

# Data-Mining for Software Engineering: Nearest Neighbors

Martin Monperrus

Creative Commons Attribution License

Copying and modifying are authorized as long  
as proper credit is given to the author.

version of Dec 4, 2012



Université  
Lille1  
Sciences et Technologies

## Sources

---

Content from:

- Automated Method Completion (Hill, Rideout), 2004
- Learning from Examples to Improve Code Completion Systems (Bruch, Monperrus, Mezini), 2009
- Analogy-Based Practical Classification Rules for Software Quality Estimation (Khoshgoftaar and Seliya), 2003
- Estimating Maintenance Effort By Analogy (Leung), 2003

# Code completion: what do you need?

```
@Override
protected Control createDialogArea(Composite parent) {
    Composite container = (Composite) super.createDialogArea(parent);
    swtTextWidget = new Text(container, SWT.BORDER);
    swtTextWidget.
    return container;
}
}
```

**What should the code completion give?**

# Motivation

The image shows a screenshot of an IDE displaying the Java API for the `Text` class. The API is organized into four columns, each showing a list of methods with their signatures and return types. A red rectangular box is superimposed over the center of the image, containing the text "All 164 Methods of Text?".

**All 164 Methods of Text ?**

The methods listed in the API include:

- `handle` : int - Control
- `addControlListener`(ControlListener listener) : void
- `addDisposeListener`(DisposeListener listener) : void
- `addDragDetectListener`(DragDetectListener listener) : void
- `addFocusListener`(FocusListener listener) : void
- `addHelpListener`(HelpListener listener) : void
- `addKeyListener`(KeyListener listener) : void
- `addListener`(int eventType, Listener listener) : void
- `addMenuDetectListener`(MenuDetectListener listener) : void
- `addModifyListener`(ModifyListener listener) : void
- `addMouseListener`(MouseListener listener) : void
- `addMouseMoveListener`(MouseMoveListener listener) : void
- `addMouseTrackListener`(MouseTrackListener listener) : void
- `addMouseWheelListener`(MouseWheelListener listener) : void
- `addPaintListener`(PaintListener listener) : void
- `addSelectionListener`(SelectionListener listener) : void
- `addTransferListener`(TransferListener listener) : void
- `addVerticalTextChangeListener`(VerticalTextChangeListener listener) : void
- `append`(String string) : void
- `clearSelection`() : void
- `computeHorizontalBounds`() : Rectangle - Control
- `computeVerticalBounds`() : Rectangle - Control
- `copy`() : void
- `cut`() : void
- `dispose`() : void - Widget
- `dragDetect`(Event event) : boolean - Control
- `dragDetect`(MouseEvent event) : boolean - Control
- `equals`(Object obj) : boolean - Object
- `forceFocus`() : boolean - Control
- `getAccessible`() : Accessible - Control
- `getBackground`() : Color - Control
- `getBackgroundImage`() : Image - Control
- `getBorderWidth`() : int - Text
- `getBounds`() : Rectangle - Control
- `getCaretLineNumber`() : int - Text
- `getCaretLocation`() : Point - Text
- `getCaretPosition`() : int - Text
- `getCharCount`() : int - Text
- `getClass`() : Class<?> - Object
- `getClientArea`() : Rectangle - Scrollable
- `getCursor`() : Cursor - Control
- `getData`() : Object - Widget
- `getData`(String key) : Object - Widget
- `getDisplay`() : Display - Widget
- `getDoubleClickEnabled`() : boolean - Text
- `getDragDetect`() : boolean - Control
- `getEchoChar`() : char - Text
- `getEditable`() : boolean - Text
- `getEnabled`() : boolean - Control
- `getFont`() : Font - Control
- `getForeground`() : Color - Control
- `getHorizontalBar`() : ScrollBar - Scrollable
- `getLayoutData`() : Object - Control
- `getLineCount`() : int - Text
- `getLineDelimiter`() : String - Text
- `getLineHeight`() : int - Text
- `getListeners`(int eventType) : Listener[] - Widget
- `getSelectionText`() : String - Text
- `getShell`() : Shell - Control
- `getSize`() : Point - Control
- `getStyle`() : int - Widget
- `getTabs`() : int - Text
- `getText`() : String - Text
- `getText`(int start, int end) : String - Text
- `getTextLimit`() : int - Text
- `getToolTipText`() : String - Control
- `getTopIndex`() : int - Text
- `getTopPixel`() : int - Text
- `getVerticalBar`() : ScrollBar - Scrollable
- `getVisible`() : boolean - Control
- `hashCode`() : int - Object
- `insert`(String string) : void - Text
- `internal_dispose_GC`(int hDC, GCData data) : void
- `internal_new_GC`(GCData data) : int - Control
- `isDisposed`() : boolean - Widget
- `isEnabled`() : boolean - Control
- `isFocusControl`() : boolean - Control
- `isListening`(int eventType) : boolean - Widget
- `isReparentable`() : boolean - Control
- `isVisible`() : boolean - Control
- `moveAbove`(Control control) : void - Control
- `moveBelow`(Control control) : void - Control
- `notify`() : void - Object
- `notifyAll`() : void - Object
- `notifyListeners`(int eventType, Event event) : void
- `pack`() : void - Control
- `pack`(boolean changed) : void - Control
- `paste`() : void - Text
- `print`(GC gc) : boolean - Control
- `redraw`() : void - Control
- `removeModifyListener`(ModifyListener listener) : void
- `removeMouseListener`(MouseListener listener) : void
- `removeMouseMoveListener`(MouseMoveListener listener) : void
- `removeMouseTrackListener`(MouseTrackListener listener) : void
- `removeMouseWheelListener`(MouseWheelListener listener) : void
- `removePaintListener`(PaintListener listener) : void
- `removeSelectionListener`(SelectionListener listener) : void
- `removeTraverseListener`(TraverseListener listener) : void
- `removeVerifyListener`(VerifyListener listener) : void
- `selectAll`() : void - Text
- `setBackground`(Color color) : void - Control
- `setBackgroundImage`(Image image) : void - Control
- `setBounds`(Rectangle rect) : void - Control
- `setBounds`(int x, int y, int width, int height) : void - Control
- `setCapture`(boolean capture) : void - Control
- `setCursor`(Cursor cursor) : void - Control
- `setData`(Object data) : void - Widget
- `setData`(String key, Object value) : void - Widget
- `setDoubleClickEnabled`(boolean doubleClick) : void - Control
- `setDragDetect`(boolean dragDetect) : void - Control
- `setEchoChar`(char echo) : void - Text
- `setEditable`(boolean editable) : void - Text
- `setEnabled`(boolean enabled) : void - Control
- `setFocus`() : boolean - Control
- `setFont`(Font font) : void - Text
- `setForeground`(Color color) : void - Control
- `setLayoutData`(Object layoutData) : void - Control
- `setLocation`(Point location) : void - Control
- `setLocation`(int x, int y) : void - Control
- `setMenu`(Menu menu) : void - Control
- `setMessage`(String message) : void - Text
- `setOrientation`(int orientation) : void - Text
- `setParent`(Composite parent) : boolean - Control
- `setTextLimit`(int limit) : void - Text
- `setToolTipText`(String string) : void - Control
- `setTopIndex`(int index) : void - Text
- `setVisible`(boolean visible) : void - Control
- `showSelection`() : void - Text
- `toControl`(Point point) : Point - Control
- `toControl`(int x, int y) : Point - Control
- `toDisplay`(Point point) : Point - Control
- `toDisplay`(int x, int y) : Point - Control
- `toString`() : String - Widget
- `traverse`(int traversal) : boolean - Control
- `update`() : void - Control
- `wait`() : void - Object
- `wait`(long timeout) : void - Object
- `wait`(long timeout, int nanos) : void - Object
- `DELIMITER` : String - Text
- `LIMIT` : int - Text

