Несбалансированные классы и поиск аномалий

МАКСИМОВСКАЯ
АНАСТАСИЯ

Несбалансированные классы

Метрики качества

Те, что мы уже знаем:

- F-beta score

$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}}.$$

- TPR, FPR, ROC-AUC
- Выставляем параметры (лучше macro-averaged)

Метрики качества

Fowlkes–Mallows index (G-measure)

$$FM = \sqrt{PPV \cdot TPR} = \sqrt{\frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}}$$

- Принимает значения от 0 до 1 (1 – максимальное при наилучшей бинарной классификации)
- Поскольку индекс прямо пропорционален количеству истинно положительных результатов, более высокий индекс означает большее сходство между двумя кластерами, используемыми для определения индекса.

Метрики качества

Jaccard Index

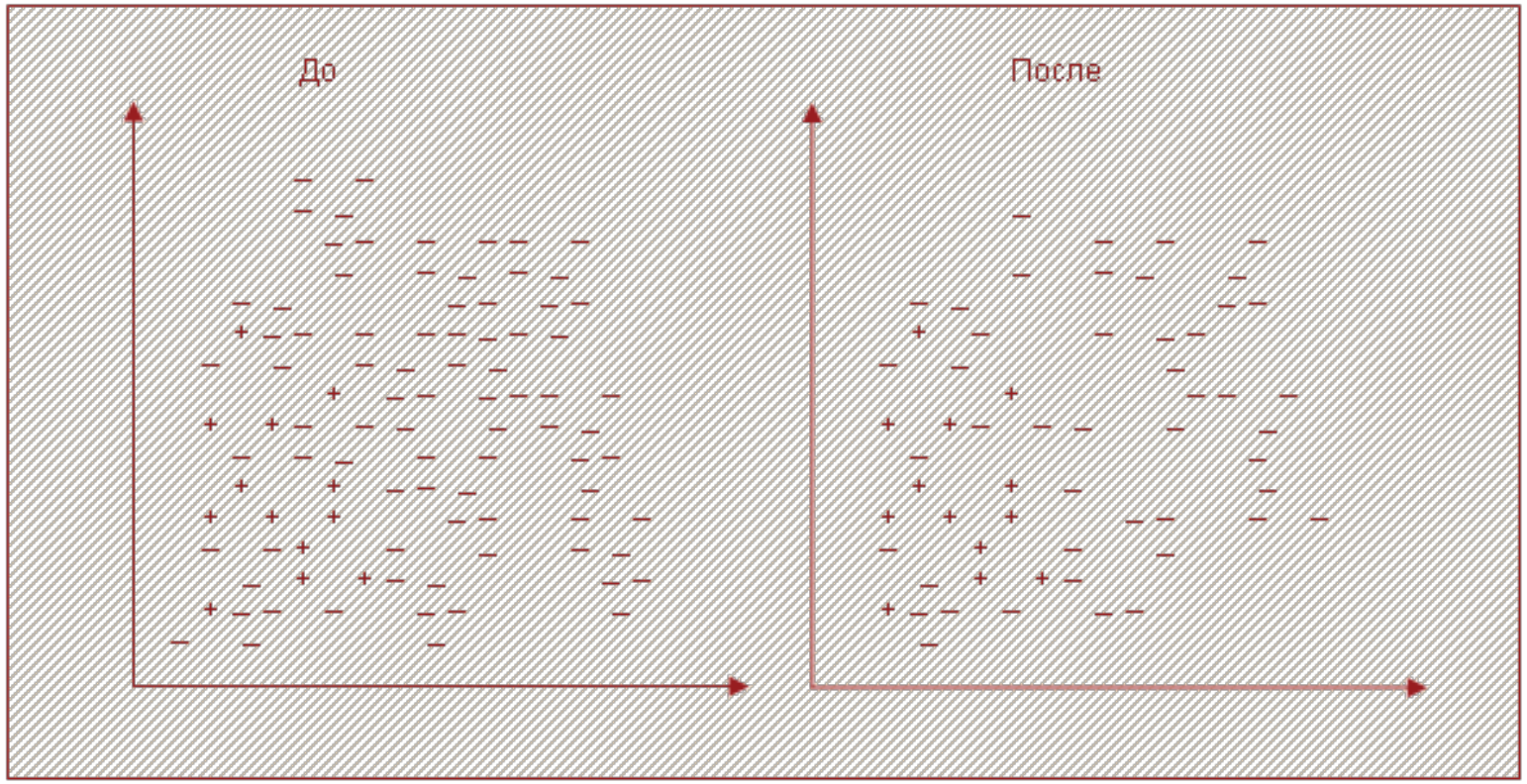
$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

$$J = \frac{TP}{TP + FP + FN}$$

- Также принимает значения от 0 до 1 и используется как мера схожести между двумя кластерами

