%
0	100	50.00%	100.00%

Приведение скоринга. Функция *M*

Функция *M* задаётся на основе ***таблицы нормализации*** (см.предыдущий слайд). Все промежуточные значения скоринга вычисляются *линейной интерполяцией*.

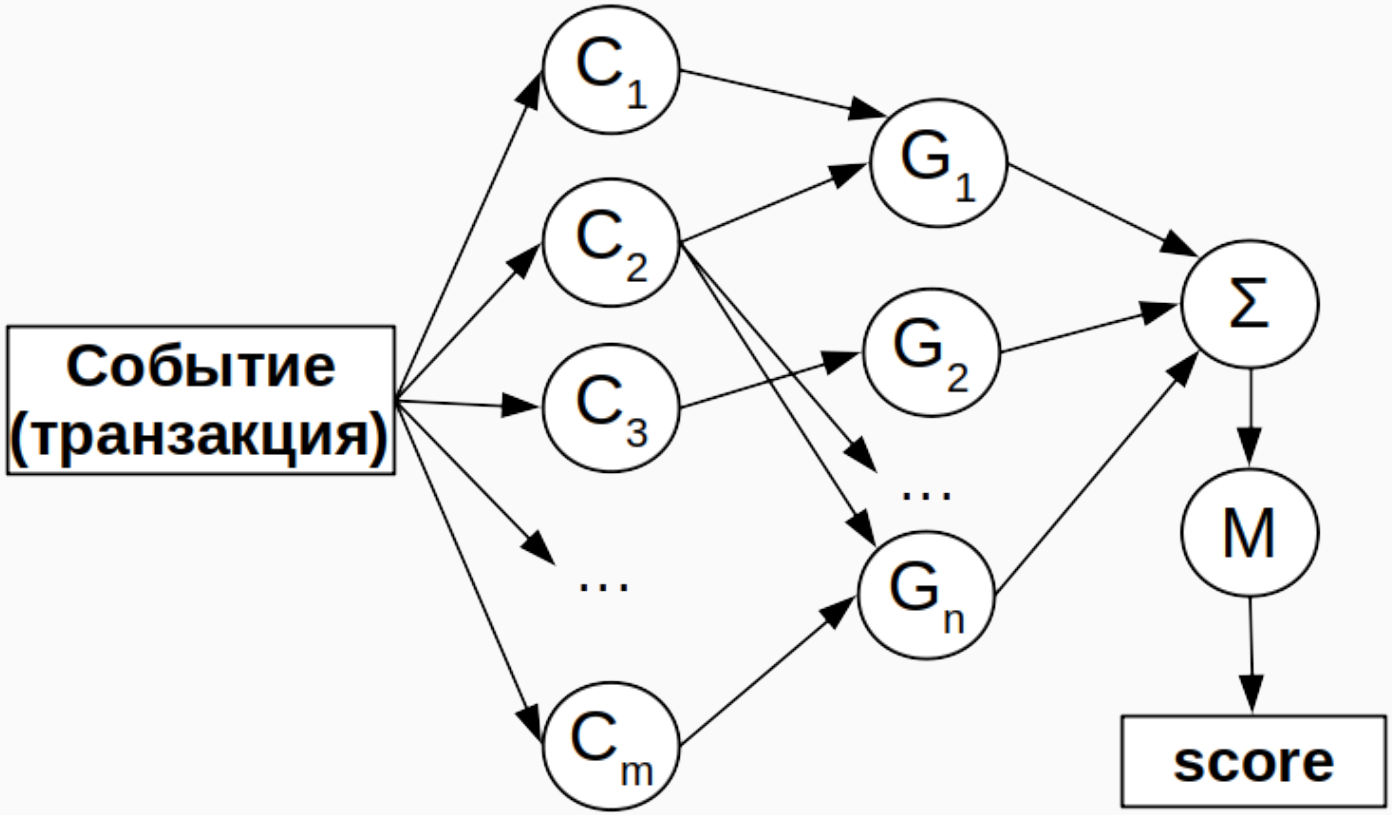
Замечание

Настоящая таблица нормализации задаётся с шагом 10, а не 100. То что на слайде №35 – это из маркетинговых материалов. Особенно важны шаги в диапазонах от 950 до 1000.

