

Oracle Machine Learning 23c 新機能

Oracle Database 23c新機能セミナー

出口 龍之介

日本オラクル株式会社
2023/10/20



Oracle Machine Learning 23c 新機能

※今回の内容は新機能部分のみの記載となっております。
機能全般に関してご興味のある方は別途ご連絡ください。

既存のアルゴリズムの強化

- 言語処理系アルゴリズムの機能強化
- XGBoost(高精度で定評のあるアルゴリズム)関連のアップデート
- 時系列分析機能の強化

その他のアップデート

- Booleanデータ型のサポート
- テーブル最大4096列
- 自動データ準備の機能向上
- 一般化線形モデル(GLM)の適用範囲の拡大
- 異常検知に期待値最大化(EM)アルゴリズムが登場
- モデル生成に使用されたデータ取得クエリの記録



言語処理系アルゴリズムの機能強化

明示的セマンティック分析(ESA)でエンベッディングをサポート

- 概要
 - スコアリング結果を用いて分類やクラスタリングの品質を向上させることが可能に
 - 明示的セマンティック分析(ESA)…テキスト・ドキュメントの意味類似性の計算と明示的なトピックのモデリング（教師なし）、テキスト・ドキュメントの分類（教師あり）に使用されるアルゴリズム。
 - エンベッディング…単語やフレーズ、文書を実数のベクトルにマッピングすること。
 - ESAモデルのスコアリング時にエンベッディングを生成する。
 - 文書そのものの分散表現を行うDoc2Vecと似たアプローチ
- メリット・ユースケース
 - 文章の分類と類似性の評価
 - ESAのスコアリング結果を分類やクラスタリングなど他のモデルの入力に追加することで、精度の向上が期待できる



XGBoost(高精度で定評のあるアルゴリズム)関連のアップデート

相互作用制約・単調性制約の追加

XGBoost…回帰と分類のための高効率でスケーラブルな勾配ブースティングを使用した機械学習アルゴリズム。複数の弱い機械学習モデル（弱学習器）を直列に組み合わせることで、精度の向上を図る。

- **予測性能の向上**…どの変数が相互作用できるか指定することによりノイズが除去できる
 - XGBoostのアルゴリズム設定：interaction_constraintsパラメータで相互作用を許可する特徴量をリストで指定
- **モデルを解釈しやすくする**…モデルに「増加するトレンド」や「減少するトレンド」を制約として加える
 - XGBoostのアルゴリズム設定：increase_constraintsパラメータ、decrease_constraintsパラメータで増加/減少制約に従う特徴量を指定



XGBoost(高精度で定評のあるアルゴリズム)関連のアップデート

相互作用制約・単調性制約の追加

```
DECLARE
    v_setlst DBMS_DATA_MINING.SETTING_LIST;
BEGIN
    v_setlst('ALGO_NAME')    := 'ALGO_XGBOOST';
    v_setlst('PREP_AUTO')    := 'ON';
    v_setlst('max_depth')   := '2';
    v_setlst('eta')          := '1';
    v_setlst('num_round')   := '100';
    v_setlst('interaction_constraints') := '[[YRS_RESIDENCE, OCCUPATION],
                                              [OCCUPATION, Y_BOX_GAMES],
                                              [BULK_PACK_DISKETTES,
                                               BOOKKEEPING_APPLICATION]]';

    DBMS_DATA_MINING.CREATE_MODEL2(
        MODEL_NAME           => 'XGB_CLASS_MODEL_INTERACTIONS',
        MINING_FUNCTION     => 'CLASSIFICATION',
        DATA_QUERY          => 'SELECT * FROM TRAIN_DATA_CLAS',
        SET_LIST             => v_setlst,
        CASE_ID_COLUMN_NAME => 'CUST_ID',
        TARGET_COLUMN_NAME   => 'AFFINITY_CARD');

    DBMS_OUTPUT.PUT_LINE('Created model: XGB_CLASS_MODEL_INTERACTIONS');
END;
/
```



XGBoost(高精度で定評のあるアルゴリズム)関連のアップデート

故障予知(生存分析)のためのモデルAFTのサポート

生存分析的アプローチで故障予知などが可能に

生存分析…生物における死や機械システムにおける故障など、1つ(以上)の事象が発生するまでの経過時間を調べる

AFTモデル(Accelerated Failure Time : 加速故障時間)…Cox比例ハザードモデル*とともに、生存分析でよく使われるモデルの1つ。特に製品や装置の寿命予測で使われる。故障率が時間とともに増加すると仮定し計算する。

