Analisi e Descrizione Iniziali dello script
ITOA AnomalyDetection V1.1

Tommaso Fontana - Wuerth Phoenix

28/08/2019

Abstract

Descrizione ed Analisi degli script per il rilevamento automatico di anomalie.
Il valore di questo script e' quello di poter analizzare e identificare nei dati storici casi di comportamenti anomali del sistema informativo ed identificarne la potenziale causa radice.
Verranno analizzati vari metodi di AnomalyDetection, e la scelta di quale utilizzare sara' basata sulla natura dei dati interessati.

Descrizione del problema

Lo scopo di questo plugin e' quello di rilevare automaticamente le situazioni "anomale" su vari KPI riferiti a sistemi ed applicativi.

Il riconoscimento automatico di queste situazioni puo' essere fondamentale per riuscire ad essere reattivi ai cambiamenti dello stato del sistema informativo.

INPUT: I dati sorgente verranno letti da metriche eterogenee dal sistema InfluxDB quale parte della soluzione NetEye4.

OUTPUT: Il risultato dell'applicazione del machine learning (successivamente descritto) verra' inserito all'interno di measurement InfluxDB - come naming convention si utilizzera' *nomemeasurementsorgente_ML* - cosi' da poter essere rappresentato ed incrociato attraverso i pannelli del NetEye ITOA (grafana)

Descrizione Implementazione del Plugin

Descrizione Installazione e Configurazione

Lo script e' pensato principalmente per sistemi Unix-like ma e' generalizzabile anche a sistemi Windows e/o applicativi (i.e. Jboss, DB resultset, Tomcat, etc) .
Esistono due alternative di installazione.
Nel primo caso lo script non avra' bisogno di alcuna dipendenza poiche' autocontenuto nella cartella.
Questo e' ottenibile avendo tutto l'ambiente python gia' configurato al interno del plugin.
Questa comodita' e' pagata con la dimensione totale del plugins che sara' del'ordine dei 500Mb.
Alternativamente, per ottimizzare lo spazio si puo' usare un virtual-environment il quale non va a
modificare niente del sistema, ma necessita che python3 sia installato sulla macchina.
Questo porta la dimensione totale nel ordine dei 5Mb.

Poiche' in entrambi i casi il software e' praticamente auto-contenuto, la procedura di installazione sara' soltanto la copia della cartella sul sistema e la creazione del file di cron-job che avra' sintassi simile a:

```
*/5 * * * * /var/anomalydetection/anomalydetection
```

Aggiunta dello script al PATH

Si puo' creare un link simbolico:

```
$ sudo ln -s /var/anomalydetection/anomalydetection   /bin/anomalydetection
```

cosi che in tutto il sistema si possa chiamare *anomalydetection* come qualsiasi altro commando nel *PATH*.

Cio' permette di avere il file di cron che non deve avere hardcodato il path di installazione di *anomalydetection*:

```
*/5 * * * * anomalydetection
```

Configurazione accesso InfluxDB

Nella cartella ci sara' un file *db_settings.json* che conterra' le settings per connettersi al DB che avra' sintassi simile a:

```
{
    "database": "icinga2",
    "host": "127.0.0.1",
    "port": 8086,
    "username": "root",
    "password": "",
    "ssl": true,
    "verify_ssl": true,
    "timeout": 60,
    "retries": 3,
    "use_udp": false,
    "udp_port": 4444,
    "proxies": {},
    "path": ""
  }
```

Se necessario e' possibile generalizzare questa configurazione ad n-databases scrivendo una lista di oggetti.

