

StatisticalSpatialJoin

Title StatisticalSpatialJoin

Description

This is a modified version of the [Spatial Join](#) tool that is optimized for use in automated workflows that are focused on accumulating join features to target features using join fields.

Usage

- A spatial join involves matching rows from the Join Features to the Target Features based on their relative spatial locations.
- By default, all attributes of the join features are appended to attributes of the target features and copied over to the output feature class. You can define which of the attributes will be written to the output by manipulating them in the Field Map of Join Features parameter.
- Two new fields, Join_Count and TARGET_FID, are always added to the output feature class. Join_Count indicates how many join features match each target feature (TARGET_FID).
- Another new field, JOIN_FID, is added to the output when Join one to many (JOIN_ONE_TO_MANY in Python) is specified in the Join Operation parameter.
- When the Join Operation parameter is Join one to many, there can be more than one row in the output feature class for each target feature. The JOIN_FID field makes it easier to determine which feature is joined to which target feature (TARGET_FID). A value of -1 for JOIN_FID field means no feature meets the specified spatial relationship with the target feature.
- All input target features are written to the output feature class only if:
- The Join Operation is set to Join one to one, and Keep All Target Features is checked (join_type = "KEEP_ALL" in Python).
- Merge rules specified in the Field Map of Join Features parameter only apply to attributes from the join features and when more than one feature is matched to a target feature (when Join_Count > 1). For example, if three features with DEPTH attribute values of 15.5, 2.5, and 3.3 are joined, and a merge rule of Mean is applied, the output field will have a value of 6.1. Null values in join fields are ignored for statistic calculation. For example, 15.5, <null>, and 2.5 will result in 9.0 for Mean and 2 for Count.
- When the Match Option is set to Closest or Closest geodesic, it is possible that two or more join features are at the same distance from the target feature. When this situation occurs, one of the join features is randomly selected as the matching feature (the join feature's FID does not influence this random selection). If you want to find the 2nd, 3rd, or Nth closest feature, use the Generate Near Table tool.
- Learn more about how proximity is calculated

- If a join feature has a spatial relationship with multiple target features, then it is counted as many times as it is matched against the target feature. For example, if a point is within three polygons, then the point is counted three times, once for each polygon.
- For more information about using the three-dimensional spatial relationships Intersect 3D and Within a distance 3D see Select by Location 3D relationships.

Syntax

StatisticalSpatialJoin (Target_Feature_Class, Join_Feature_Class, Output_Feature_Class, Prepend_Name, Join_Operation, Join_Type, Match_Option, {Search_Radius}, {Sum_Fields}, {Mean_Fields}, {Median_Fields}, {Mode_Fields}, {Standard_Deviation_Fields}, {Minimum_Fields}, {Maximum_Fields}, {Range_Fields}, {Count_Fields}, {First_Fields})

Parameter	Explanation	Data Type
Target_Feature_Class	<p>Dialog Reference</p> <p>Attributes of the target features and the attributes from the joined features are transferred to the output feature class.</p> <p>There is no python reference for this parameter.</p>	Feature Layer
Join_Feature_Class	<p>Dialog Reference</p> <p>The attributes from the join features are joined to the attributes of the target features. See the explanation of the join_operation parameter for details on how the aggregation of joined attributes are affected by the type of join operation.</p> <p>There is no python reference for this parameter.</p>	Feature Layer
Output_Feature_Class	<p>Dialog Reference</p> <p>A new feature class containing the attributes of the target and join features. By default, all attributes of target features and the attributes of the joined features are written to the output. However, the set of attributes to be transferred can be controlled by the field map parameter.</p> <p>There is no python reference for this parameter.</p>	Feature Class
Prepend_Name	<p>Dialog Reference</p> <p>A string of text that is added between the merge rule and the join field name being added to the output feature class. Helps with labeling data sources.</p> <p>There is no python reference for this parameter.</p>	String
Join_Operation	<p>Dialog Reference</p> <p>Determines how joins between the target features</p>	String

and join features will be handled in the output feature class if multiple join features are found that have the same spatial relationship with a single target feature.

JOIN_ONE_TO_ONE —If multiple join features are found that have the same spatial relationship with a single target feature, the attributes from the multiple join features will be aggregated using a field map merge rule. For example, if a point target feature is found within two separate polygon join features, the attributes from the two polygons will be aggregated before being transferred to the output point feature class. If one polygon has an attribute value of 3 and the other has a value of 7, and a Sum merge rule is specified, the aggregated value in the output feature class will be 10. This is the default.