- XGBoostのアルゴリズム設定 : objectiveパラメータをsurvival:aftと指定

*サポート済

番号	生存期間の下限	年数	稼働状態	...	生存期間の上限
1	235	10	2	...	NULL
2	444	12	2	...	444
3	806	9	1	...	NULL



番号	予測生存期間
1	1011
2	477
3	1659

XGBoost(高精度で定評のあるアルゴリズム)関連のアップデート

故障予知(生存分析)のためのモデルAFTのサポート

```
DECLARE
    v_setlst DBMS_DATA_MINING.SETTING_LIST;
BEGIN
    v_setlst('ALGO_NAME') := 'ALGO_XGBOOST';
    v_setlst('max_depth') := '6';
    v_setlst('eval_metric') := 'aft-nloglik';
    v_setlst('num_round') := '100';
    v_setlst('objective') := 'survival:aft';
    v_setlst('aft_right_bound_column_name') := 'rbound';
    v_setlst('aft_loss_distribution') := 'normal';
    v_setlst('aft_loss_distribution_scale') := '1.20';
    v_setlst('eta') := '0.05';
    v_setlst('lambda') := '0.01';
    v_setlst('alpha') := '0.02';
    v_setlst('tree_method') := 'hist';

    DBMS_DATA_MINING.CREATE_MODEL2(
        MODEL_NAME      => 'XGB_SURVIVAL_MODEL',
        MINING_FUNCTION => 'REGRESSION',
        DATA_QUERY      => 'SELECT * FROM SURVIVAL_DATA',
        TARGET_COLUMN_NAME => 'LBOUND',
        CASE_ID_COLUMN_NAME => NULL,
        SET_LIST        => v_setlst);
END;
/
```



時系列分析機能の強化

自動モデルタイプ選択/複数の時系列予測

自動モデルタイプ選択

周期性や上昇傾向/下降傾向のあるデータに対する予測が容易になった

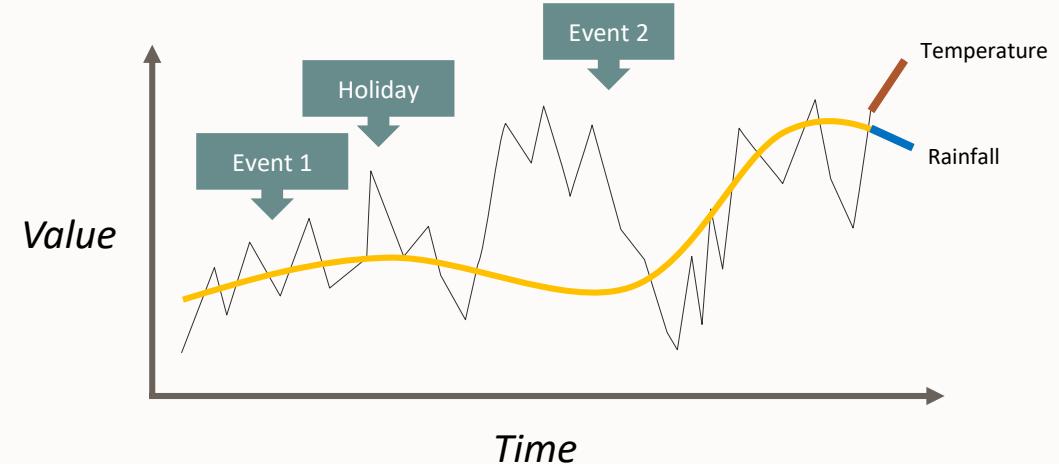
時系列予測(指数平滑法)のモデル作成時に、モデル・タイプ(EXSM_MODEL)を指定しない場合、アルゴリズムが最適なモデル・タイプを自動的に決定

複数の時系列予測

1つまたは複数の時系列に対するバックキャスト(過去のデータに対してモデルが生成した推定値)と予測を生成

結果は他のMLアルゴリズムの入力として使用できる

(例) 降雨量をターゲットにしつつ、温度と湿度の予測も行うことができる。



Timestamp	Weekday	...	Temp	DM\$Temp	DM\$Rainfall	Rainfall
<ts-1>	0		26	Backcasts	Backcasts	3.0
<ts-2>	1		30	0
<ts-3>	2		28	0
...
<ts-n+1>				Forecasts	Forecasts	予測対象



