Empirical Assessment of Classification Accuracy of Local SVM

Nicola Segata Enrico Blanzieri

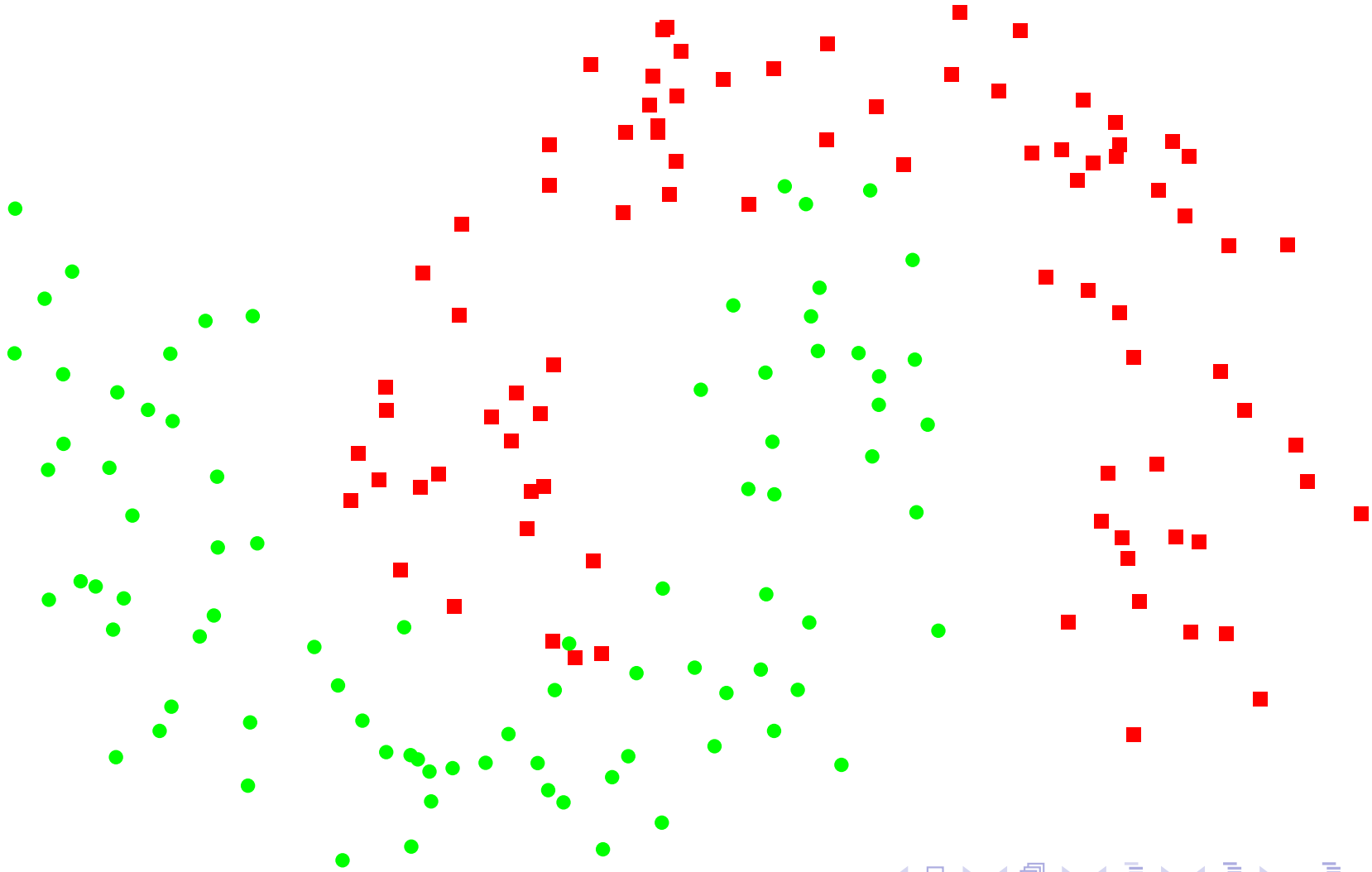
Department of Engineering and Computer Science (DISI)
University of Trento, Italy.
segata@disi.unitn.it

