- **Reference Instances**: a subset of training cases used by the similarity based method.

- Reasons why one should select the reference instances:

  1. if training set very large – most of the cases have no influence on classification, including all decreases the computing performance.

  2. if data noisy: possible increase in prediction ability on unseen cases.

  3. large number of training cases: hard to understand the structure of the data, reference selection allows to find the most informative (interesting) prototypes.

- **SBL-PM** algorithm (the kernel):

1. Set the partial memory of the system (reference set) to the enire training set: $R = T = \{\mathbf{R_i}\},\ i=1,\ldots,N$.

2. Set the classification accuracy Δ to the value obtained from the leave-one-out test on T or to the value given by the user.

3. For $i=1$ to N:

   (a) Select one case $\mathbf{R_i}$ form R and set the temporary reference set to $R^{'} = R - \mathbf{R_i}$.

   (b) Using the current reference set $R^{'}$ as the training set and the whole original training set T as the test set calculate the prediction accuracy A_c.

   (c) if $A_c \geq \Delta$ set $R = R^{'}$.

1. Use the reference set R as a training set to calculate the prediction ability on unseen cases.

- The Δ parameter controls the number of reference cases that remain in partial memory: in general the greater is its value the more cases remain in partial memory.

- **The Extended Batch Version**

1. Set the partial memory of the system (reference set) to the entire training set: $R = T = \{\mathbf{R_i}\},\ i = 1,\ldots,N$.

2. Set the classification accuracy Δ to Δ_1 obtained from the leave-one-out test on T and the lowest accuracy that should be considered Δ_m.

3. Define the δ parameter determining steps in which the target accuracy Δ is lowered, (Ex. $\delta = 0.05$).

   (a) Until $\Delta < \Delta_m$

      i. For $i = 1$ to N:

      ii. Select one case $\mathbf{R_i}$ form R and set the temporary reference set to $R' = R - \mathbf{R_i}$.

iii. Using the current reference set $R^{'}$ as the training set and the whole original training set T as the test set calculate the prediction accuracy A_c.

iv. if $A_c \geq \Delta$ set $R = R^{'}$.

(b) Set $A_e(\Delta) = A_c$ to record the accuracy at the end of this step.

(c) Set $R(\Delta) = R$ to remember the reference vectors at this stage.

(d) Change $\Delta \leftarrow \Delta - \delta$

4. Select the references obtained for the highest $A_e(\Delta)$.

1. **The on-line version**

   (a) The off-line versions of **SBL-PM** require access to all cases in the training set.

   (b) On-line version has to decide weather the new case X_k coming from the input stream should be added to the partial memory of past cases.

   (c) The **SBL-PM On-Line** builds a partial memory forgetting cases that did not appear for a longer time.

1. **SBL-PM On-Line** algorithm:

   - Set the maximum number of reference vectors N^r_{max} and the maximum number of training vectors N^t_{max}.

   - Take the first incoming vector $\mathbf{X}_1$ as the first reference $R = \{\mathbf{X}_1\}$ and the training vector $T = \{\mathbf{X}_1\}$.

   - Repeat for all incoming vectors $\mathbf{X}_k$:

     - Add the incoming vector $\mathbf{X}_k$ to the training set T created so far.

     - determine the class $C(\mathbf{X}_k)$ of this vector using the reference set created so far.