Oversampling / Undersampling

Random Undersampling



Источник: URL

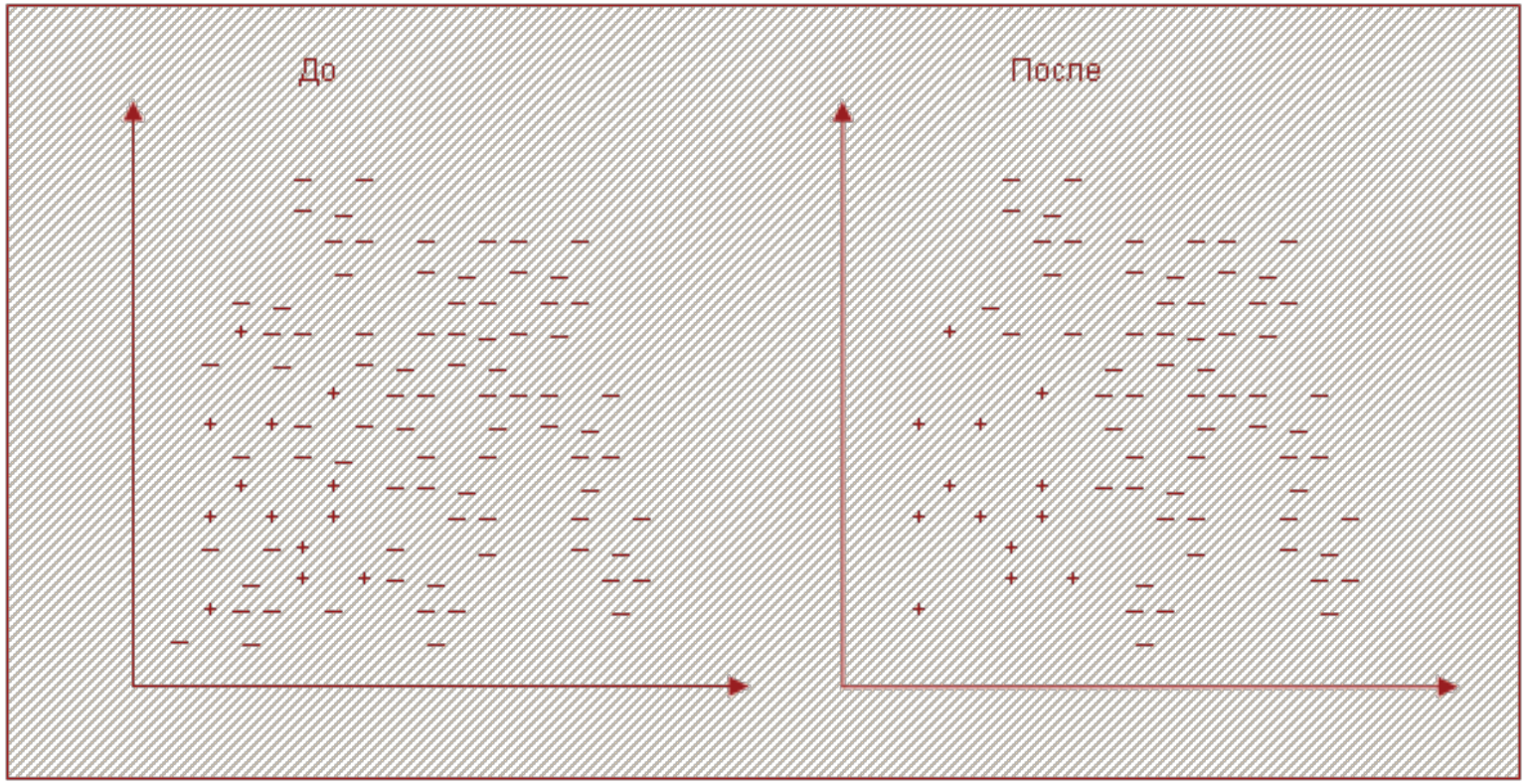
Tomek Links

Пусть примеры E_i и E_j принадлежат к различным классам, $d(E_i, E_j)$ – расстояние между указанными примерами. Пара (E_i, E_j) называется связью Томека, если не найдется ни одного примера E_l такого, что будет справедлива совокупность неравенств:

$$\begin{cases} d(E_i, E_l) < d(E_i, E_j) \\ d(E_j, E_l) < d(E_i, E_j)' \end{cases}$$

Согласно данному подходу, все мажоритарные записи, входящие в связи Томека, должны быть удалены из набора данных. Этот способ хорошо удаляет записи, которые можно рассматривать в качестве «зашумляющих»