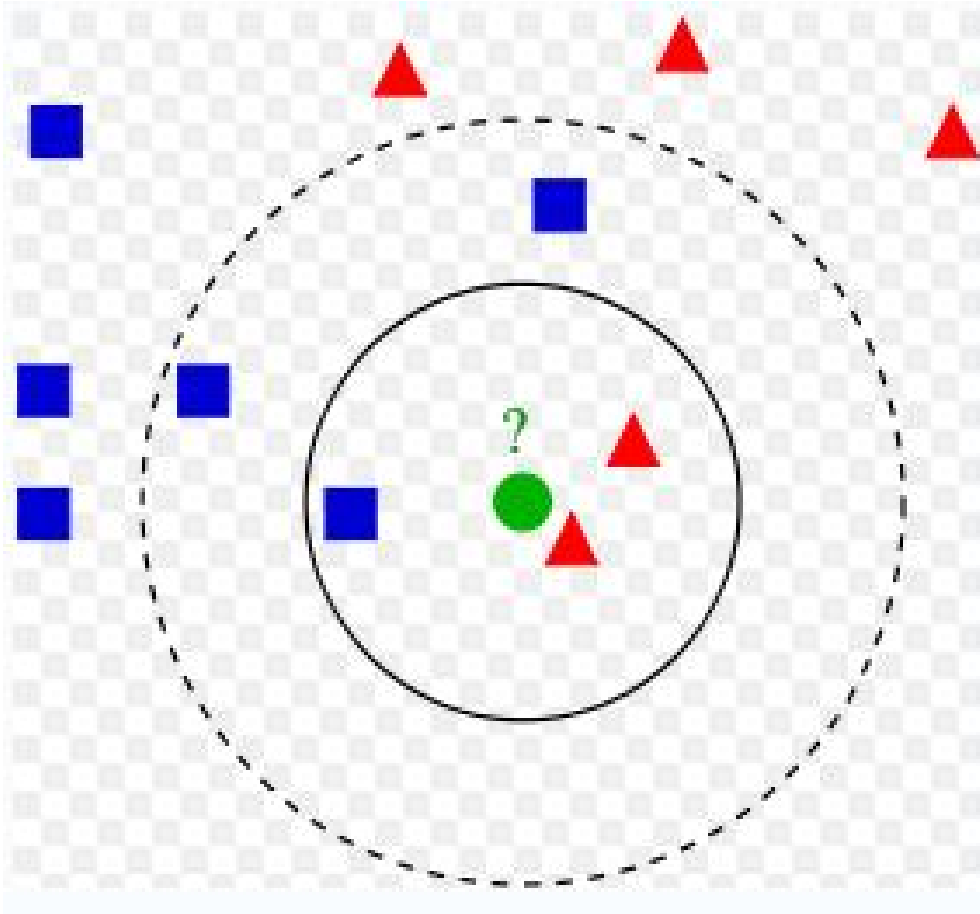
# Code completion: what do you need?

```
@Override
protected Control createDialogArea(Composite parent) {
    Composite container = (Composite) super.createDialogArea(parent);
    swtTextWidget = new Text(container, SWT.BORDER);
    swtTextWidget.
    return container;
}
}
```