Descrizione Aggiornamento

Lo script verrà sviluppato in un repository quindi una volta rilasciata la nuova versione sarà suffieicnete eseguire il seguente comando (o comunque una sua versione alternativa) nella cartella di installazione per aggiornare:

```
$ git pull
```

Descrizione Utilizzo

```
usage: anomalydetector [-h] [--detection-time-window DETECTION_TIME_WINDOW]
                [--normal-time-window NORMAL_TIME_WINDOW]
                [--min-n-of-values MIN_N_OF_VALUES] [--output-name OUTPUT_NAME]
                [--input-database-settings INPUT_DATABASE_SETTINGS]
                [--output-database-settings OUTPUT_DATABASE_SETTINGS]
                [--dry-run]
                HOST MEASUREMENT

AnomalyDetector is a free software developed by Tommaso Fontana for Wurth
Phoenix S.r.l. under GPL-2 License. Given the host and measurement it will
detect if any anomlaies occurred in the detection time window set. If there is
a plugin for the measurement it will be used, else the script will analyze the
data from all the Numeric fields.

positional arguments:
  HOST                  the host to select for the analysis
  MEASUREMENT           the measurement to select for the analysis

optional arguments:
  -h, --help            show this help message and exit
  --detection-time-window DETECTION_TIME_WINDOW, -dtw DETECTION_TIME_WINDOW
                        the time to consider for the Detection e.g. -w
                        1w2d3h4m5.6s, it defaults to 1w
  --normal-time-window NORMAL_TIME_WINDOW, -ntw NORMAL_TIME_WINDOW
                        the time to consider to profile normal data e.g. -w
                        1w2d3h4m5.6s, it defaults to 4w
  --min-n-of-values MIN_N_OF_VALUES, -min MIN_N_OF_VALUES
                        Minumum number of datapoint needed for the analysis,
                        it defaults to 1000
  --output-name OUTPUT_NAME, -on OUTPUT_NAME
                        The name of the measurement where the script will
                        write the results. It defaults to <measurement>_ML
  --input-database-settings INPUT_DATABASE_SETTINGS, -idbs INPUT_DATABASE_SETTINGS
                        Filepath to the JSON with the settings to conenct to
                        the InfluxDB to read the measurement, if not passed it
                        will default to the one in the same folder as this
                        executable.
  --output-database-settings OUTPUT_DATABASE_SETTINGS, -odbs OUTPUT_DATABASE_SETTINGS
                        Filepath to the JSON with the settings to conenct to
                        the InfluxDB to write the results, if not passed it
                        will default to the one in the same folder as this
                        executable.
  --dry-run, -dr        If this flag is enabled, the results will not be
                        written on the DB.
```

Cornercases

- Se lo script non puo' connettersi al DB ritornera' con **exit(2)**.
- Se lo script non trova la measurements nel DB ritornera' con **exit(1)**.
- Se il numero di datapoint per lo specifico host sara' meno di *min-n-of-values*, lo script ritornera' con **exit(1)**.
- Se uno dei path passati a *–input/output-database-settings* punta ad un file non esistente, ritornera' **exit(1)**.
- Se lo script finira' con successo ritornera' **exit(0)**.

Output

Lo script, se non viene passato il parametro –dry-run, scrivera' il risultato della analisi sul Database sul measurement selezionato con *–output-name* con schema *time host kpi_name anomaly_value*.

e.g. Dopo aver eseguito:

```
$ anomalydetector speedytouch.wuerth-phoenix.com disk
```

Sul DB di Output ci sara' una measurement *disk_ml* con valori:

```
time                host                              kpi_name     anomaly_value
----                ----                              --------     -------------
1556573376000000000 speedytouch.wuerth-phoenix.com inodes_total 0.46
1556573376000000000 speedytouch.wuerth-phoenix.com free          0.51
1556573376000000000 speedytouch.wuerth-phoenix.com all           0.65
1556573373000000000 speedytouch.wuerth-phoenix.com inodes_total 0.87
1556573373000000000 speedytouch.wuerth-phoenix.com free          0.82
1556573373000000000 speedytouch.wuerth-phoenix.com all           1.00
```

Dove il kpi_name *all* rappresenta lo stato di anomalia generale di tutto il measurement
Se il il measurement dove il risultato andra' scritto non esiste, questo verra' creato.