Tomek Links

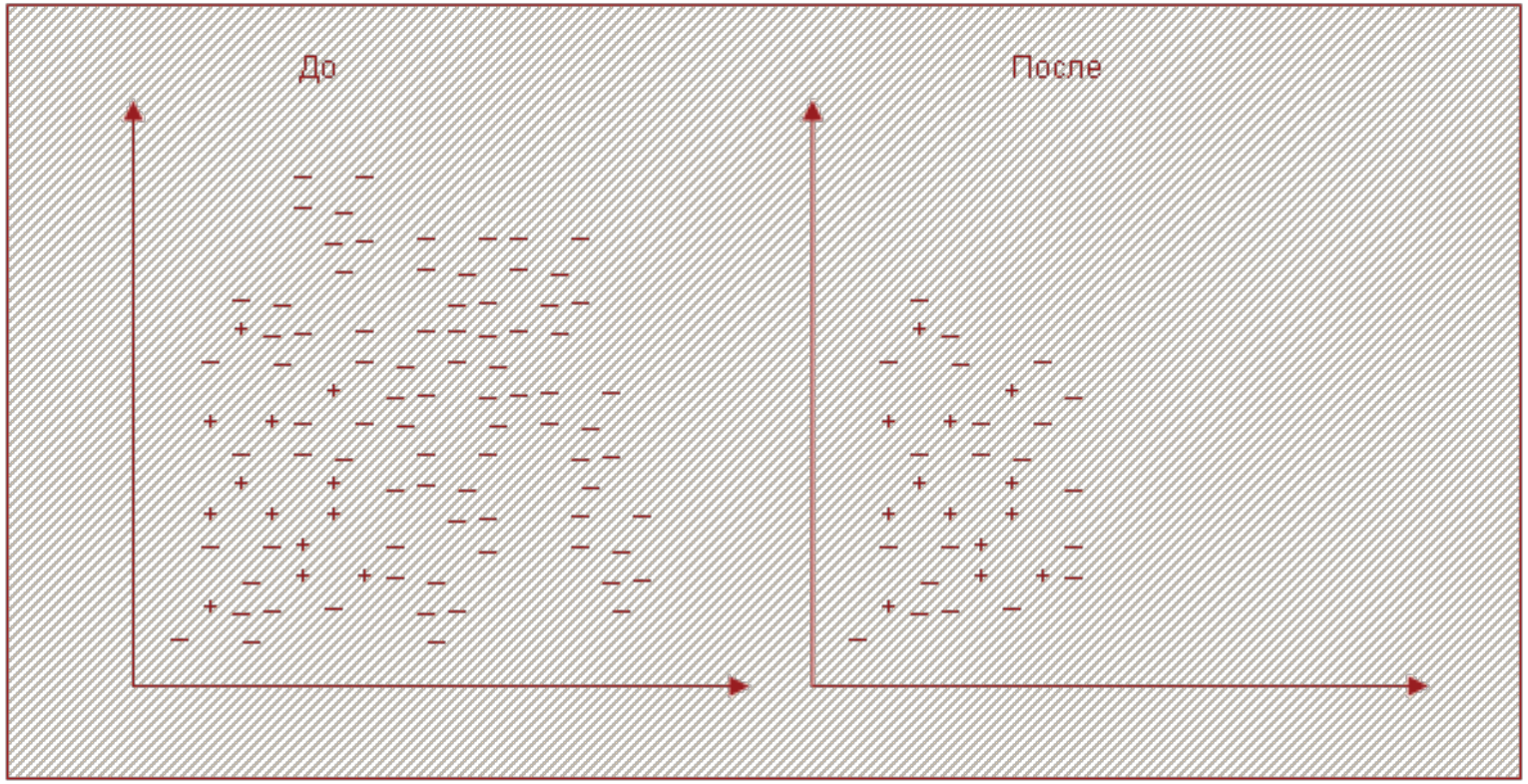


Condensed Nearest Neighbor

Пусть L – исходный набор данных. Из него выбираются все миноритарные примеры и (случайным образом) один мажоритарный. Обозначим это множество как S. Все примеры из L классифицируются по правилу одного ближайшего соседа (1-NN). Записи, получившие ошибочную метку, добавляются во множество S

Таким образом, мы будем учить классификатор находить отличие между похожими примерами, но принадлежащими к разным классам.

Condensed Nearest Neighbor



One-side sampling

Главная идея этой стратегии – это последовательное сочетание предыдущих двух, рассмотренных выше.

Для этого на первом шаге применяется правило сосредоточенного ближайшего соседа, а на втором – удаляются все мажоритарные примеры, участвующие в связях Томека.

Таким образом, удаляются большие «сгустки» мажоритарных примеров, а затем область пространства со скоплением миноритарных очищается от потенциальных шумовых эффектов.

Neighborhood cleaning rule

Эта стратегия также направлена на то, чтобы удалить те примеры, которые негативно влияют на исход классификации миноритарных наблюдений.

Для этого все примеры классифицируются по правилу трех ближайших соседей. Удаляются следующие мажоритарные примеры:

- получившие верную метку класса;
- являющиеся соседями миноритарных примеров, которые были неверно классифицированы.

Oversampling

Самый простой метод – это дублирование примеров миноритарного класса. В зависимости от того, какое соотношение классов необходимо, выбирается количество случайных записей для дублирования.

SMOTE

- Synthetic Minority Oversampling Technique

Алгоритм:

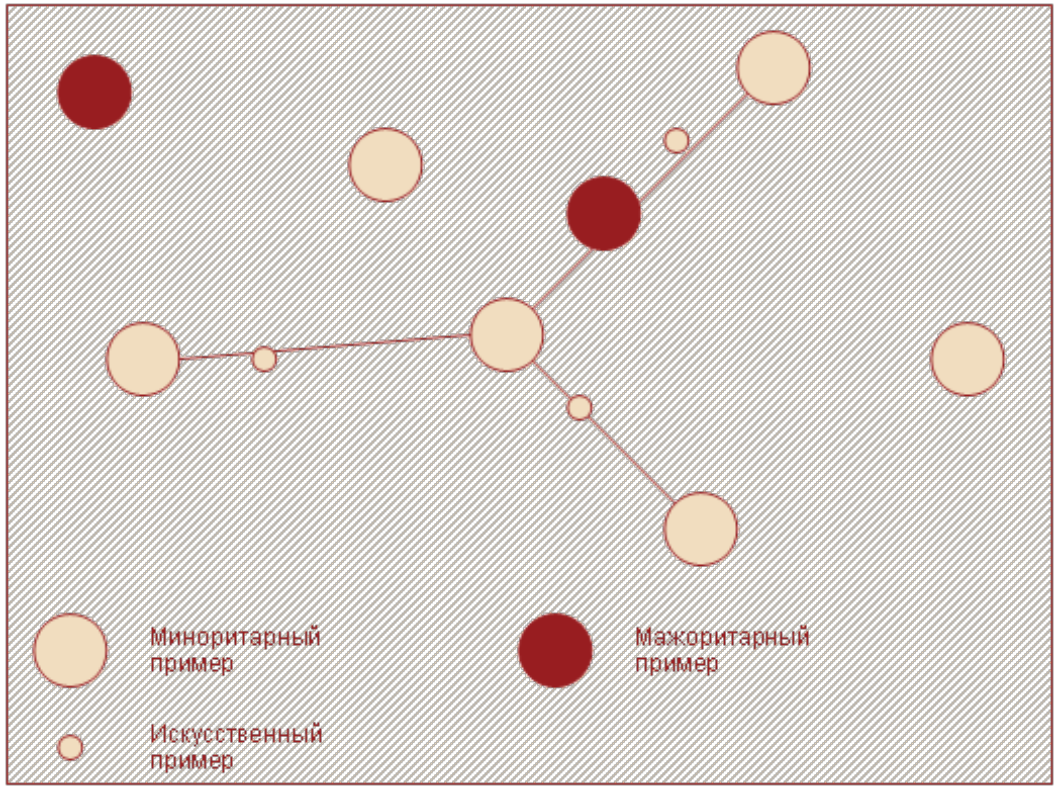
- Для создания новой записи находят разность $\mathrm{d} = X_b - X_a$, где X_a, X_b – векторы признаков «соседних» примеров a и b из миноритарного класса.
- Их находят, используя алгоритм ближайшего соседа (KNN). В данном случае необходимо и достаточно для примера b получить набор из k соседей, из которого в дальнейшем будет выбрана запись b. Остальные шаги алгоритма KNN не требуются.