その他のOML関連のアップデート

- **一般化線形モデル(GLM)の適用範囲の拡大(リンク関数の追加)**
これまでLogitのみだったがProbit, Cloglog, Cauchitが追加
データの確率分布によってリンク関数を選択することでモデルの精度が向上する
- **OMLのBoolean型のサポート**
- **テーブル最大4096列**
- **異常検出アルゴリズムに期待値最大化(Expectation Maximization)が追加**
OML4SQLがサポートする異常検出アルゴリズム : MSET-SPRT, SVM, **Expectation Maximization**
- **モデルのメタデータへの入力データソースの追加**
*_MINING_MODELSビューのbuild_source列に、モデル生成に使用されたデータ取得クエリが記録される

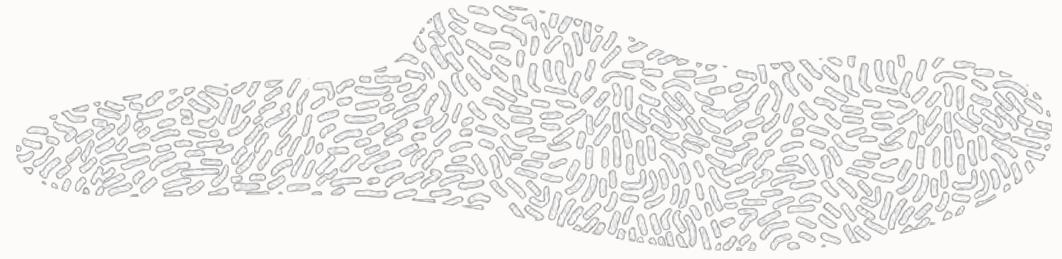


Oracle Spatial 23c 新機能



Oracle Spatial 23c 新機能

※今回の内容は新機能部分のみの記載となっております。
機能全般に関してご興味のある方は別途ご連絡ください。



SDO_GEOMETRY作成がより簡単に

- ・緯度経度ポイント・データの使いやすい構文
- ・ジオメトリタイプと座標系の事前定義済み定数の利用

空間索引作成がより簡単に

- ・ジオメトリ・オブジェクトのメタデータ登録の自動化



RasterデータのREST APIサポート

- ・データアクセス
- ・処理
- ・インポート/エクスポート
- ・仮想モザイク



点群の**差分検出**

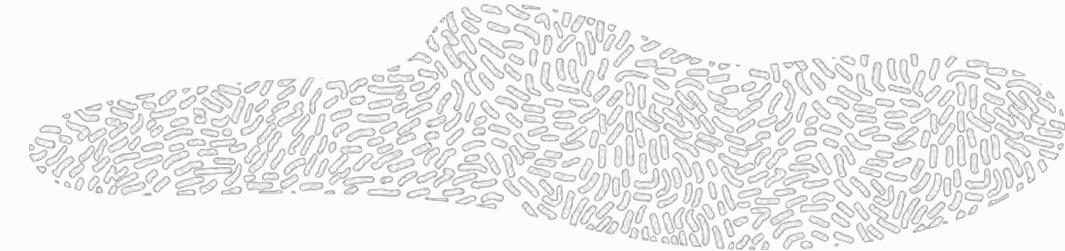
点群の**断面計算**

3Dメッシュ



緯度経度ポイント・データの使いやすい構文

より簡単な緯度経度データ作成が可能に



従来

```
INSERT INTO t1 VALUES
(sdo_geometry(2001, 4326,
sdo_point_type(-73.45, 45.2, null),
null, null));
```

23c

```
-- コンストラクタを使用して緯度経度データをINSERT
INSERT INTO t1 VALUES
(sdo_geometry(-73.45, 45.2));
```



SDO_GTYPEおよびSDO_SRID値の事前定義済み定数の利用

空間クエリの簡素化



- 概要

- SDO_GEOMETRYコンストラクタおよび空間クエリで、SDO_GTYPEおよびSDO_SRID値に事前定義済みの定数を使用することが可能になりました
- 従来はSDO_GTYPEやSDO_SRIDに指定された値を入力する必要がありましたが、分かりやすい定数を使うことができるようになります