18th Annual Belgian-Dutch Conference on Machine Learning

Tilburg University May 19, 2009

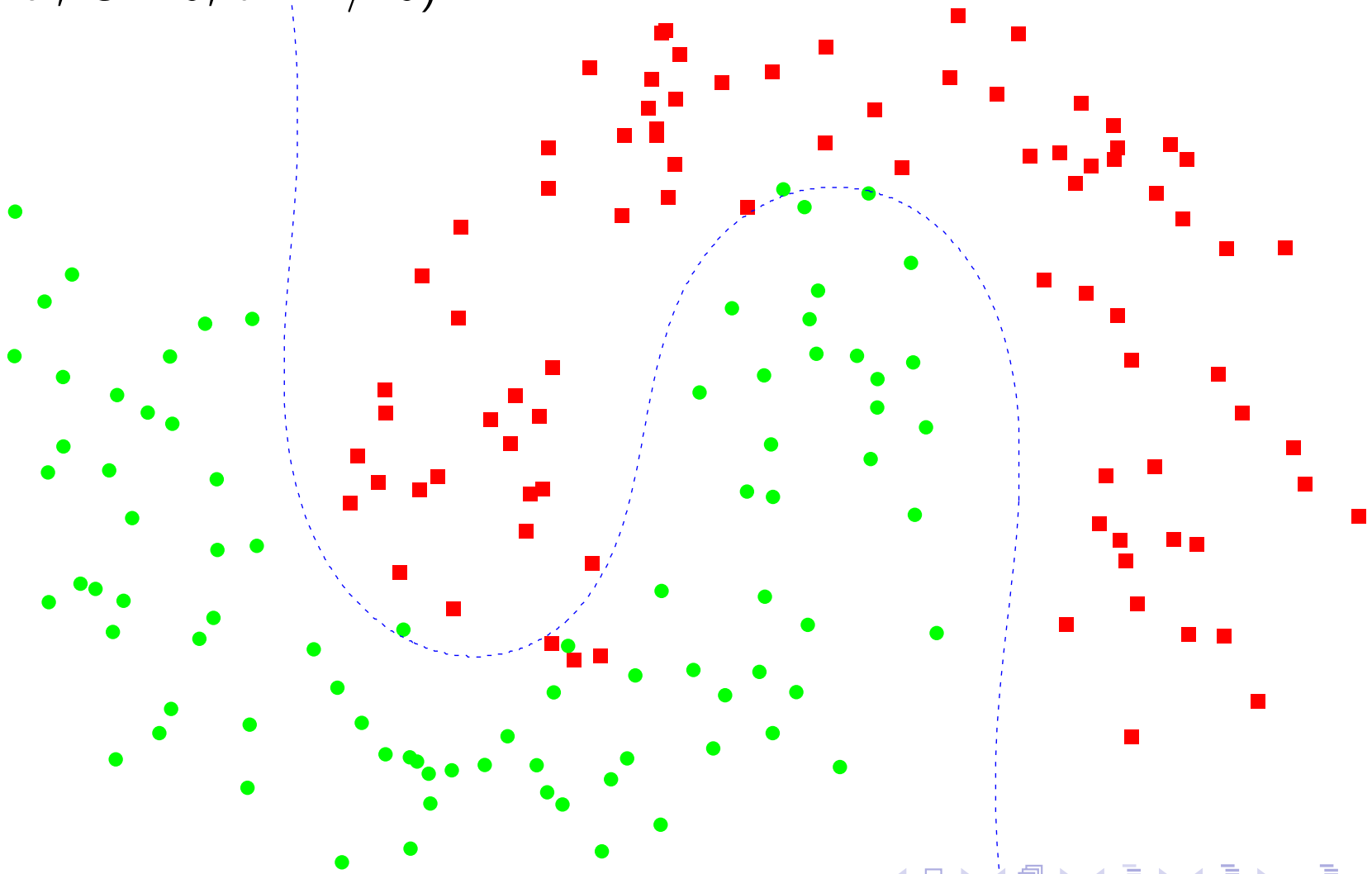
Outline of the talk

The traditional global approach with SVM



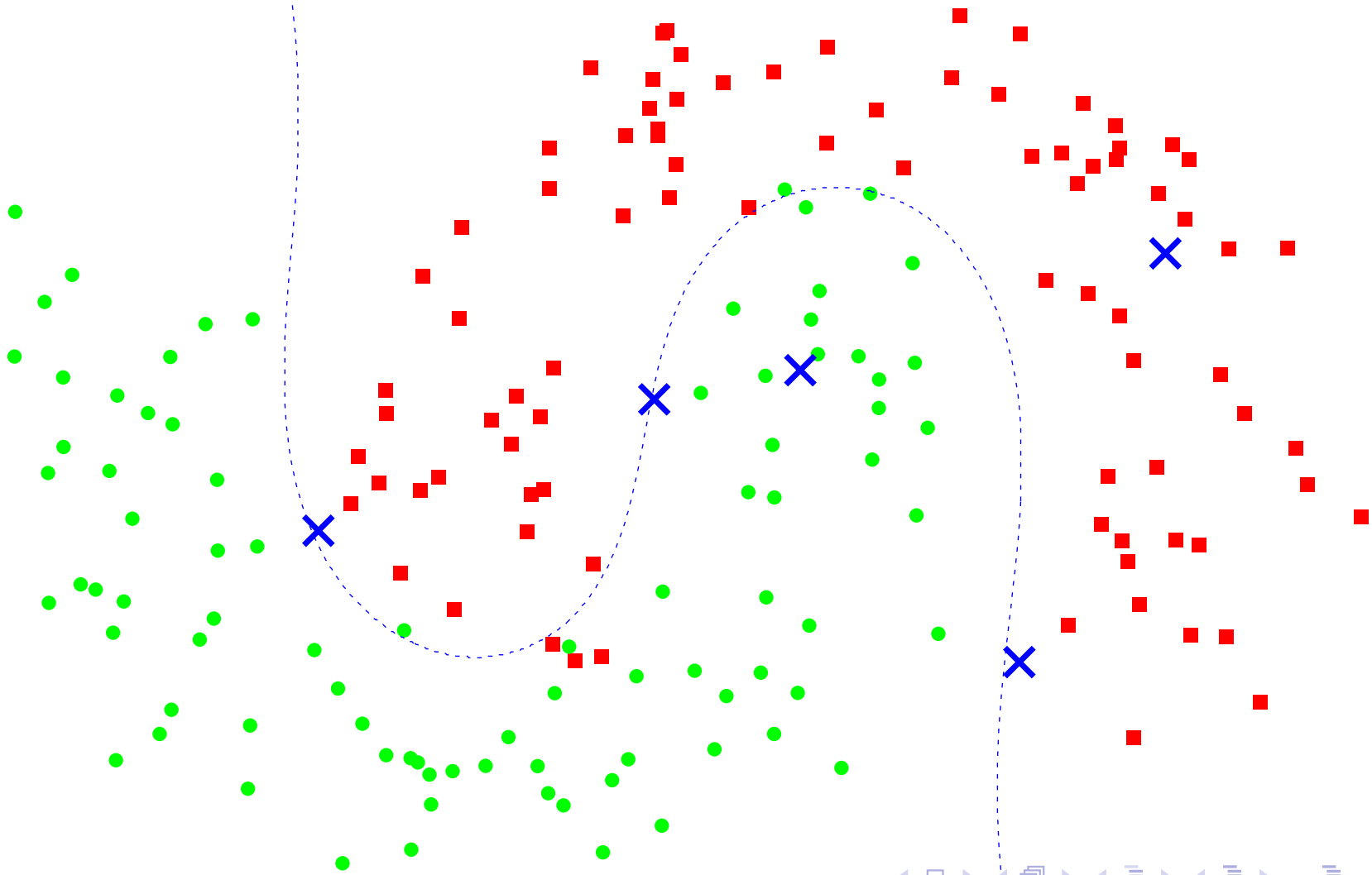
The traditional global approach with SVM