SMOTE

- Далее из d путем умножения каждого его элемента на случайное число в интервале (0, 1) получают $\hat{d}$. Вектор признаков нового примера вычисляется путем сложения X_a и $\hat{d}$.

Алгоритм SMOTE позволяет задавать количество записей, которое необходимо искусственно сгенерировать. Степень сходства примеров a и b можно регулировать путем изменения значения k (числа ближайших соседей).

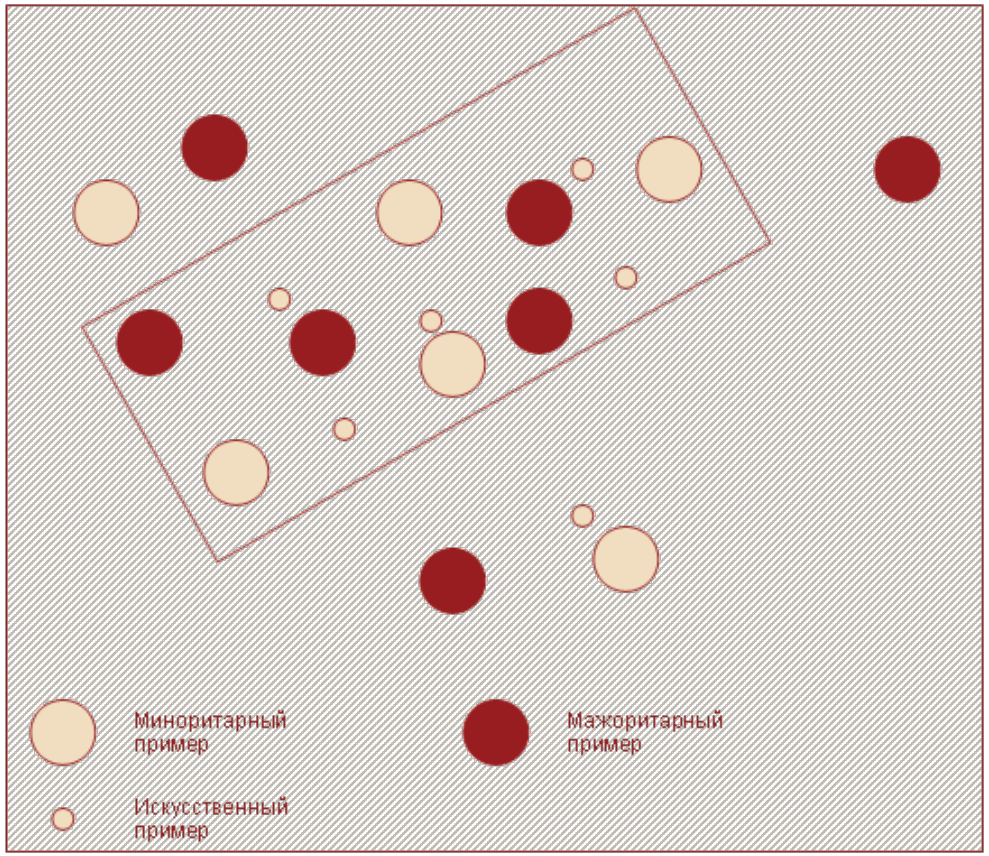
SMOTE


Миноритарный пример
Мажоритарный пример
Искусственный пример

SMOTE

- Данный подход имеет недостаток в том, что «вслепую» увеличивает плотность примерами в области слабо представленного класса
- В случае, если миноритарные примеры равномерно распределены среди мажоритарных и имеют низкую плотность, алгоритм SMOTE только сильнее перемешает классы.