従来

```
INSERT INTO t1 VALUES  
(SDO_Geometry(2003, 4326, null,  
SDO_ELEM_INFO_ARRAY(1,1003,3),  
SDO_ORDINATE_ARRAY(-100.1,20.2,-  
90.2,23.4)));
```

23c

```
INSERT INTO t1 VALUES  
(SDO_Geometry(SDO_POLYGON2D,SDO_LONLAT,  
null, SDO_ELEM_INFO_ARRAY(1,1003,3),  
SDO_ORDINATE_ARRAY(-100.1,20.2,-  
90.2,23.4)));
```





USER_SDO_GEOM_METADATAビューの自動更新

ジオメトリ・オブジェクトのメタデータ登録の自動化



- 概要

- 空間列に空間索引を作成する場合、その表の空間メタデータが作成されていなければ、USER_SDO_GEOM_METADATAビューに自動で誤差許容値0.05で空間メタデータが登録されます
- 空間メタデータを手動で挿入することはオプションになりました

```
INSERT INTO points_table VALUES(1, SDO_GEOGRAPHY(-73.45, 45.2));
COMMIT;
```

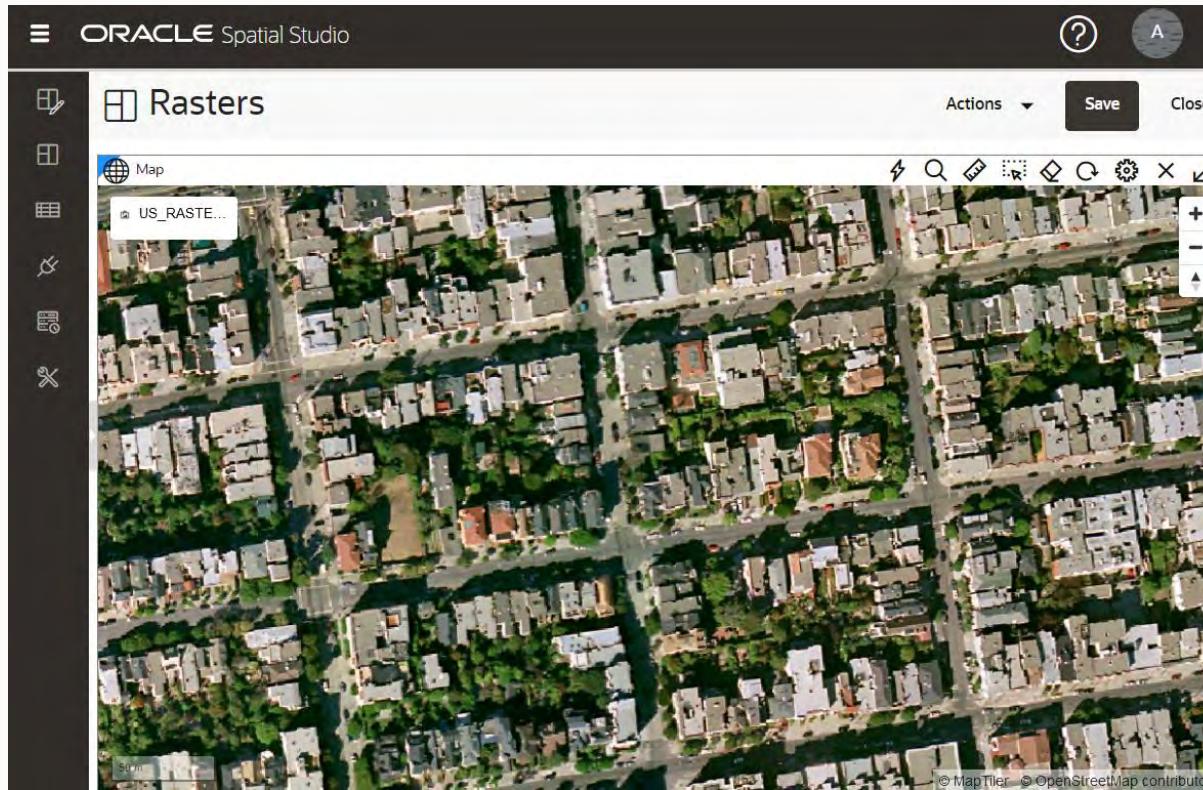
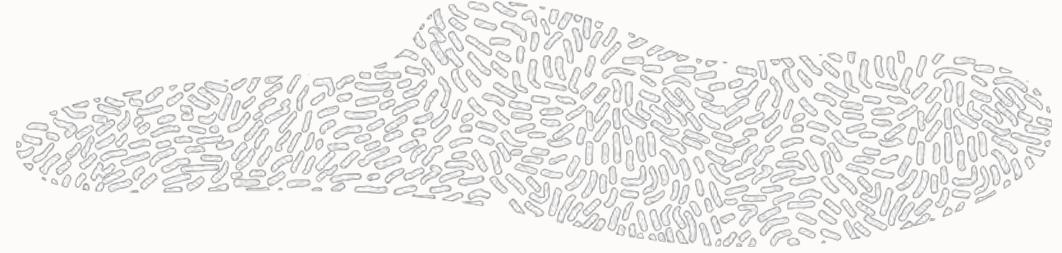
```
CREATE INDEX pt_sidx ON points_table (geometry) INDEXTYPE IS MDSYS.SPATIAL_INDEX_V2;
```