Descrizione Cartella

La cartella dello script avra' la seguente struttura:

```
drwxr-xr-x 5 user user 4,0K 29 giu 14.25 anomalydetection_venv
drwxr-xr-x 4 user user 4,0K 29 giu 14.21 doc
drwxr-xr-x 3 user user 4,0K  2 lug 23.01 src
drwxr-xr-x 3 user user 4,0K  2 lug 23.01 plugins
drwxr-xr-x 2 user user 4,0K 29 giu 13.51 tests
-rw-r--r-- 1 user user  281 29 giu 13.33 db_settings.json
-rw-r--r-- 1 user user  18K 29 giu 14.22 license
-rw-r--r-- 1 user user   95 29 giu 14.24 README.rst
-rwxr-xr-x 1 user user  102 29 giu 16.26 anomalydetectionlinux
-rwxr-xr-x 1 user user  102 29 giu 16.26 anomalydetectionwin
```

Legenda:

- **anomalydetection_venv** E' la cartella del virtual environment di python 3 gia' configurato.
- **doc** La cartella contenete la documentazione e gli esempi di utilizzo dello script.
- **src** La cartella con i sorgenti python dello script.
- **plugins** La cartella che conterra' i plugins per le measurements che andranno analizzate in modo particolare.
- **tests** La cartella conetenente i test da poter eseguire post installazione per poter testare che lo script funzioni correttamente.
- **db_settings.json** E' il file di configurazione per configurare come connettersi al DB.
- **license** La licenza GPL2 dello script.
- **README.rst** File di intro della repo e tutorial veloce su come installare, aggiornare ed utilizzare lo script.
- **anomalydetectionlinux** E' l'eseguibile per Linux.
- **anomalydetectionwin** E' l'eseguibile per Windows.

Descrizione del metodo di predizione

0.1 Descrizione del problema

Data una serie di valori $p_i = (t_i, u_i)$ su un intervallo di tempo Δt dove $u \in \mathbb{R}$ e' il valore della metrica considerata e $t \in \mathbb{N}$ e' il timestamp (epoch) della misurazione di u vogliamo sapere se, e quando, si sono verificate anomalie.

Una anomalia puo' essere definita come un evento "improbabile" rispetto al andamento normale.

0.2 Soluzioni proposte

Questa sezione andra' a presentare due approcci alternativi, con proprieta' molto diverse, per risolvere il problema.

Per mantenere un giusto bilancio tra accuratezza dei modelli ed il loro costo computazionale, entrambi i modelli verranno "traninati" sui dati della settimana precedente e del mese precedente.

Questo implica che entrmabi i modelli andranno "ri-trainati" una volta a settimana.

Per evitare discontinuita' delle anomalie, dovute al cambio di modello settimanalmente, se necessario, possiamo introdurre lentamente il nuovo modello combinando i risultati dei due modelli con una combinazione convessa.

$$f(x) = \alpha f^{t-1}(x) + (1-\alpha) f^{t}(x)$$

In modo che:

$$\alpha = 0 \Rightarrow f(x) = f^{t}(x) \quad , \quad \alpha = 1 \Rightarrow f(x) = f^{t-1}(x)$$

e che:

$$\forall \alpha \in [0,1] \subset \mathbb{R} \qquad \max(f^{t}(x), f^{t-1}(x)) \geq f(x) \geq \min(f^{t}(x), f^{t-1}(x))$$

Dove si puo' settare il valore di α in modo che decada lentamente nel corso della settimana.

$$\alpha = e^{\tau(t-t_0)}$$

Possiamo scegliere τ in modo che α valga 1 il lunedi, 0.5 il mercoledi, 0 il venerdi, cosi da avere i modelli, delle due settimane precedenti, che si "cedono il posto" in modo "morbido".

In seguito verano presentati 3 metodi principali per risolvere questo problema.