1. Use all training samples to estimate the decision function (SVM with RBF kernel, C=10, σ =1/10)



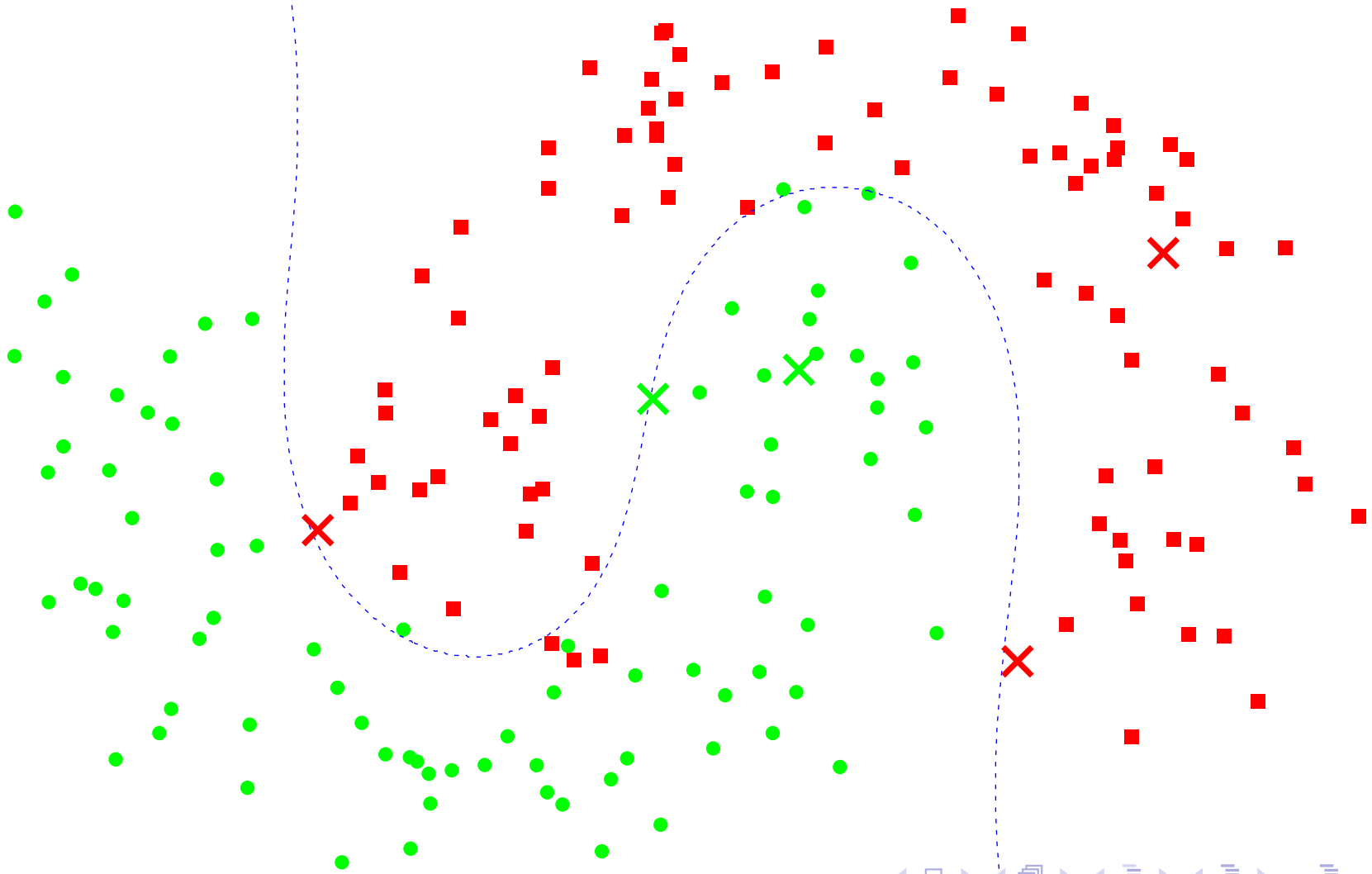
The traditional global approach with SVM