SMOTE


Миноритарный пример
Мажоритарный пример
Искусственный пример

ASMO

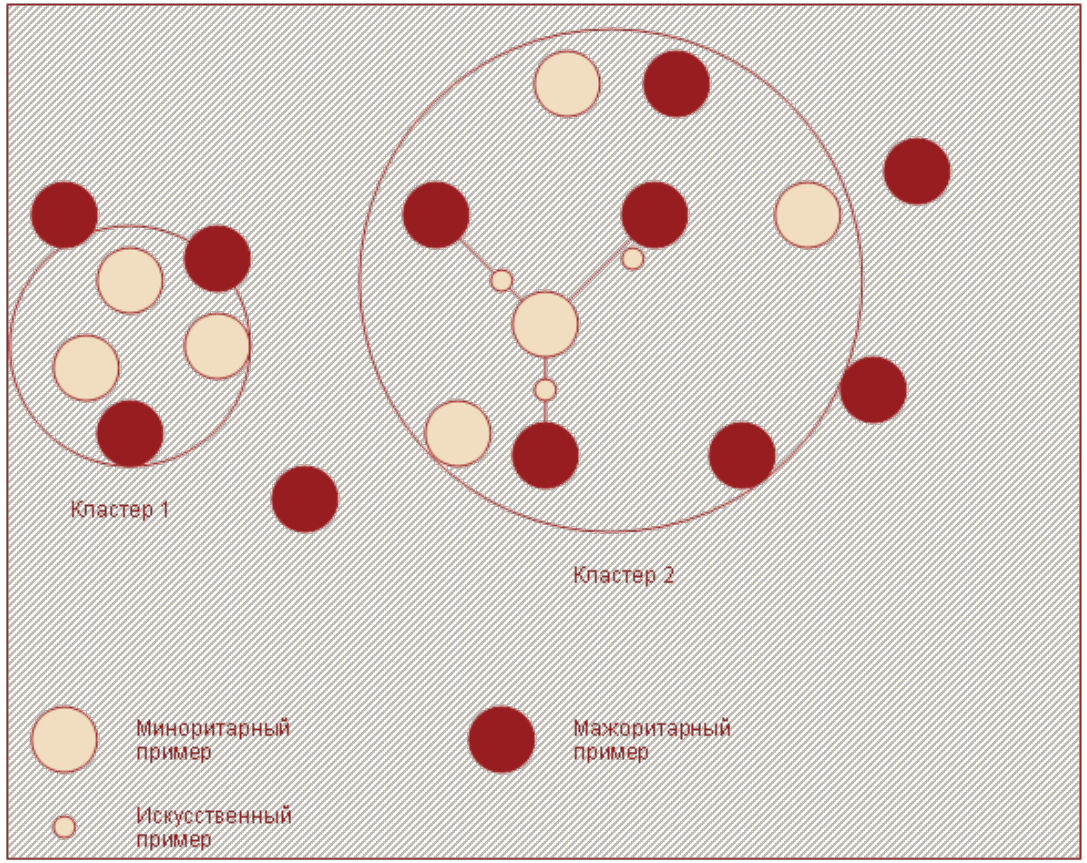
- Adaptive Synthetic Minority Oversampling
- Если для каждого i-ого примера миноритарного класса из k ближайших соседей g ($g \leq k$) принадлежит к мажоритарному, то набор данных считается «рассеянным».
- В этом случае используют алгоритм ASMO, иначе применяют SMOTE (как правило, g задают равным 20).

ASMO

Алгоритм:

1. Используя только примеры миноритарного класса, выделить несколько кластеров (например, алгоритмом k-means).
2. Сгенерировать искусственные записи в пределах отдельных кластеров на основе всех классов. Для каждого примера миноритарного класса находят m ближайших соседей, и на основе них (также как в SMOTE) создаются новые записи.

ASMO


Кластер 1
Кластер 2
Миноритарный пример
Искусственный пример
Мажоритарный пример

Одноклассовая классификация

Непараметрический подход

- Согласно одному из определений неотрицательная функция *p(x)* является плотностью распределения случайной величины ξ, если её значение в каждой точке равно пределу

$$p(x) = \lim_{h \to 0} \frac{1}{2h} \mathbb{P}(\xi \in [x - h, x + h]).$$

- Построим эмпирическую оценку плотности (h — ширина окна, регулирующая гладкость эмпирической плотности):

$$\hat{p}(x) = \frac{1}{2h} \frac{1}{\ell} \sum_{i=1}^{\ell} [|x - x_i| < h] = \frac{1}{\ell h} \sum_{i=1}^{\ell} \frac{1}{2} \left[\frac{|x - x_i|}{h} < 1 \right]$$

Непараметрический подход

Заменим индикатор того, что расстояние меньше ширины окна, на некоторую гладкую функцию *K(z)*:

$$\hat{p}(x) = \frac{1}{\ell h} \sum_{i=1}^{\ell} K\left(\frac{x - x_i}{h}\right)$$

Непараметрический подход

$K(z)$ – ядро, которое должно удовлетворять следующим требованиям:

- четность: $K(-z) = K(z)$;
- нормированность: $\int K(z)dz = 1$;
- неотрицательность: $K(z) \geq 0$;
- невозрастание при z > 0.

Пример: гауссово ядро

$$K(z) = (2\pi)^{-1/2} \exp(-0.5z^2)$$

Непараметрический подход

Оценку плотности легко обобщить на многомерный случай:

$$\hat{p}(x) = \frac{1}{\ell V(h)} \sum_{i=1}^{\ell} K\left(\frac{\rho(x, x_i)}{h}\right), \qquad V(h) = \int K\left(\frac{\rho(x,x_i)}{h}\right) dx$$

Число объектов, необходимое для качественной оценки плотности, растет экспоненциально по мере роста числа признаков.

Из-за этого непараметрические методы подходят только для обнаружение аномалий в маломерных пространствах.

Параметрический подход

Параметрический подход состоит в приближении плотности с помощью распределения $p(x \mid \theta)$ из некоторого семейства $\{p(x \mid \theta) \mid \theta \in \Theta\}$ с помощью метода максимального правдоподобия:

$$\sum_{i=1}^{\ell} \log p(x_i \mid \theta) \to \max_{\theta}$$

В качестве распределений могут выступать, например, нормальные или смеси нормальных.