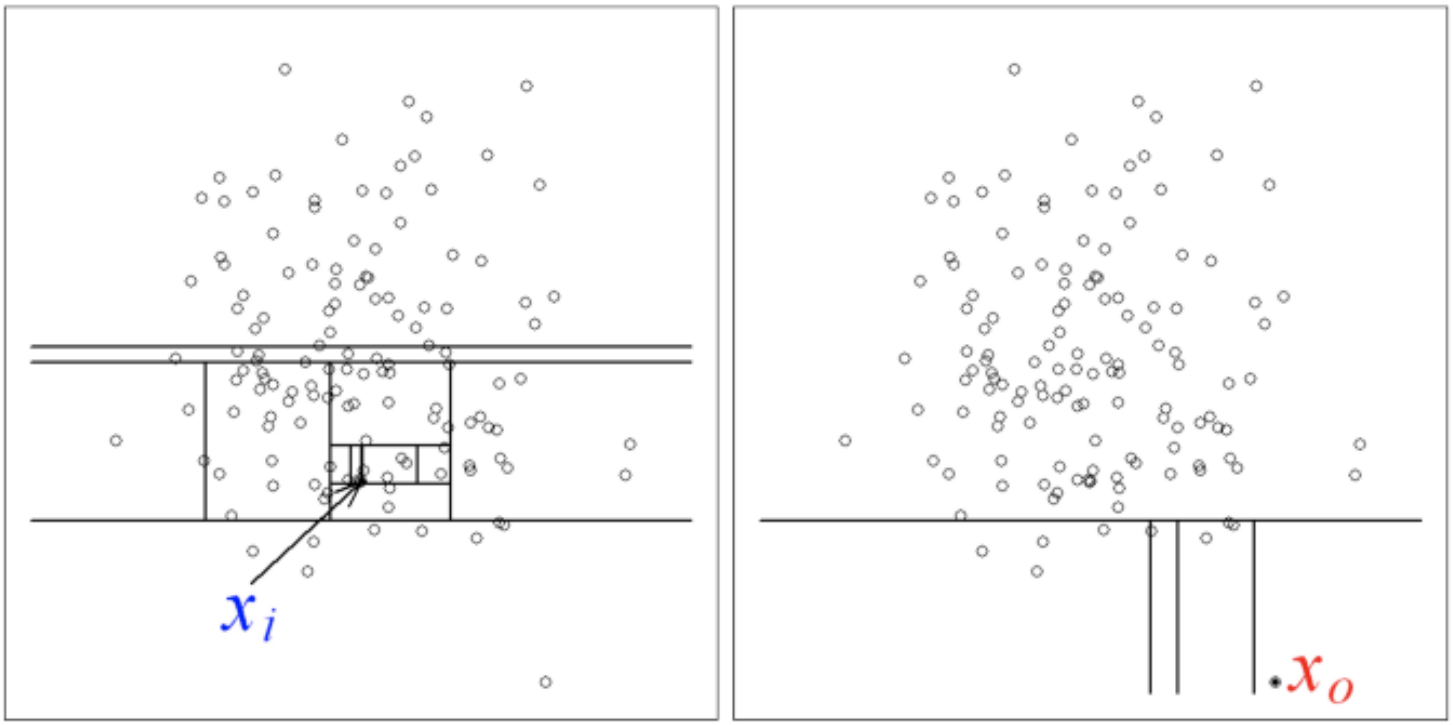
- Isolation Forest
- Clustering Using REpresentatives
- Input Reconstruction Method

0.2.1 Isolation Forest

L'idea principale, del metodo Isolation Forest, e' di isolare le anomalie senza aver bisogno di "profilare" i dati normali.
Quindi essere in grado di identificare le anomalie senza nessun tipo di ulteriore informazione in modo totalmente non supervisionato (quindi in modo automatico, senza bisogno di interventi umani).

Isolation Forest e' un metodo basato sugli alberi di decisione.
In particolare, questo funziona scegliendo casualmente delle "feature" e poi scegliendo un valore per dividere lo spazio di ricerca di due regioni distinte.
Questo valore ovviamente va scelto casualmente tra il minimo e massimo valore della "feature" selezionata.
Di conseguenza lo scopo dell'algoritmo e' di dividere lo spazio fino ad identificare una area che contenga solamente il punto da classificare.

Il principio su cui si basa questa idea, e' quello che le i dati "normali" sono piu' frequenti dei dati "anomali" ed in generale sono piu' distanti dagli altri dati.
Quindi partizionando casualmente lo spazio delle "feature", le anomalie dovrebbero essere piu' vicine alla radice del albero.
Di conseguenza avranno una lunghezza media, del percorso sull'albero, minore poiche' sono necessari meno partizionamenti.