JOIN_ONE_TO_MANY —If multiple join features are found that have the same spatial relationship with a single target feature, the output feature class will contain multiple copies (records) of the target feature. For example, if a single point target feature is found within two separate polygon join features, the output feature class will contain two copies of the target feature: one record with the attributes of one polygon, and another record with the attributes of the other polygon.

There is no python reference for this parameter.

Join_Type	<p>Dialog Reference</p> <p>Determines if all target features will be maintained in the output feature class (known as outer join), or only those that have the specified spatial relationship with the join features (inner join).</p> <p>KEEP_ALL —All target features will be maintained in the output (outer join). This is the default.</p> <p>KEEP_COMMON — Only those target features that have the specified spatial relationship with the join features will be maintained in the output feature class (inner join). For example, if a point feature class is specified for the target features, and a polygon feature class is specified for the join features, with match_option = "WITHIN", the output feature class will only contain those target features that are within a polygon join feature; any target features not within a join feature will be excluded from the output.</p> <p>There is no python reference for this parameter.</p>	Boolean
-----------	---	---------

Match_Option	<p>Dialog Reference</p> <p>(Optional)</p>	String
--------------	---	--------

Defines the criteria used to match rows. The match options are:

INTERSECT —The features in the join features will be matched if they intersect a target feature. This is the default. Specify a distance in the `search_radius` parameter.

INTERSECT_3D — The features in the join features will be matched if they intersect a target feature in three-dimensional space (x, y, and z). Specify a distance in the `search_radius` parameter.

WITHIN_A_DISTANCE —The features in the join features will be matched if they are within a specified distance of a target feature. Specify a distance in the `search_radius` parameter.

WITHIN_A_DISTANCE_GEODESIC —Same as **WITHIN_A_DISTANCE** except that geodesic distance is used rather than planar distance. Choose this if your data covers a large geographic extent or the coordinate system of the inputs is unsuitable for distance calculations.

WITHIN_A_DISTANCE_3D —The features in the join features will be matched if they are within a specified distance of a target feature in three-dimensional space. Specify a distance in the `search_radius` parameter.

CONTAINS —The features in the join features will be matched if a target feature contains them. The target features must be polygons or polylines. For this option, the target features cannot be points, and the join features can only be polygons when the target features are also polygons.

COMPLETELY_CONTAINS —The features in the join features will be matched if a target feature completely contains them. Polygon can completely contain any feature. Point cannot completely contain any feature, not even a point. Polyline can completely contain only polyline and point.

CONTAINS_CLEMENTINI —This spatial relationship yields the same results as **COMPLETELY_CONTAINS** with the exception that if the join feature is entirely on the boundary of the target feature (no part is properly inside or outside) the feature will not be matched. Clementini defines the boundary polygon as the line separating inside and outside, the boundary of a line is defined as its end points, and the boundary of a point is always empty.

WITHIN —The features in the join features will be

matched if a target feature is within them. It is opposite to CONTAINS. For this option, the target features can only be polygons when the join features are also polygons. Point can be join feature only if point is target.

COMPLETELY_WITHIN —The features in the join features will be matched if a target feature is completely within them. This is opposite to COMPLETELY_CONTAINS.

WITHIN_CLEMENTINI —The result will be identical to WITHIN except if the entirety of the feature in the join features is on the boundary of the target feature, the feature will not be matched. Clementini defines the boundary polygon as the line separating inside and outside, the boundary of a line is defined as its end points, and the boundary of a point is always empty.

ARE_IDENTICAL_TO —The features in the join features will be matched if they are identical to a target feature. Both join and target feature must be of same shape type—point-to-point, line-to-line, and polygon-to-polygon.

BOUNDARY_TOUCHES —The features in the join features will be matched if they have a boundary that touches a target feature. When the target and join features are lines or polygons, the boundary of the join feature can only touch the boundary of the target feature and no part of the join feature can cross the boundary of the target feature.

SHARE_A_LINE_SEGMENT_WITH —The features in the join features will be matched if they share a line segment with a target feature. The join and target features must be lines or polygons.

CROSSED_BY_THE_OUTLINE_OF —The features in the join features will be matched if a target feature is crossed by their outline. The join and target features must be lines or polygons. If polygons are used for the join or target features, the polygon's boundary (line) will be used. Lines that cross at a point will be matched, not lines that share a line segment.