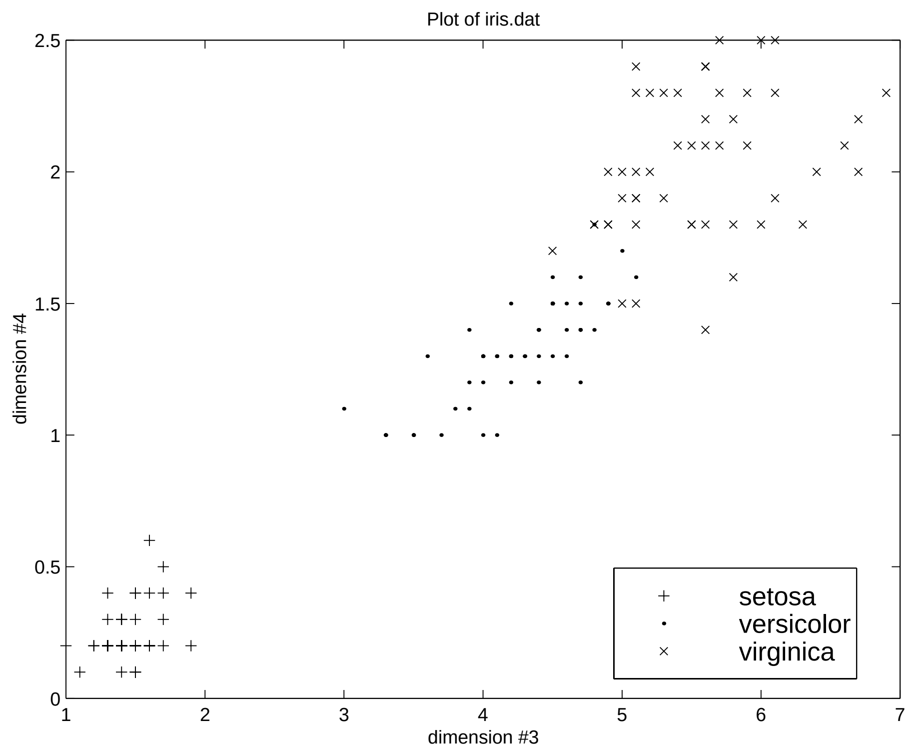
     - If $C(\mathbf{X}_k)$ is not correct add $\mathbf{X}_k$ to the current R.

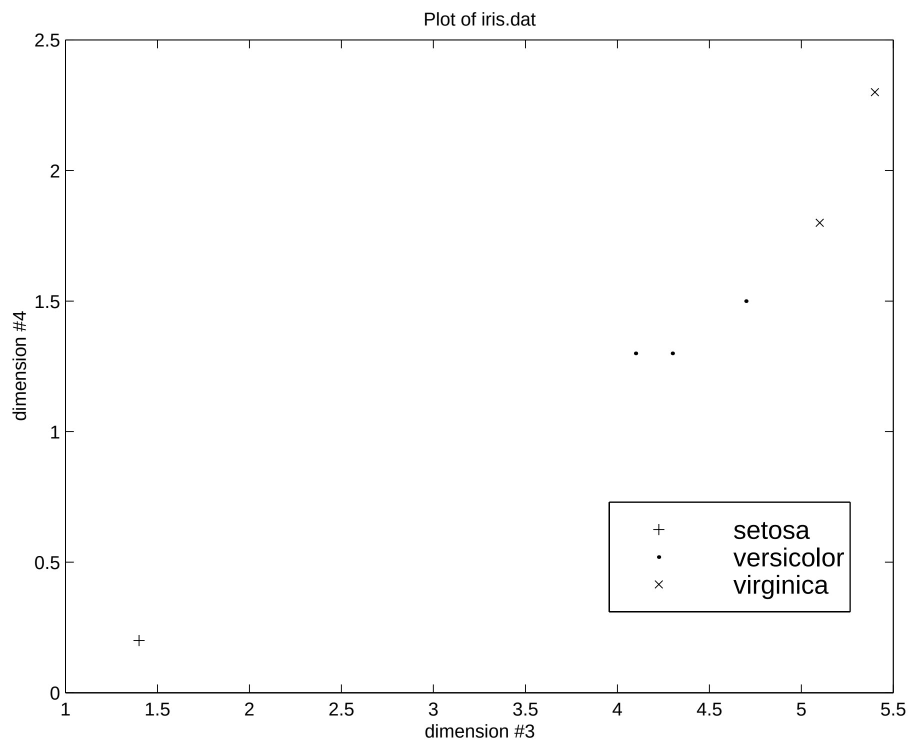
     - If $N_r \geq N^r_{max}$ or $N_t \geq N^t_{max}$, where $N_r(N_t)$ is the number of vectors in R (T), then

* Perform the batch step reducing R.
* Empty the training set T.

- **Results**

Dataset	**Remaining**	**SBL-PM**	**k-NN**
Append., CV	2.76, 106	82.95 ± 3.18	81.95 ± 1.45
Hepat., CV	4.3, 155	81.07 ± 2.84	78.77 ± 1.04
Ionosphere	19, 200	93.33	92
Iris, CV	6.7, 150	95.3 ± 1.7	95.8 ± 0.3


Plot of iris.dat
dimension #4
dimension #3
setosa
versicolor
virginica
0
0.5
1
1.5
2
2.5
1
2
3
4
5
6
7


Plot of iris.dat
dimension #4
dimension #3
setosa
versicolor
virginica
0
0.5
1
1.5
2
2.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5