Esempio di ricerca di, a sinistra un punto "normale" ed a destra un punto "anomalo". Si puo' notare che il caso anomalo richiede molti meno partizionamenti.

Dal metodo descritto precedentemente, lo score finale $s(x, n)$ e' definito come:

$$s(x, n) = 2^{-\frac{E[h(x)]}{c(n)}}$$

Dove:

- x e' il dato che vogliamo classificare.
- n e' il numero di nodi esterni.
- $E[x]$ Indica la media della variabile x.

- $h(x)$ e' la lunghezza del percorso della osservazione x.
- $c(x)$ e' la lunghezza media del percorso delle ricerche senza successo in un albero di ricerca binario.

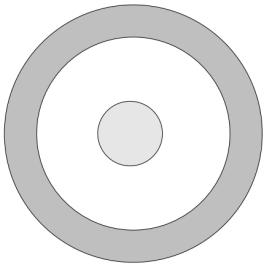
Quindi ne segue che $s(x, n) \in [0, 1] \subset \mathbb{R}$.
Di conseguenza il risultato puo' essere interpretato come:

- $s(x, n) \approx 1$ allora vi e' una anomalia.
- $s(x, n) \approx 0.5$ allora non e' possibile individuare anomalie **distinte**.
- $s(x, n) << 0.5$ allora e' una condizione normale.

Pro	Contro
Semplice ed Interpretabile	E' un algoritmo probabilistico
Veloce	Non c'e' un modo naturale per identificare quale "feature" e' la causa della anomalia

0.2.2 Clustering Using REpresentatives

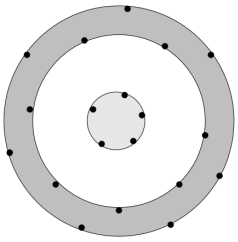
Nel caso la mole di dati da analizzare fosse significativa, L'algoritmo Clustering Using REpresentatives puo' essere una soluzione. E' un metodo di Clustering molto efficente, col quale e' possibile riconoscere le anomalie e sopratutto l'unica assunzione che fa sui dati e' che provengano da uno spazio euclideo (quindi che la distanza sia definita come $d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$).
In particolare, al contrario degli altri principali metodi di clustering, I cluster possono assumere forme ,che sarebbero patologiche per gli altri metodi, come gli anelli concentrici.



Esempio di cluster concentrici

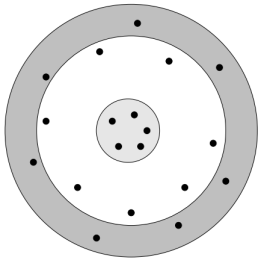
L'algoritmo CURE funziona come segue:

- Una piccola porzione dei dati va caricata e clusterizzata. Si suggerisce un metodo di clustering gerarchico dove i cluster sono uniti quando hanno una coppia di punti vicini.
- Una volta decisi i clusters, una piccola porzione dei punti appartenenti ad ogni cluster sono selezionati per essere i punti "rappresentativi". Questi punti dovrebbero essere scelti come i piu' distanti fra loro in ogni cluster, ma questo e' riconducibile ad un Max Diversity Problem che e' un problema **NP-Completo**. Quindi bisognera' ricorrere a qualche euristica.



Esempio di cluster concentrici con i punti rappresentativi evidenziati

- Infine si "allontanano" i punti rappresentativi di una frazione fissata della distanza tra il punto ed il centroide del cluster (Il punto medio dei punti appartenenti al cluster). Di solito questa frazione e' fissata al 20%. Questo step e' quello da cui nasce la richiesta di uno spazio euclideo poiche' altrimenti non potrebbe essere definito il concetto di linea che passa tra due punti.



Esempio di cluster concentrici con i punti rappresentativi evidenziati e spostati del 20%

0.2.3 Input reconstruction tecnique

L'idea e' di avere una funzione f che approssimi la funzione Identita', dovremo forzare del' "under-fitting", cioe' che il modello di approssimazione non sia abbastanza potente da approssimare completamente la funzione.

$$f_\theta(\bar{x}) \approx \bar{x} \quad \Rightarrow \quad f_\theta \approx I$$

Questa caratteristica forzera' l'approssimatore a "concentrarsi" sui valori "importanti" e quindi "imparera'" ad approssimare al meglio i dati "normali" rispetto a quelli "anomali".