**We can look for similar code and recommend methods that are in similar code.**

**Similar code = nearest neighbors**

# Reminder



Task: classifying the observation (in green) as triangle or square

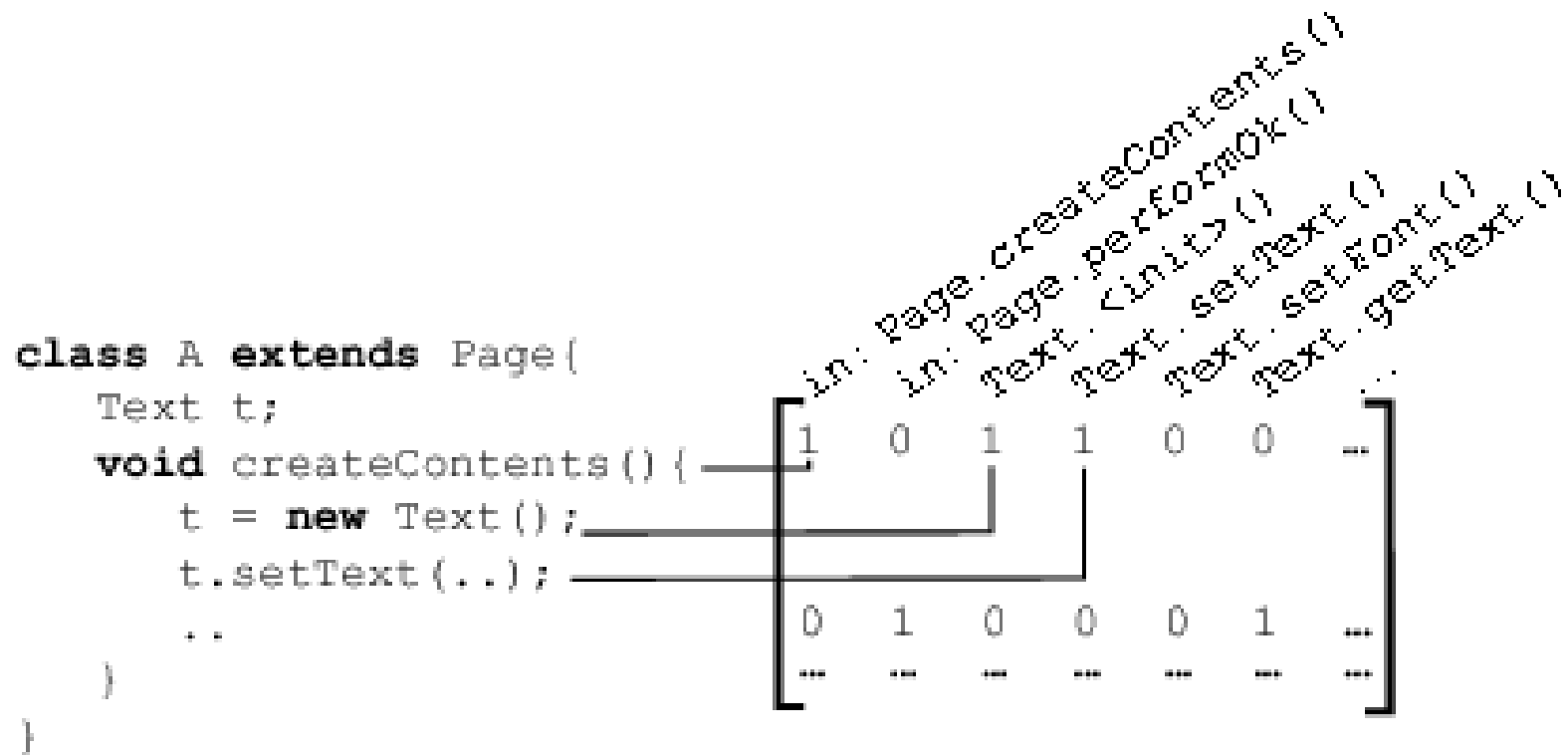
Space: 2-dimensional real features

Metric: euclidean distance

Strategy: majority

See also: Nearest neighbor pattern classification (Cover and Hart), 1967