Метрические методы

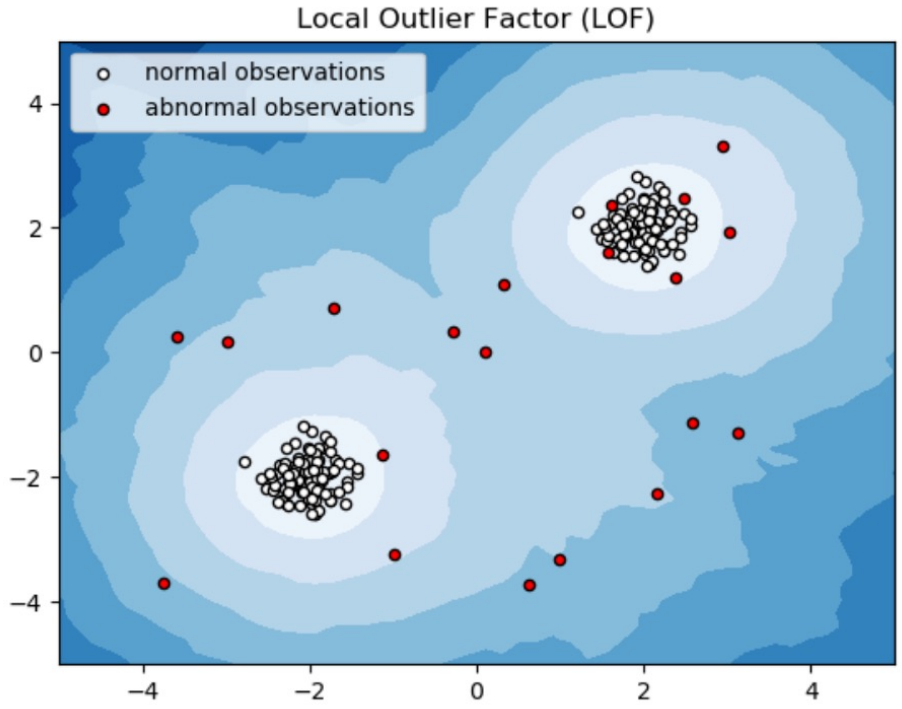
Простейший подход основан на выделении объектов, которые расположены от других существенно дальше, чем объекты в среднем удалены друг от друга. А именно, объект x объявим аномальным, если p или меньше процентов объектов имеют до него расстояние меньше ε:

$$\frac{1}{\ell}\sum_{i=1}^{\ell}[\rho(x,x_i)<\varepsilon]\leqslant p.$$

Пороги p и ε являются параметрами, которые должны настраиваться по известным примерам аномалий или исходя из априорных предположений.

Local outlier factor

Задаем плотность распределения в точке, используя k ближайших соседей, точки, плотность распределения в которых значительно меньше. чем v соседей – выбросы.



Одноклассовый SVM

Оптимизационная задача:

$$\begin{cases} \frac{1}{2}\|w\|^2 + \frac{1}{\nu\ell}\sum_{i=1}^{\ell}\xi_i - \rho \to \min\limits_{w,\xi,\rho} \\ \langle w, x_i\rangle \geqslant \rho - \xi_i, \quad i = 1, \ldots, \ell, \\ \xi_i \geqslant 0, \quad i = 1, \ldots, \ell. \end{cases}$$

Решающее правило (-1 означает выброс):

$$a(x) = \operatorname{sign}\left(\langle w, x\rangle - \rho\right)$$

Одноклассовый SVM

Ищем такую гиперплоскость, что:

- она отделяет как можно больше объектов выборки от нуля (чем меньше ν, тем больше объектов мы будем отделять);
- она имеет большой отступ;
- она при этом как можно сильнее отдалена от нуля (то есть ρ как можно большее значение).

Одноклассовый SVM

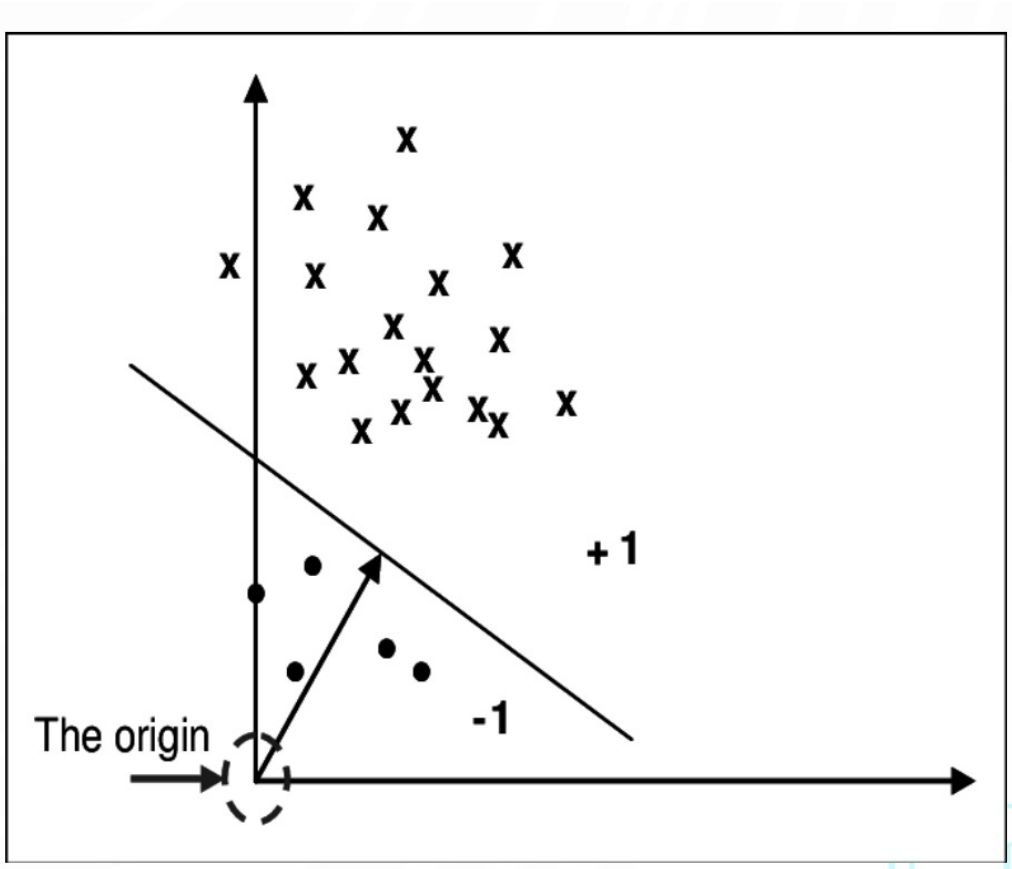
Двойственная задача:

$$\begin{cases} \frac{1}{2}\sum\limits_{i,j=1}^{\ell}\lambda_i\lambda_j K(x_i,x_j) \to \min\limits_{\lambda} \\ 0 \leqslant \lambda_i \leqslant \frac{1}{\nu\ell}, \quad i=1,\ldots,\ell, \\ \sum\limits_{i=1}^{\ell}\lambda_i = 1. \end{cases}$$

Модель для нее имеет вид:

$$a(x) = \operatorname{sign}\left(\sum_{i=1}^{\ell}\lambda_i K(x,x_i) - \rho\right)$$

Одноклассовый SVM



Isolation Forest

Идея: чем сильнее объект отличается от большинства, тем раньше он будет отделен от основной выборки случайными разбиениями => выбросы – объекты, которые оказались на небольшой глубине.

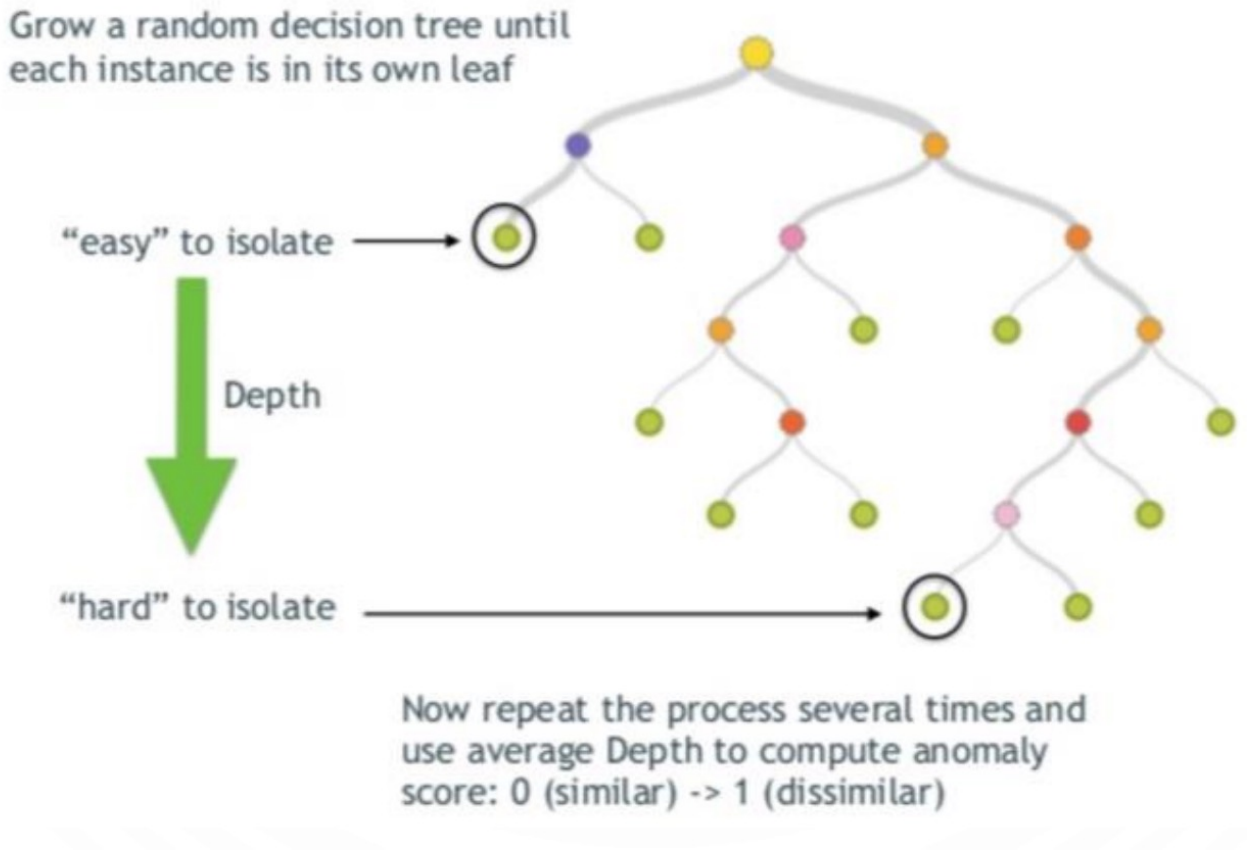
Isolation Forest

- На этапе обучения будем строить лес, состоящий из N деревьев.
- Каждое дерево будем строить стандартным жадным алгоритмом, но при этом признак и порог будем выбирать случайно.
- Строить дерево будем до тех пор, пока в вершине не окажется ровно один объект, либо пока не будет достигнута максимальная высота.
- Высоту дерева можно ограничить величиной $log_2 l$.