L'analisi verra' fatta su un vettore di features $x \in \mathbb{R}^n$.
E' realistico assumere che queste features provengano da una distribuzione gaussiana multivariata.

$$\bar{x} \sim \mathcal{N}(\bar{\mu}, \bar{\sigma})$$

Successivamente, per migliorare la convergenza della rete, buona idea normalizzare i dati:

$$\frac{\bar{x} - \bar{\mu}}{\sqrt{\bar{\sigma}}} \sim \mathcal{N}(\bar{0}, \bar{1})$$

E quindi aggiungere i parametri $\bar{\mu}, \bar{\sigma}$ al modello.
E' possibile fare inferenza di entrambi usando i normali stimatori puntuali corretti in media quadratica.

$$\bar{\mu} = E[\bar{x}] = \frac{1}{n}\sum_i^n x_i$$

$$\bar{\sigma} = E[\bar{x}^2] - E[\bar{x}]^2 = \frac{1}{n}\left[\sum_i^n \bar{x_i}^2 - \left(\sum_i^n \bar{x_i}\right)^2\right]$$

Quindi essendo questa una regressione, la funzione di loss da minimizzare avra' forma:

$$L(\bar{x}, \bar{\theta}) = \left(f_{\bar{\theta}}\left(\frac{\bar{x} - \bar{\mu}}{\sqrt{\bar{\sigma}}}\right) - \bar{x}\right)^2$$

Infine una volta "trainata" la rete, per analizzare se un valore e' un anomalia o meno possiamo analizzare l'errore tra il dato e la ricostruzione.

La funzione e' trainata per ricostruire l'input, quindi e' realistico assumere che anche l'output sia distribuito normalmente.

$$f_{\bar{\theta}}\left(\frac{\bar{x} - \bar{\mu}}{\sqrt{\bar{\sigma}}}\right) \sim \mathcal{N}(\bar{0}, \bar{1})$$

Poiche' la differenza tra distribuzioni normali rimane normale, la funzione di loss e' assimilabile ad una distribuzione chi-quadro con n gradi di liberta':

$$L(\bar{x}, \bar{\theta}) = \left(f_{\bar{\theta}}\left(\frac{\bar{x} - \bar{\mu}}{\sqrt{\bar{\sigma}}}\right) - \bar{x}\right)^2 \sim \sum^n \left(\mathcal{N}(0,1) - \mathcal{N}(0,1)\right)^2 \sim \sum^n \mathcal{N}^2(0,1) \sim \chi^2(n)$$

Possiamo scegliere, un valore $z_{n,\alpha}$ tale per cui valga:

$$P(L(\bar{x}, \bar{\theta}) < z_{n,\alpha}) = \alpha$$

Quindi una volta scelto α il quale normalmente assume valori pari a $\{0.9, 0.95, 0.99\}$ che quindi nel nostro caso corrisponderebbero a considerare anomalie i valori che rispettivamente hanno probabilita' inferiore a $90\%, 95\%, 99\%$.

Quindi alla fine:

- Se $L(\bar{x}, \bar{\theta}) < z_{n,\alpha}$ allora il valore e' considerato normale.
- Se $L(\bar{x}, \bar{\theta}) \geq z_{n,\alpha}$ allora il valore e' considerato Anomalo.

Ulteriormente, potrebbe aver senso scegliere due valori $z_{n,\alpha}, z_{n,\beta}$ i cui parametri potrebbero essere $\alpha = 0.9 \quad , \quad \beta = 0.99$ in modo da avere due threshold, una critica ed una di warning:

- Se $L(\bar{x}, \bar{\theta}) < z_{n,\beta}$ allora il valore e' considerato normale.
- Se $L(\bar{x}, \bar{\theta}) \geq z_{n,\beta} \quad \wedge \quad L(\bar{x}, \bar{\theta}) < z_{n,\alpha}$ allora il valore e' considerato potenzialmente Anomalo.
- Se $L(\bar{x}, \bar{\theta}) \geq z_{n,\alpha}$ allora il valore e' considerato Anomalo.