~~-- SDO_Metadata~~

~~INSERT INTO USER_SDO_GEOM_METADATA ...~~



RasterデータのREST APIサポート



```
# Example: List all resources in a dataset
curl https://<host>:<port>/oraclespatial/georaster/v1/ds1
```

任意の HTTP クライアントからラスター機能に
RESTアクセス可能に

- GeoRaster リソースのリスト
- メタデータ/データの表示・更新
- インポート/エクスポート
- 仮想モザイクの作成・管理

RESTエンドポイント

https://docs.oracle.com/cd/F82042_01/ndmcr/rest-endpoints.html





2つの点群の差を点群として返すPC_DIFFERENCE

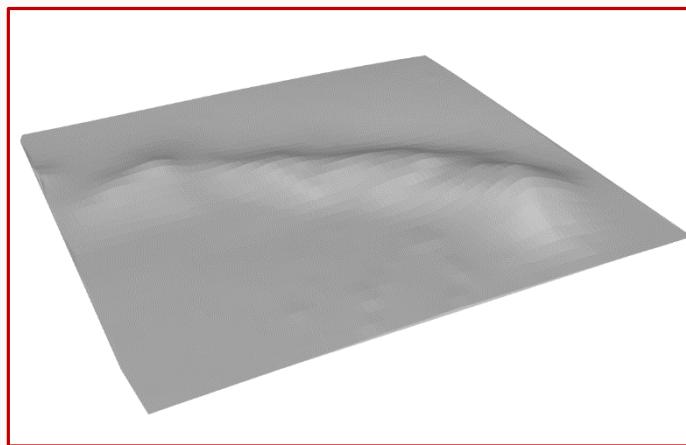


- 概要
 - Oracle Spatialの点群機能SDO_PC_PKGパッケージに、2つの点群の比較を行うPC_DIFFERENCEプロシージャが追加
 - 点群Aにある点のうち、点群Bに地理的に近接していない点の算出が可能
 - 以前はそこになかった車などの実際の点群の変化
 - レーザーが届かない障害物による変化
- ユースケース
 - 火災、洪水、地滑り、地震などの自然災害による損害の評価
 - 森林キャノピー（森林の上層部）の変化のモデル化
 - 開発・インフラプロジェクトの経年変化の測定