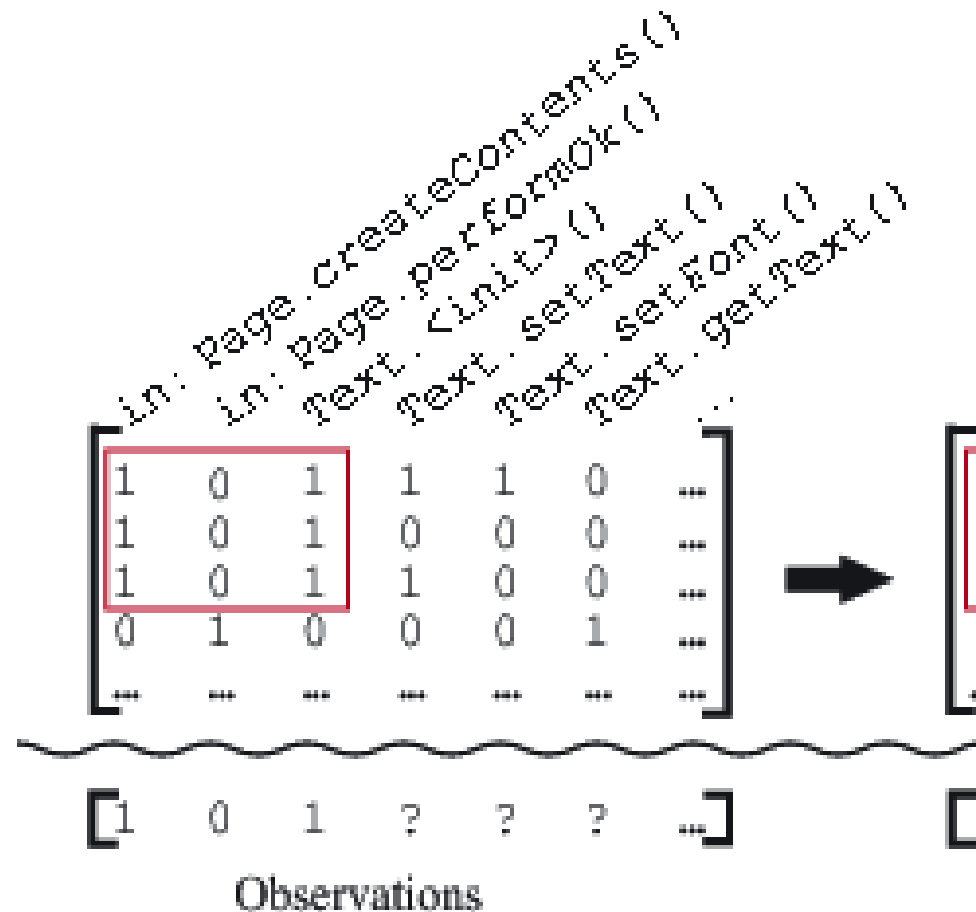
# Feature Space



Each variable is a point

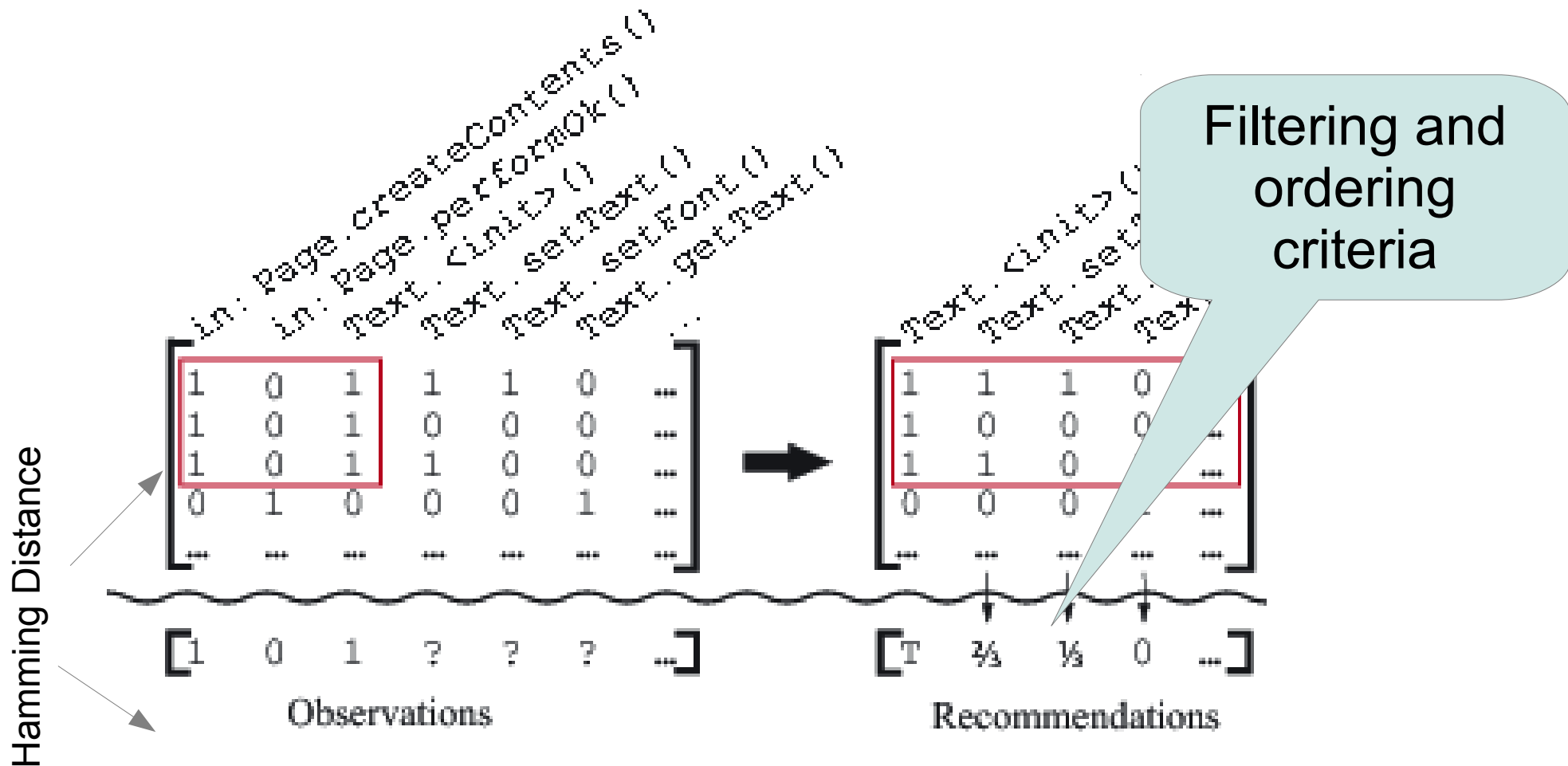
In a binary space of size 1500+

# Distance metric



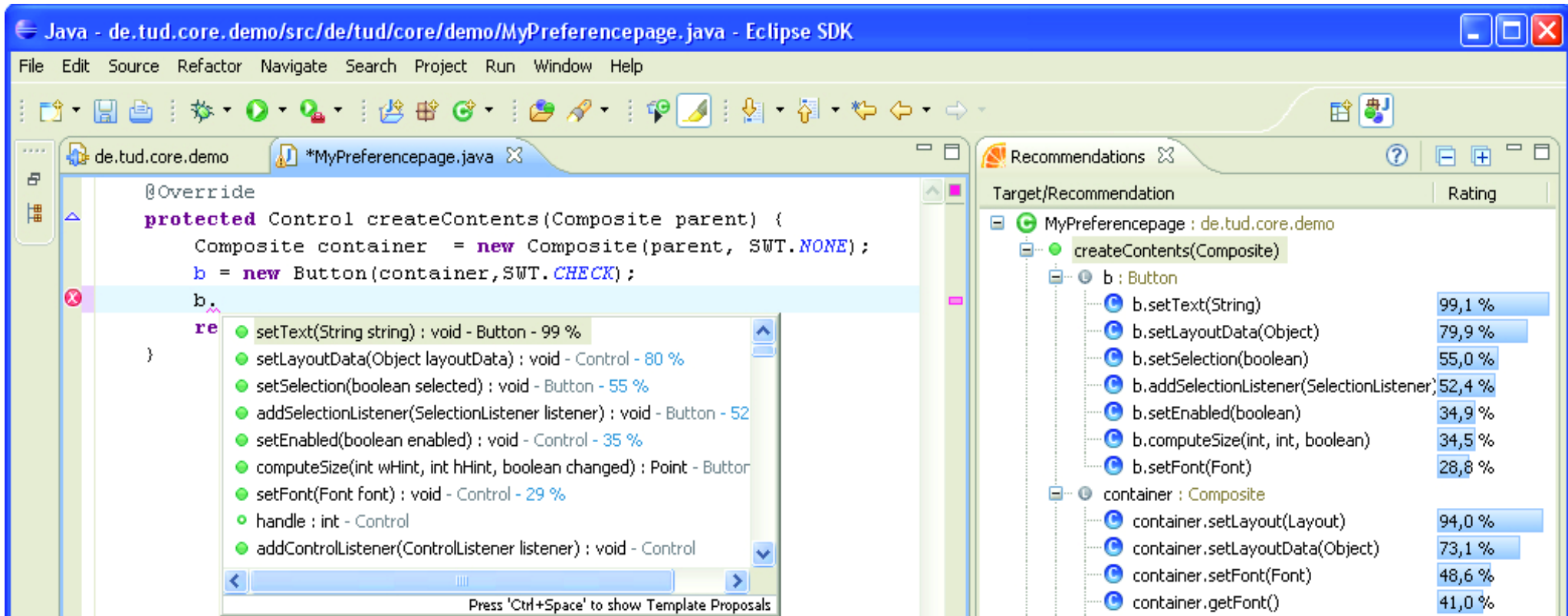
**Distance metric: only ones in the observation are taken into accounts**

# Classification? No, prediction!



**No classification but prediction of the calls to be put in the code completion systems based on the nearest neighbors**

# Screenshot



- ~~Ordered by alphabetical order~~
- ~~Poorly context aware~~
- Too many (useless) methods proposed?

## Automatic Evaluation

---

- simulate queries to code completion based on real data
- arbitrary choice of removing 50% of the calls
- train and test dataset are different (cross-validation)
- large-scale (50000+ different queries)
- precision and recall as performance metrics