Le threshold dell'esempio, nel caso vi siano 5 "features", saranno:

$$z_{5,0.9} = 9.236 \quad , \quad z_{5,0.99} = 15.086$$

.

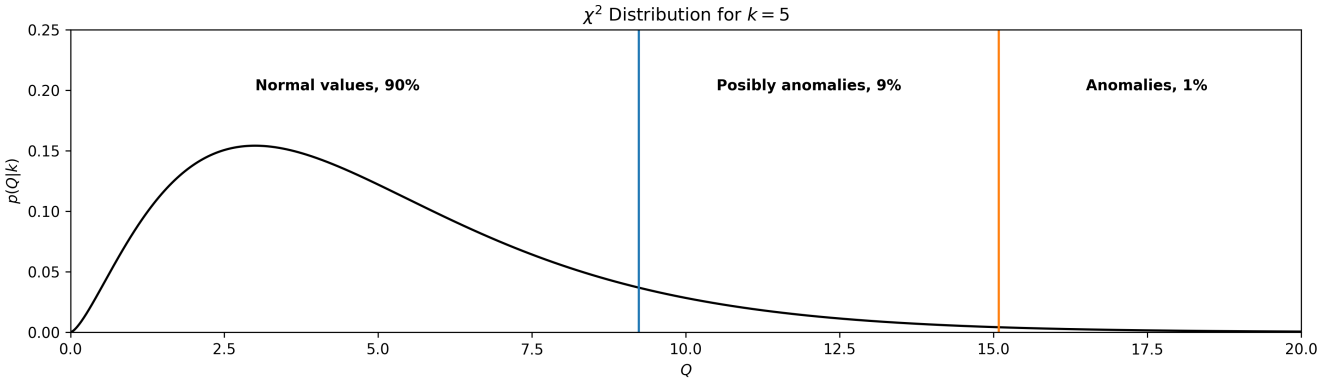


Figure 1: Visualizzazione delle due threhsold, nel caso dell' esempio, quindi con 5 "features".

Una volta identificato se il valore e' anomalo, e' facilmente identificale quali "feature" abbiano contribuito maggiormente a questa classificazione. E' sufficente trovare i le "feature" con errore maggiore.
Inoltre, come descritto sopra, l'errore della singola feature sara' distribuito normalmente, quindi possiamo applicare lo stesso metodo di classificazione, descritto prima, anche per la singola feature.

- Se $|f_{\bar{\theta}}(\bar{x}) - \bar{x}|_i < z_{0.9}$ il valore e' normale.
- Se $|f_{\bar{\theta}}(\bar{x}) - \bar{x}|_i \geq z_{0.9} \quad \wedge \quad |f_{\bar{\theta}}(\bar{x}) - \bar{x}|_i < z_{0.99}$ il valore e' potenzialmente anomalo.
- Se $|f_{\bar{\theta}}(\bar{x}) - \bar{x}|_i \geq z_{0.99}$ il valore e' anomalo.

Dove $|\quad|_i$ e' il valore assoluto delle componenti del vettore e selezione della i-esima "feature".

I valori delle threshold per questo esempio sono:

$$z_{0.9} = \pm 1.65 \quad , \quad z_{0.99} = \pm 2.58$$

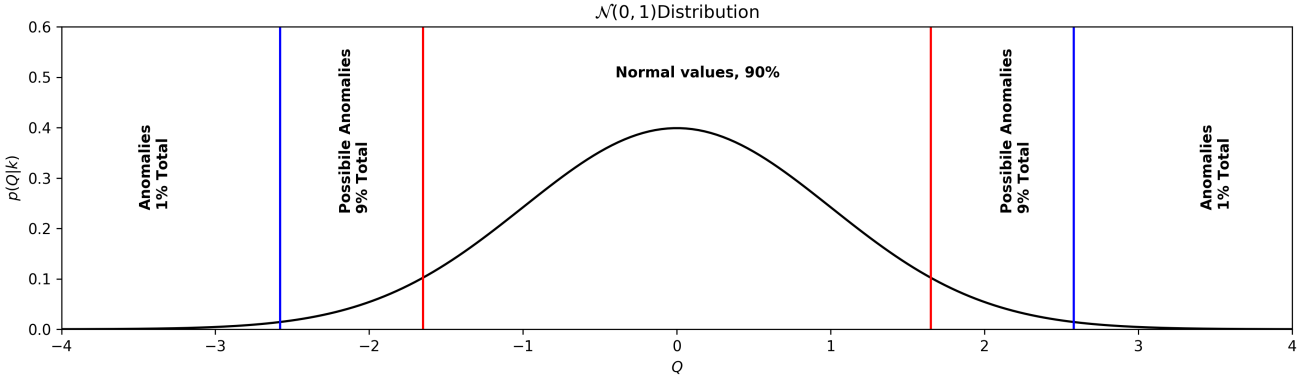


Figure 2: Visualizzazione delle due threhsold, nel caso dell' esempio, per una singola "feature"

Pro	Contro
Piu' sofisticato del IsolationForest	Richiede conoscenza dei metodi usati per poter configurarlo
E' un algoritmo deterministico	Puo' richiede training computazionalmente pesante
E' naturale estrarre ogni "feature" quanto contribuisce alla anomalia.	Puo' richiedere settaggio degli hyperparametri

La funzione di approssimazione puo' essere scelta tra vari metodi, ad esempio:

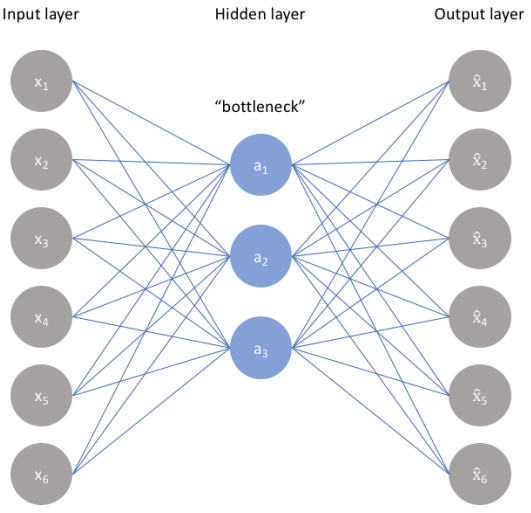
- **Radial Basis Function**, con un kernel gaussiano. Approssima la funzione come combinazione lineare di funzioni guassiane.
- **Principal Component Analysis**, metodo di riduzione di dimensionalita', veloce e molto semplice.
- **Foruier Series**, Metodo indicato per funzioni che presentano andamenti periodici.
- **Sviluppo in serie di Taylor / Laurent** si approssima la funzione con una serie di potenze.
- **QR decomposition**, con questo metodo e' possibile risolvere sistemi lineari di equazioni sovradeterminati, quindi e' possibile approssimare la funzione come combinazione lineare di un dato numero di funzion scelte a priori.etodo facile, ed interpretabile statisticamente. Quindi questo metodo puo' essere visto come una generalizzazione dei metodi elencati fino a qui.
- **Autoencoder**, tipo di rete neurale, e' un metodo sofisticato ma richiede potenza computazionale e tuning degli hyper-parametri.

In particolare il metodo suggerito e' di usare le Radial Basis Functions.

$$f(x) \approx \sum_{i=1}^{N} w_i e^{-(x-x_o)^2}$$

Questo metodo assicura l'assunzione che abbiamo fatto in generale che anche l'uscita sia $\mathcal{N}(\mu, \sigma)$ poiche' la approssimazione e' fatta da una somma di distribuzioni guassiane.

O alternativamente si puo' usare un Autoencoder.
Un Autoencoder e' un tipo di rete neurale che viene "trainata" per imparare la funzone identita'.
La particolairta' e' che il layer centrale sara' piu' "stretto" rispetto al' input ed output, cosi da formare una forma a "clessidra".
Questa strozzatura impone alla rete di imparare una rappresentazione latente dello spazio dei dati di dimensionalita' ridotta, quindi una sorta di "compressione" dei dati.



Esempio di Autoencoder