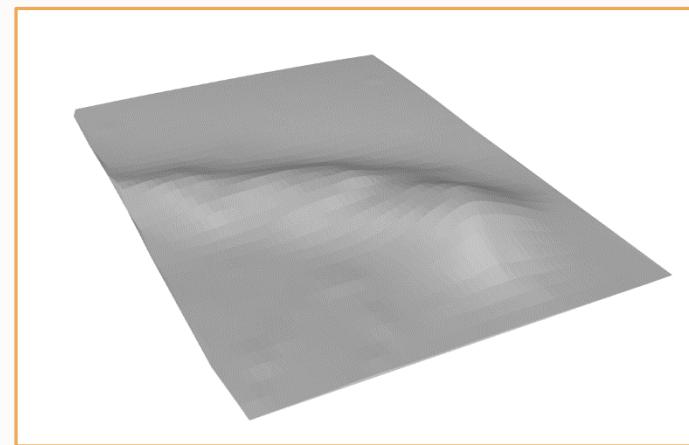


2つの点群の差を点群として返すプロシージャPC_DIFFERENCEが追加

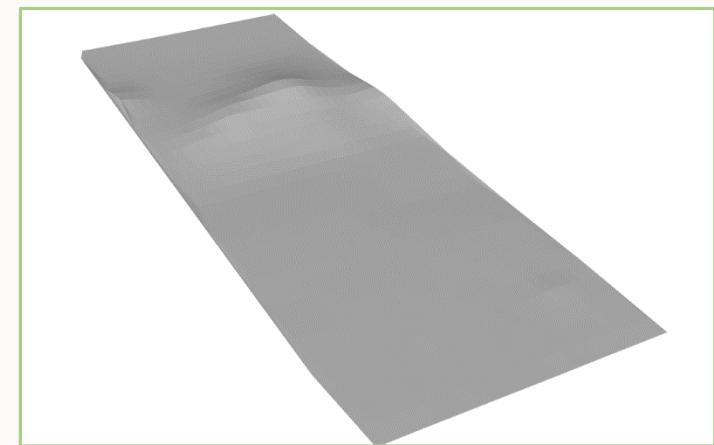
2つの点群の差分の検出



点群1



マイナス
点群2

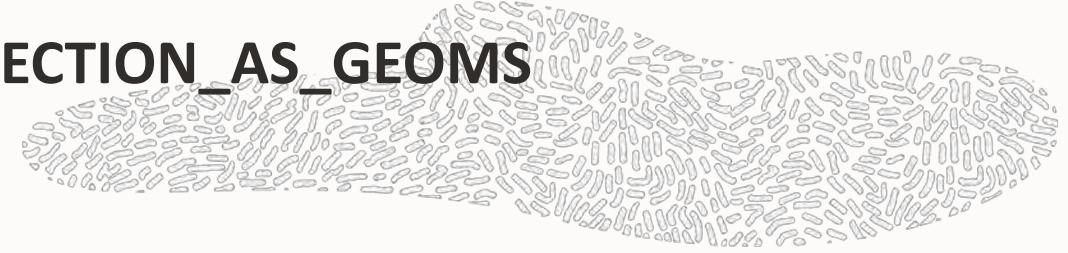


点群3





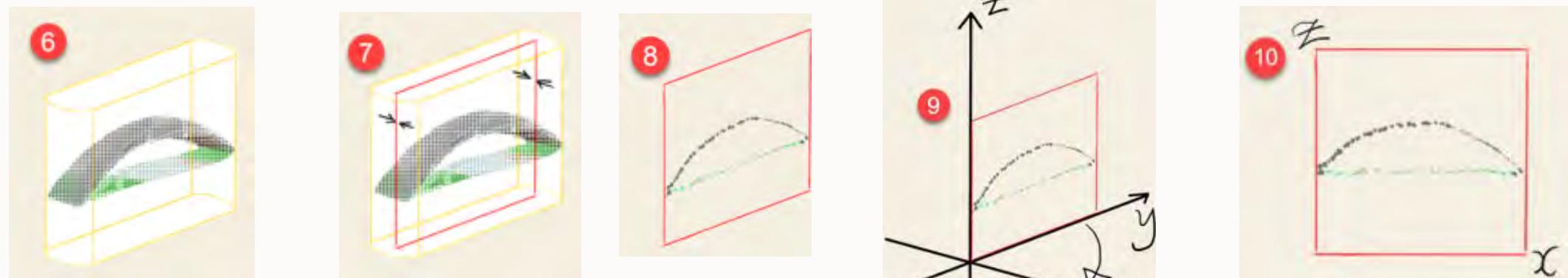
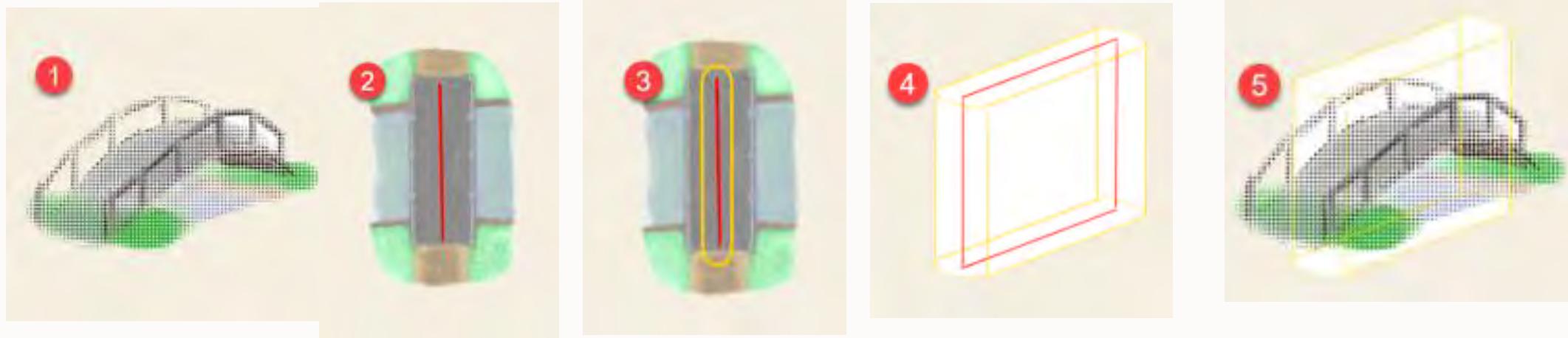
3D点群の断面を計算するGENERATE_CROSS_SECTION_AS_GEOMS