2. Each testing point is analysed using the global discriminative model



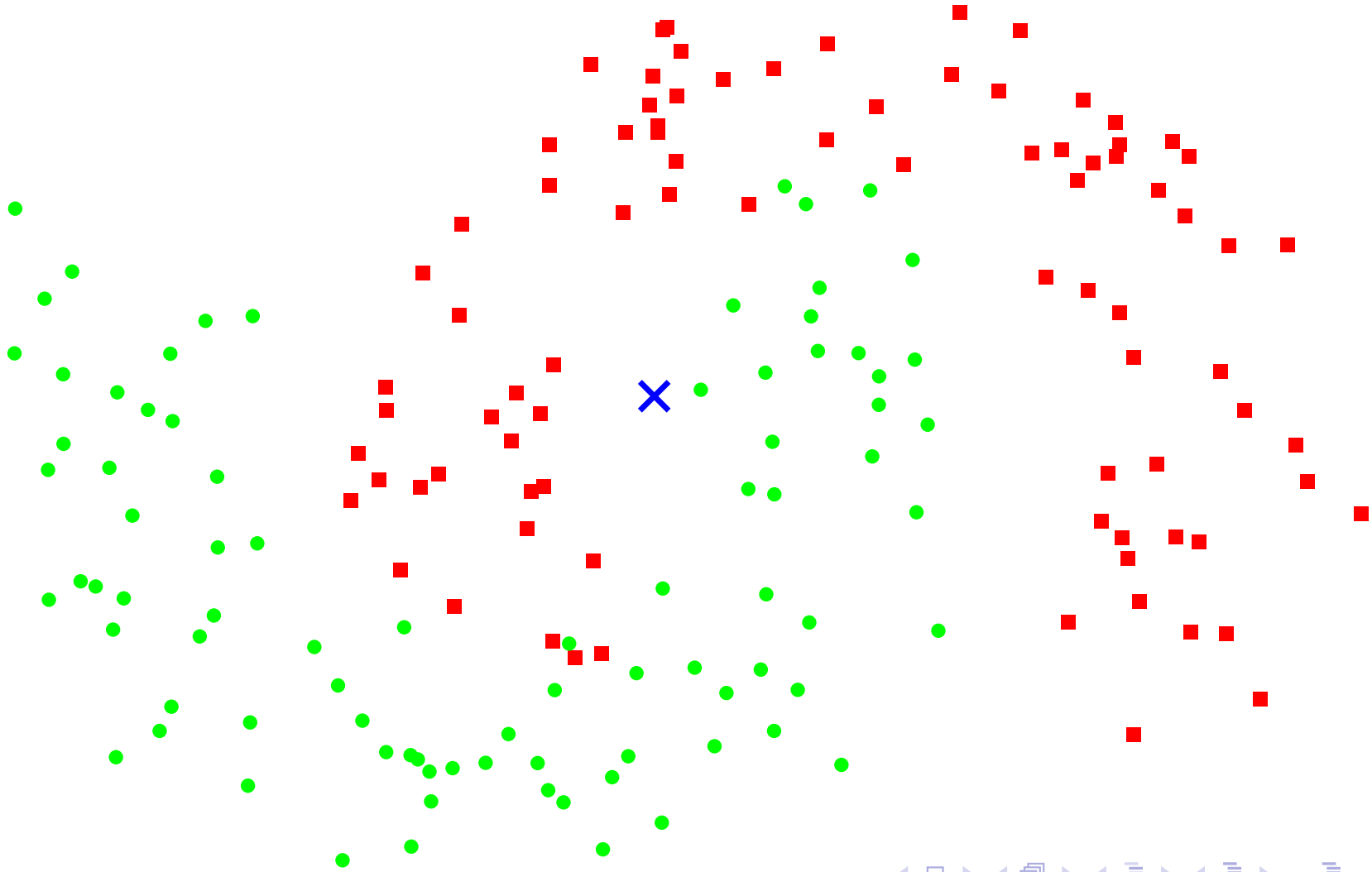
The traditional global approach with SVM