## Precision and Recall

---

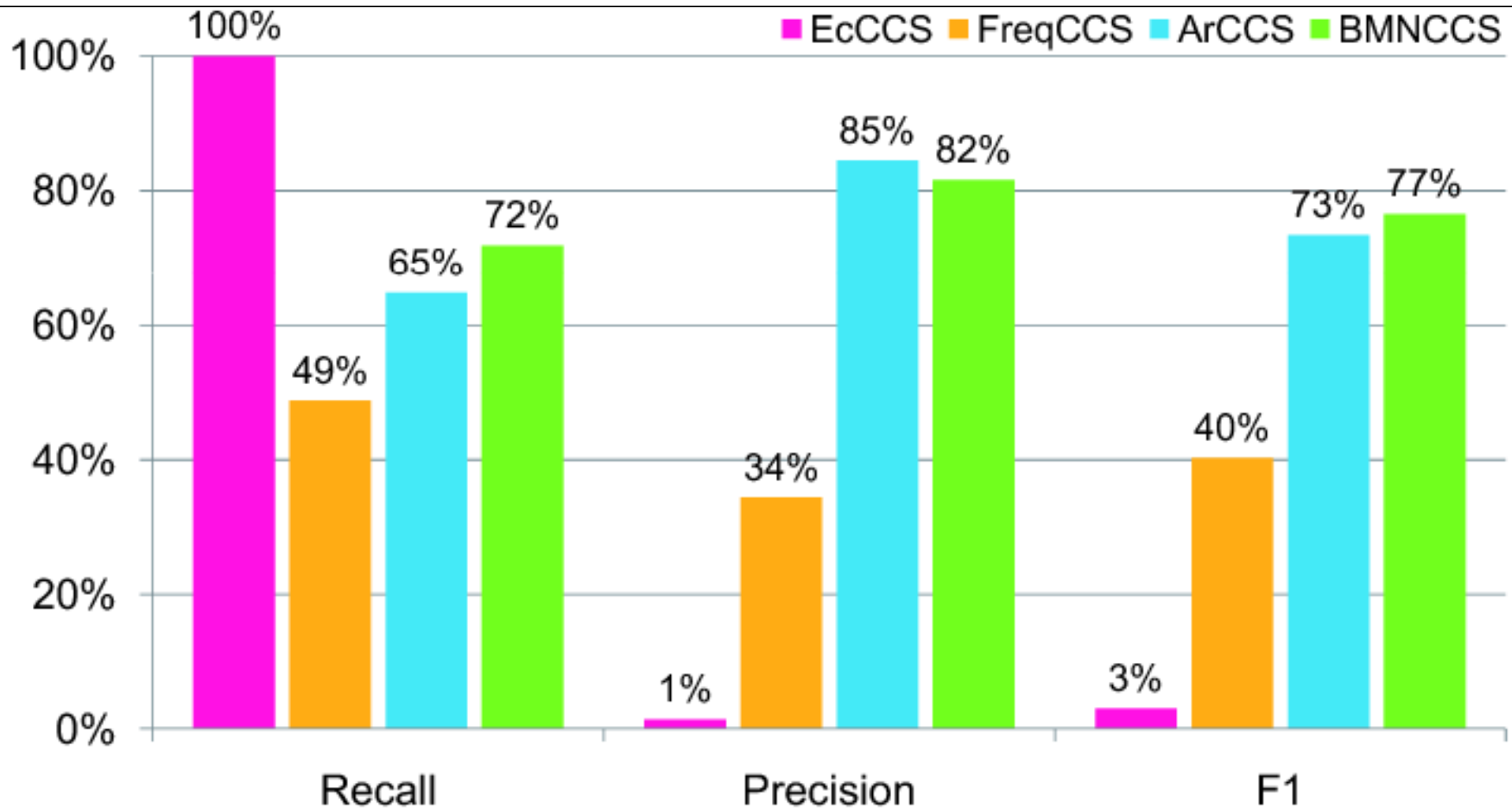
- Record: SWT::Text object, <init>, setText, setFont (real data)
- Split in:
  - Query: <init>, setText
  - Expectation: setFont
- Recommended:
  - setFont, getText
- Precision: 50%, Recall: 100%

## Competitors

---

- Type system (default Eclipse code completion)
  - based on the type system, recall = ?
- Frequency based
  - based on the observed frequency with a cut-off threshold
- Association rule based
  - out of the scope of this lecture

## Evaluation



Code completion can be improved based on:

- an appropriate feature space (context)
- an appropriate metric

# Advertisement

---

Try it by yourself !



... in 20 5 Slides

<http://www.eclipse.org/recommenders/>

## Automatic method completion

---

- Certain code clones are good
- Generally small
- They are small units of implementation, for example implementing a listener interface, or handling an keyboard event

**How to find them at development time?**

## Automatic method completion

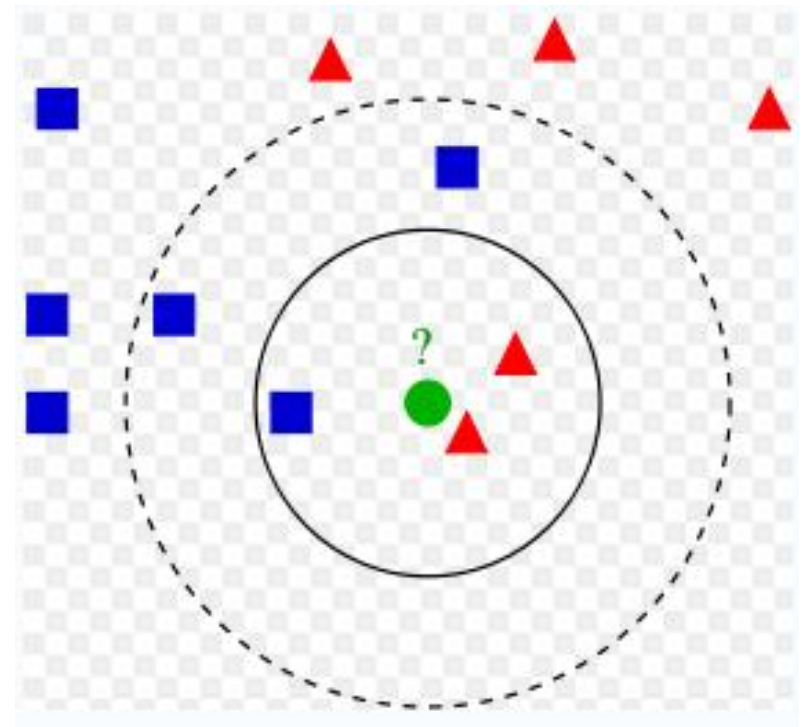
---

- For each method  $k$  in the software project, one generates a corresponding  $v^k$  of the form:  
$$v^k = [\text{lines}, \text{complexity}, \text{args}, \text{token1} \dots \text{token150}]$$
- Token: count of each of the 150 token types that the Java Language Specification
- Integer space of space 153