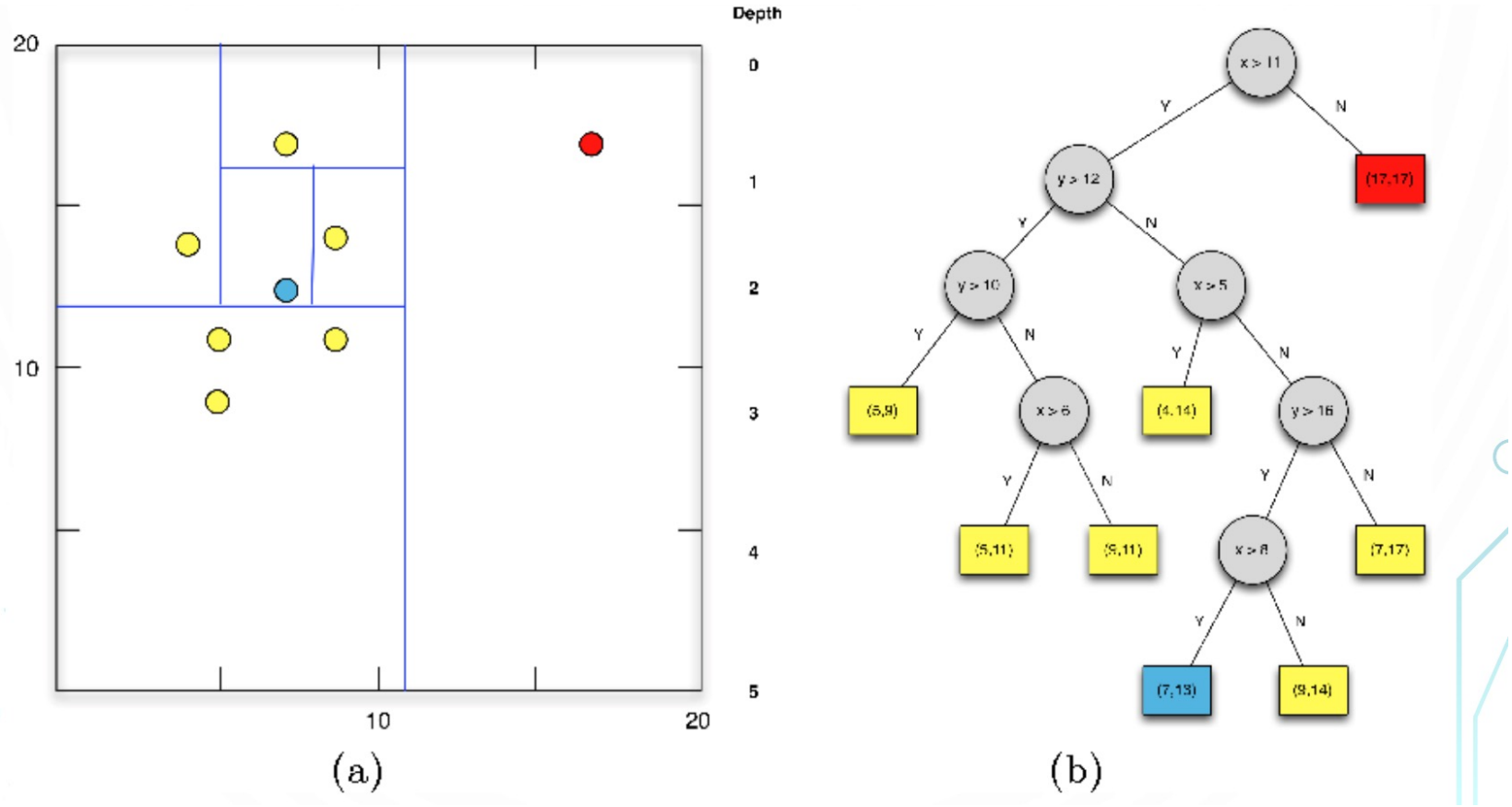
Isolation Forest

- Метод основан на предположении о том, что чем сильнее объект отличается от большинства, тем быстрее он будет отделен от основной выборки с помощью случайных разбиений.
- Соответственно, выбросами будем считать те объекты, которые оказались на небольшой глубине.

Isolation Forest



Isolation Forest



Isolation Forest

- Чтобы вычислить оценку аномальности объекта x, найдем расстояние от соответствующего ему листа до корня в каждом дереве.
- Если лист, в котором оказался объект, содержит только его, то в качестве оценки $h_n(x)$ от данного n-го дерева будем брать саму глубину k; если же в листе оказалось m объектов, то в качестве оценки возьмем величину $h_n(x) = k + c(m)$
- $c(m)$ — средняя длина пути от корня до листа в бинарном дереве поиска, которая вычисляется по формуле

$$c(m) = 2H(m-1) - 2\frac{m-1}{m} \qquad H(i) \approx \ln(i) + 0.5772156649$$

Isolation Forest

Оценку аномальности вычислим на основе средней глубины, нормированной на среднюю длину пути в дереве, построенном на выборке размера l:

$$a(x) = 2^{-\frac{\frac{1}{N}\sum_{n=1}^{N} h_n(x)}{c(\ell)}}$$

Для ускорения работы можно строить каждое дерево на подвыборке размера s; в этом случае во всех формулах выше нужно заменить l на s.

Поиск выбросов с помощью kNN

Вычисляем среднее расстояние от каждой точки до её ближайших k соседей, точки с наибольшим средним расстоянием – выбросы