3. The class of testing samples are predicted with the same global model



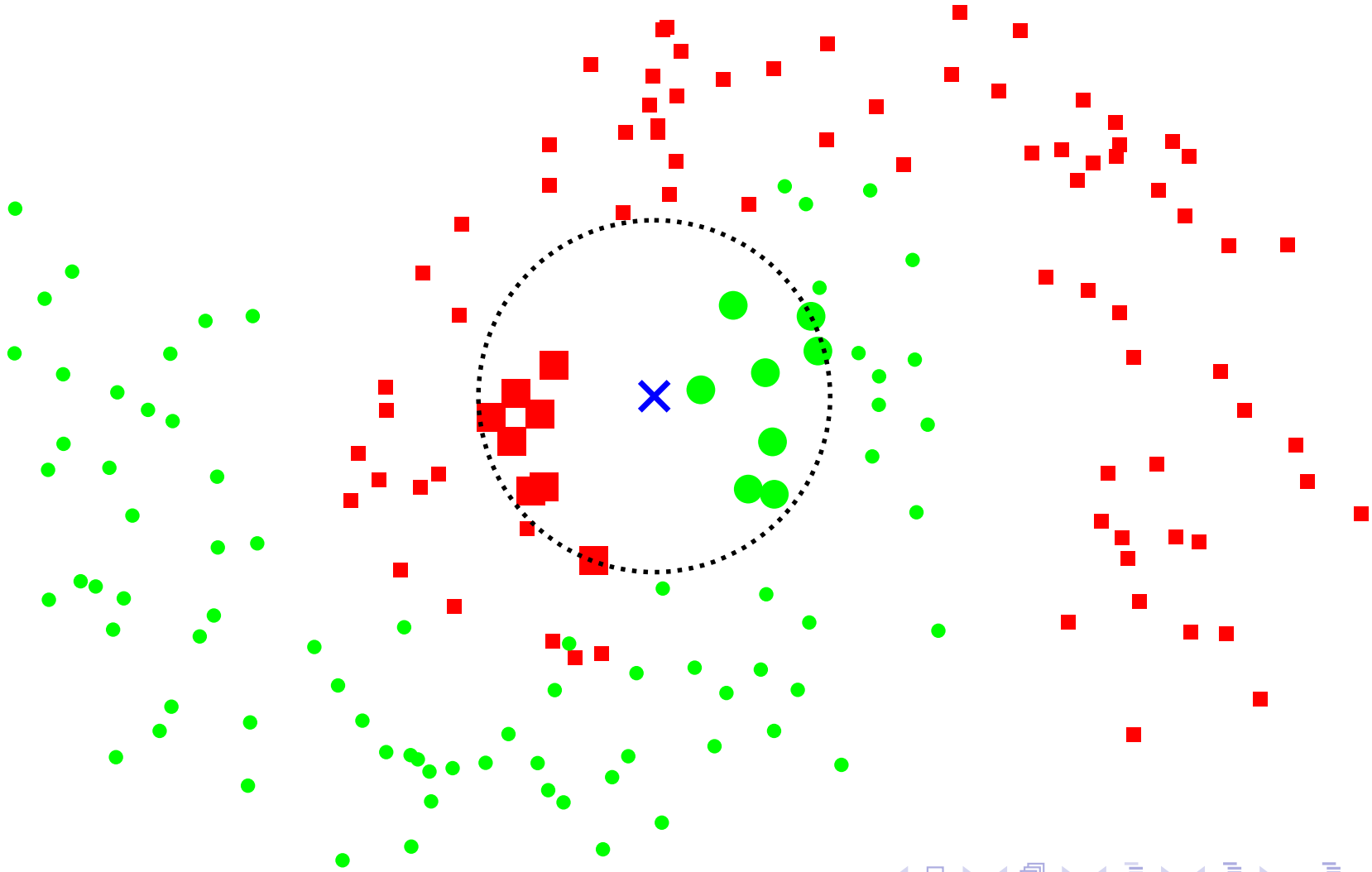
The Local SVM approach with the *k*NNSVM algorithm