- 概要
 - Oracle Spatialの点群機能SDO_PC_PKGパッケージに、垂直の2D平面を利用して、3D点群の断面計算を行うGENERATE_CROSS_SECTION_AS_GEOMSプロシージャが追加
 - オブジェクトの輪郭の決定
 - 点群内部の測定
- ユースケース
 - 点群データから地勢、高低差の把握



3D点群の断面を計算するGENERATE_CROSS_SECTION_AS_GEOMS



画像提供 : CenterOne, www.centerone.nl



3次元の点の集合から3次元メッシュを生成するCREATE_MESHES

複雑な風景などのモデリング



- 概要

- Oracle Spatialの表面のモデリング機能SDO_TIN_PKGパッケージに3Dメッシュモデリングを行うCREATE_MESHESプロシージャが追加
- TINではサポートされていなかった垂直な表面、軒、閉じたボリュームなどの3D三角形オブジェクトのモデリングができる

- ユースケース

- より複雑な風景、滝、崖、建物、車などのモデリング



3次元の点の集合から3次元メッシュを生成するCREATE_MESHESプロシージャ 複雑な風景などのモデリング



	VAL_D1	VAL_D2	VAL_D3
1	10	10	3
2	8	7	4
3	9	7	4
4	10	7	4
5	7	8	4
6	8	8	4
7	9	8	4
8	10	8	4
9	7	9	4
10	8	9	4
11	9	9	4
12	10	9	4



XYZ座標を持つ3D点データセット

```
-- 3Dメッシュの作成
begin
    sdo_tin_pkg.create_meshes(
        base_table      => 'MESHES',
        data_table      => 'MESH_BLKS',
        tin_id          => 4,
        tin_tol         => 0.05,
        blk_size        => null,
        inptab          => 'INPTAB',
        srid            => 27700,
        feature_size    => 2,
        max_angle       => 120,
        batch_size_pts  => 1000);
end;
/
```



3次元の点の集合から3次元メッシュを生成するCREATE_MESHESプロシージャ 複雑な風景などのモデリング



```
-- 3Dメッシュを.objフォーマットに変換
select
  sdo_tin_pkg.clip_tin_into_obj(
    tin_table          => 'MESHES',
    tin_column         => 'TIN',
    id_column          => 'ID',
    id                 => 4,
    query              => null,
    where_clause       => null,
    result_table_name_pts =>
    'OBJ PTS BLKS',
    result_table_name_tr  =>
    'OBJ TR BLKS',
    lods               =>
    sdo_lods_type(1, 2, 3, 4, 5, 6, 7, 8, 9,
    10),
    blkno              => 1)
from
  dual;
```



.OBJファイルとしてエクスポートされた3Dメッシュ

Visualization: 3dviewer.net



ありがとうございました