В чем достоинства и недостатки подобного подхода нормализации? Верно ли, что всегда будут транзакции с высоким скорингом? Верно ли что при «пожарах» потребуется уменьшать границы сработок большинства правил Rules Engine?

Risk Engine. Схема

Таким образом Risk Engine – это двуслойный ансамбль. Внешний слой – *суммирующий ансамбль* (расчёт неприведённого скоринга и функция *M*), внутренние слои – *max-ансамбли* (группы).



Кстати, в чем разница между многослойными ансамблями и нейронными сетями?

Термины RSA

Вопрос к залу. Who is who?

1. Разбиновка
2. Категория. R^1, R^2, R^3, R^4 категории
3. Группа
4. Неприведённый скоринг
5. Скоринг

Вопросы для самопроверки

Прочитайте, что такое «генетические алгоритмы»(genetic algorithm). В чём роль обратной связи в них?

Предложите структуру гена генетического алгоритма, создающий стратегии игры в крестики нолики.

Для желающих: реализуйте его (вместо какой-нибудь лабораторной). «стравите» разные гены друг с другом и путём «эволюции» получите самый сильный алгоритм.

Стал ли полученный алгоритм оптимальным? Сыграйте несколько партий. Сколько раз выиграли, сколько проиграли, сколько раз сыграли вничью?

Обратная связь и итеративное обучение

1. В «цепочке» (см. слайд №7) задана функция **s***core* с коэффициентами $(0.1, 0.2, 0.7)$. Каков будет скоринг, если $score_1 = 540$, $score_2 = 546$, $score_3 = 584$?
2. В «цепочке» задана функция **s***core* с коэффициентами $(0.2, 0.4, 0.8)$. Значения $score_1, score_2, score_3$ равны 345, 124 и 573. Почему **score** будет 412, а не 577 ?
3. Предположим, что $score_{i+1}$ сильно больше чем $score_i$. Что это значит? Действительно ли система работает нестабильно? Приведите контрпримеры.

RSA

- На слайде №12 не указаны действия обратной связи для ALLOW операции. Т.е. не производится никаких работ. Почему?
- На слайде №12 не указан процесс получения по жалобам клиентов. По каким решениям ФМ системы возможны жалобы? Приведите пример жалоб для ALLOW систем.
- Почему если произошла операция DENY на правило ”перевыпуск СИМ карт”, сотрудники коллцентра не могут разблокировать профиль позвонившим недовольным клиентам банка? Почему необходима либо более сложная аутентификация, чем телефонный звонок?

RSA

- Почему большинство транзакций имеют CUSTOM_MARK=Null ? (см. слайд №14)
- Почему все непромаркированные операции (CUSTOM_MARK=Null) спустя 10 и более дней считаются легитимными в обучающей выборке?

Risk Engine

1. Ответьте на вопросы о терминологии Risk Engine со слайда №39
2. Посмотрите на формулу (3). Зачем нужно брать логарифм? Почему бы просто не поделить (умножить) на *coef*?
3. Что будет, если убрать коэффициент q в формуле (3) ?
4. Если бы q задавали бы формулой:

$$q \stackrel{def}{=} \frac{\sum_{\forall bin} \Delta_1(bin)}{\sum_{\forall bin} \Delta_0(bin)}$$

   то как бы изменилась формула (3)?
5. Почему используется функция *max* для расчёта группы в формуле (6)? Почему не среднее арифметическое? Почему не *min*?
6. Зачем нужна функция *int* в формуле (9)? Что такое мантисса и порядок? Как числа хранятся в реляционной базе данных?

Risk Engine

7 Зачем нужны статичные и полустатичные категории? Приведите пример, когда они необходимы. Как вы думаете, в каких типах клиентах больше статичных категорий: в банковских системах для физических лиц, или в банковских системах для юридических лиц? Почему?

8 Почему бы формулу (7) не представить в виде:

$$C_0(bin) \equiv const$$

Объясните, почему для всех $const \neq 0$, категория $C_0(bin)$ не имеет практического смысла.

9 В Risk Engine используются max-ансамбли, но не используют min-ансамбли. Связано это с тем, что RSA разрабатывался в 80-е года, ещё до UEBA эпохи. Можете объяснить, как с помощью UEBA контрибутеров можно создавать min-ансамбли и как поправить схему на слайде №37?