1. The testing sample is available before building the model



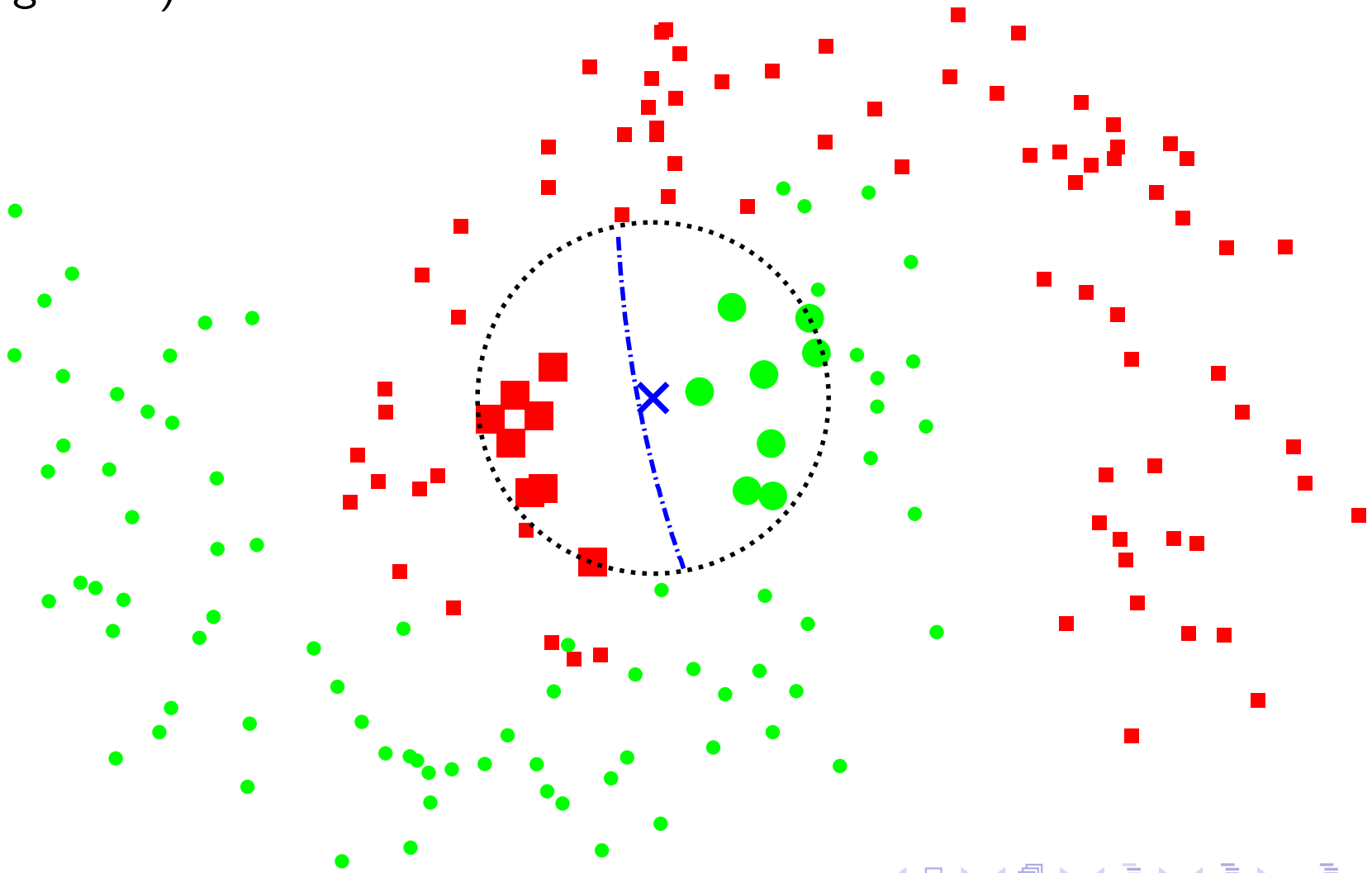
The Local SVM approach with the *k*NNSVM algorithm