HAVE_THEIR_CENTER_IN —The features in the join features will be matched if a target feature's center falls within them. The center of the feature is calculated as follows: for polygon and multipoint the geometry's centroid is used, and for line input the geometry's midpoint is used. Specify a distance in the search_radius parameter.

CLOSEST —The feature in the join features that is

closest to a target feature is matched. See the usage tip for more information. Specify a distance in the search_radius parameter.

CLOSEST_GEODESIC —Same as CLOSEST except that geodesic distance is used rather than planar distance. Choose this if your data covers a large geographic extent or the coordinate system of the inputs is unsuitable for distance calculations

There is no python reference for this parameter.

Search_Radius (Optional)	<p>Dialog Reference</p> <p>Join features within this distance of a target feature will be considered for the spatial join. A search radius is only valid when the spatial relationship (match_option) INTERSECT, WITHIN_A_DISTANCE, WITHIN_A_DISTANCE_GEODESIC, HAVE_THEIR_CENTER_IN, CLOSEST or CLOSEST_GEODESIC is specified. Using a search radius of 100 meters with the spatial relationship WITHIN_A_DISTANCE will join feature within 100 meters of a target feature. For the three WITHIN_A_DISTANCE relationships, if no value is specified for search radius then a distance of 0 is used.</p> <p>There is no python reference for this parameter.</p>	Long
Sum_Fields (Optional)	<p>Dialog Reference</p> <p>Join Fields whose merge field will be set to SUM for a spatial join. Their names will take the form: [Merge_Rule+ Prepended_Name+ Field Name].</p> <p>There is no python reference for this parameter.</p>	Multiple Value
Mean_Fields (Optional)	<p>Dialog Reference</p> <p>Join Fields whose merge field will be set to MEAN for a spatial join. Their names will take the form: [Merge_Rule+ Prepended_Name+ Field Name].</p> <p>There is no python reference for this parameter.</p>	Multiple Value
Median_Fields (Optional)	<p>Dialog Reference</p> <p>Join Fields whose merge field will be set to MEDIAN for a spatial join. Their names will take the form: [Merge_Rule+ Prepended_Name+ Field Name].</p> <p>There is no python reference for this parameter.</p>	Multiple Value
Mode_Fields (Optional)	<p>Dialog Reference</p> <p>Join Fields whose merge field will be set to MODE for a spatial join. Their names will take the form: [Merge_Rule+ Prepended_Name+ Field Name].</p>	Multiple Value

There is no python reference for this parameter.

Standard_Deviation_Fields (Optional)	<p>Dialog Reference</p> <p>Join Fields whose merge field will be set to STD for a spatial join. Their names will take the form: [Merge_Rule+ Prepended_Name+ Field Name].</p> <p>There is no python reference for this parameter.</p>	Multiple Value
Minimum_Fields (Optional)	<p>Dialog Reference</p> <p>Join Fields whose merge field will be set to MIN for a spatial join. Their names will take the form: [Merge_Rule+ Prepended_Name+ Field Name].</p> <p>There is no python reference for this parameter.</p>	Multiple Value
Maximum_Fields (Optional)	<p>Dialog Reference</p> <p>Join Fields whose merge field will be set to MAX for a spatial join. Their names will take the form: [Merge_Rule+ Prepended_Name+ Field Name].</p> <p>There is no python reference for this parameter.</p>	Multiple Value
Range_Fields (Optional)	<p>Dialog Reference</p> <p>Join Fields whose merge field will be set to RANGE for a spatial join. Their names will take the form: [Merge_Rule+ Prepended_Name+ Field Name].</p> <p>There is no python reference for this parameter.</p>	Multiple Value
Count_Fields (Optional)	<p>Dialog Reference</p> <p>Join Fields whose merge field will be set to COUNT for a spatial join. Their names will take the form: [Merge_Rule+ Prepended_Name+ Field Name].</p> <p>There is no python reference for this parameter.</p>	Multiple Value
First_Fields (Optional)	<p>Dialog Reference</p> <p>Join Fields whose merge field will be set to FIRST for a spatial join. Their names will take the form: [Merge_Rule+ Prepended_Name+ Field Name].</p> <p>There is no python reference for this parameter.</p>	Multiple Value

Code Samples

There are no code samples for this tool.

Tags

Spatial Join, Statistics

Credits

David Wasserman

Use limitations

There are no access and use limitations for this item.

You are currently using the Item Description metadata style. Change your metadata style in the Options dialog box to see additional metadata content.