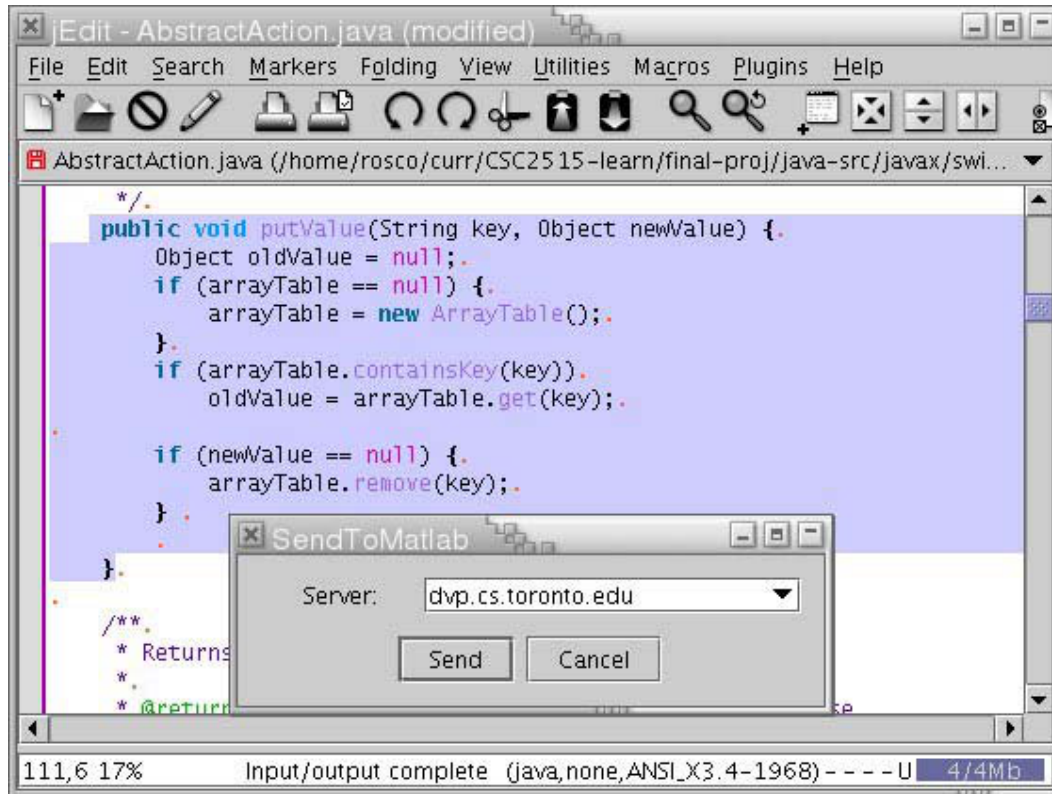
# Automatic method completion

- Euclidean distance
- The number (k) of nearest neighbors does not matter, the RSSE orders by distance

```
protected void fireCaretUpdateTwo(CaretEvent e) {  
    // Should I check if listeners is null?  
    Object[] listeners = listenerList.getListenerList();  
    // Send event to all listeners  
    for(int i = listener.length;i>=0;i-=1) {  
        ((CaretListener)listeners[i]).caretUpdate(e);  
    }  
}
```