2. The neighborhood of the test point is retrieved ($k = 15$)



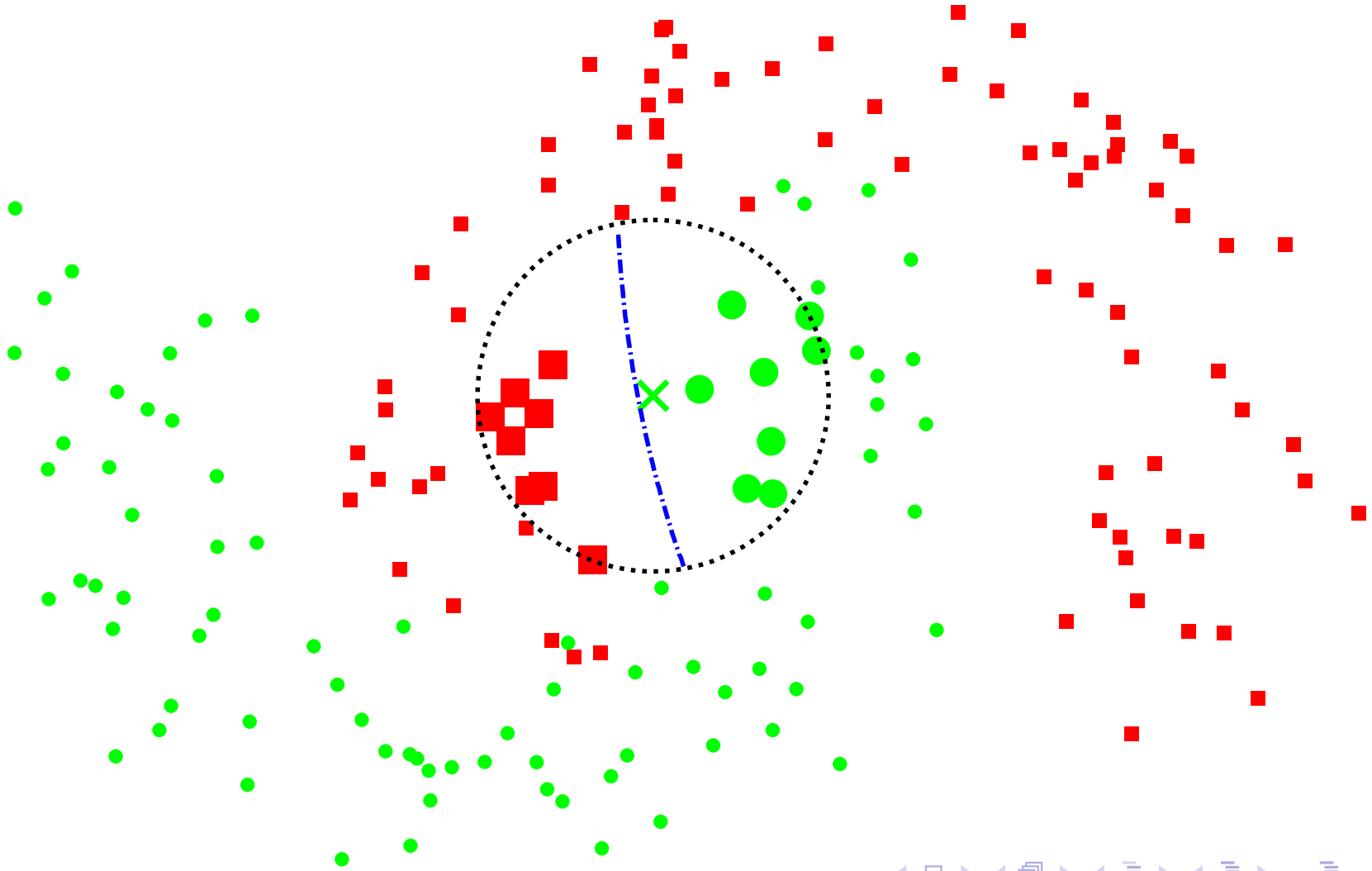
The Local SVM approach with the *k*NNSVM algorithm