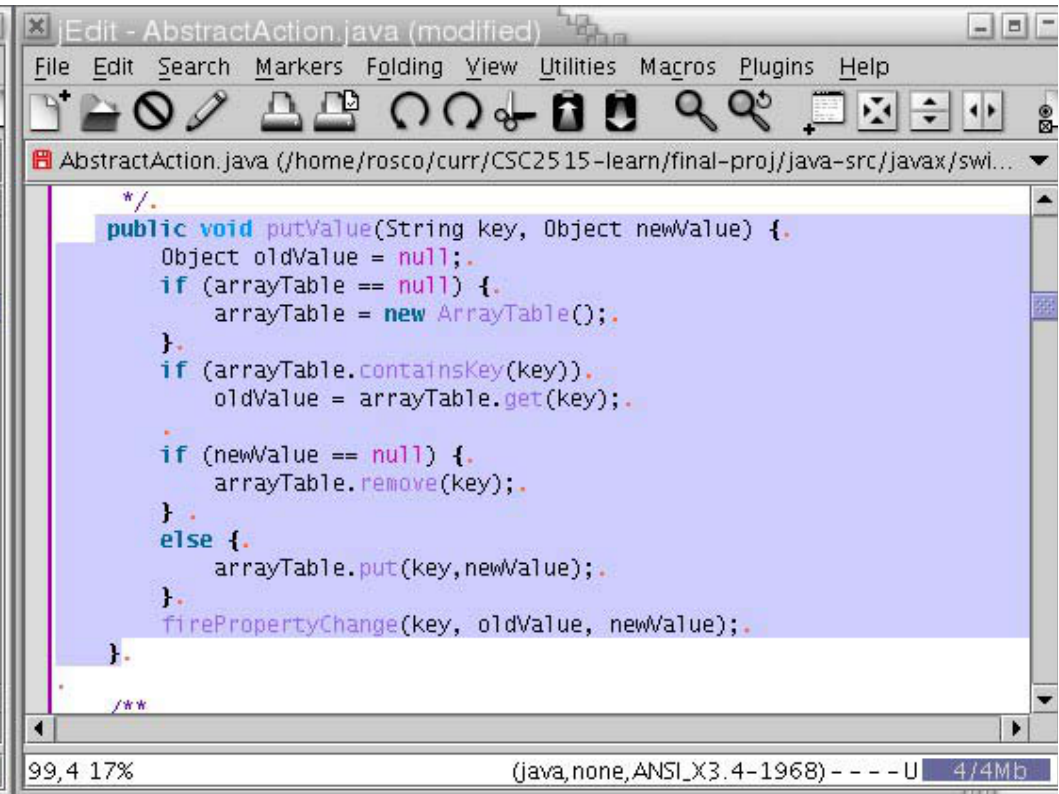


# Jedit prototype



The screenshot shows the Jedit IDE with the file 'AbstractAction.java' open. The code defines a 'putValue' method that updates an 'arrayTable' and fires a 'PropertyChange' event. A 'SendToMatlab' dialog box is overlaid on the code, with the server field set to 'dvp.cs.toronto.edu'. The status bar at the bottom indicates 'Input/output complete (java,none,ANSI\_X3.4-1968) - - - -U 4/4Mb'.

```
*/.  
public void putValue(String key, Object newValue) {  
    Object oldValue = null;.  
    if (arrayTable == null) {  
        arrayTable = new ArrayTable();.  
    }.  
    if (arrayTable.containsKey(key)).  
        oldValue = arrayTable.get(key);.  
  
    if (newValue == null) {  
        arrayTable.remove(key);.  
    }.  
}.  
  
/**.  
 * Returns  
 * .  
 * @return  
 */
```



The screenshot shows the Jedit IDE with the file 'AbstractAction.java' open. The code defines a 'putValue' method that updates an 'arrayTable' and fires a 'PropertyChange' event. The code path is different from the previous screenshot, showing the 'else' block where 'arrayTable.put' and 'firePropertyChange' are called. The status bar at the bottom indicates '(java,none,ANSI\_X3.4-1968) - - - -U 4/4Mb'.

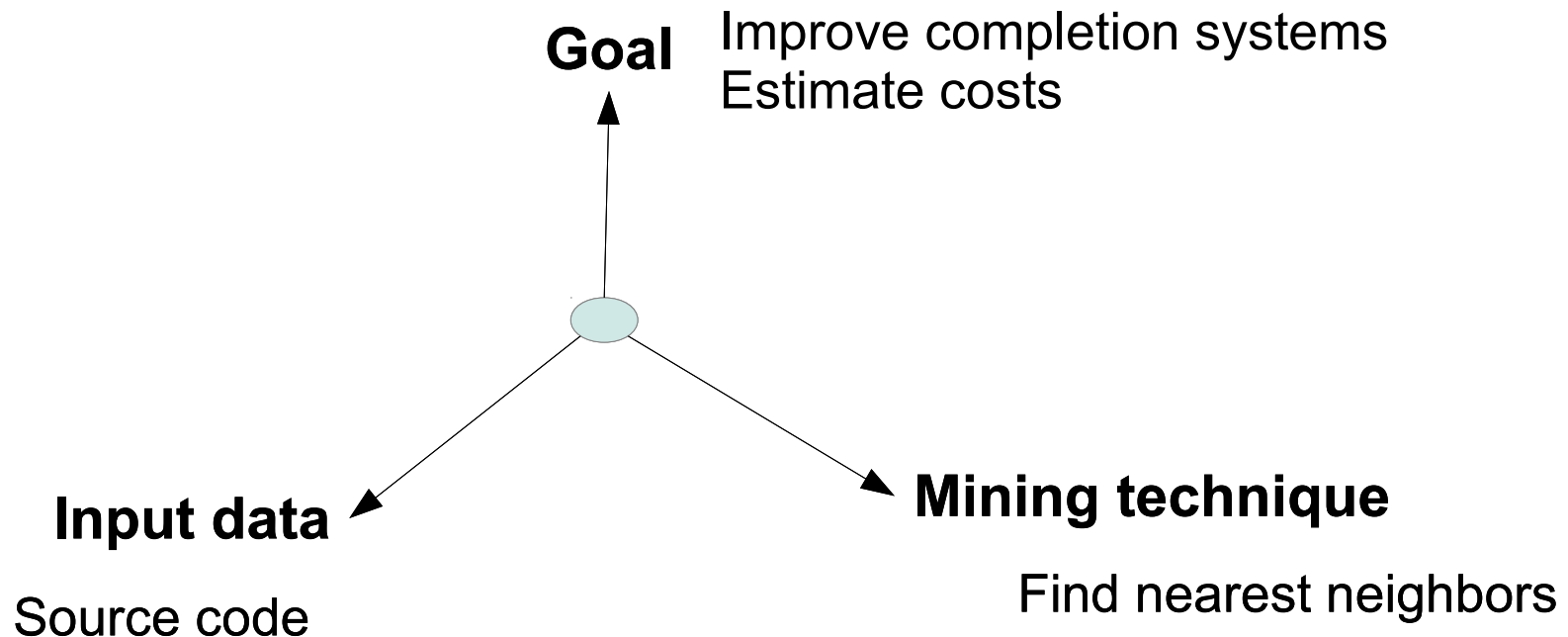
```
*/.  
public void putValue(String key, Object newValue) {  
    Object oldValue = null;.  
    if (arrayTable == null) {  
        arrayTable = new ArrayTable();.  
    }.  
    if (arrayTable.containsKey(key)).  
        oldValue = arrayTable.get(key);.  
  
    if (newValue == null) {  
        arrayTable.remove(key);.  
    }.  
    else {  
        arrayTable.put(key, newValue);.  
        firePropertyChange(key, oldValue, newValue);.  
    }.  
}.  
  
/**
```

## Summary

	Bruch et al, 2009	Hill et al. 2004	other
Goal	method call completion	method body completion	cost estimation
Feature space	binary space, ~1500	integer space, ~150	
Distance metric	hamming based	euclidean	
Classification	none	none	
Synthesis	recommendations	none	

## Recap.

- It is powerful to use data-mining techniques for solving software engineering problems
- Technique: nearest neighbors



# The big picture

## Datamining for Software Engineering

Three axes to characterize the contributions in the field.