3. An SVM model is trained on the neighborhood of the testing point (C = 10, g = 10)



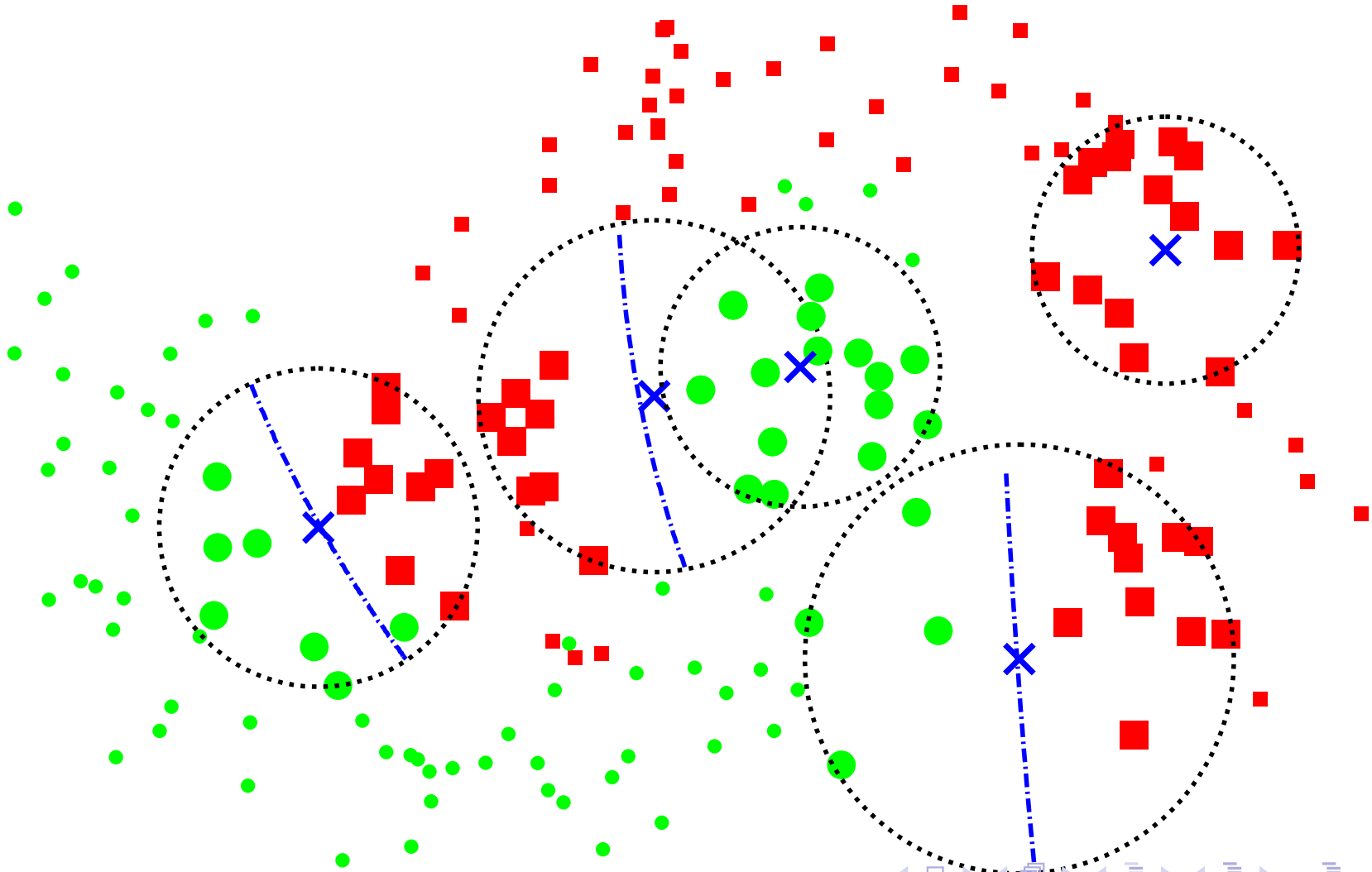
The Local SVM approach with the *k*NNSVM algorithm

4. The class of the testing point is predicted using the Local SVM model



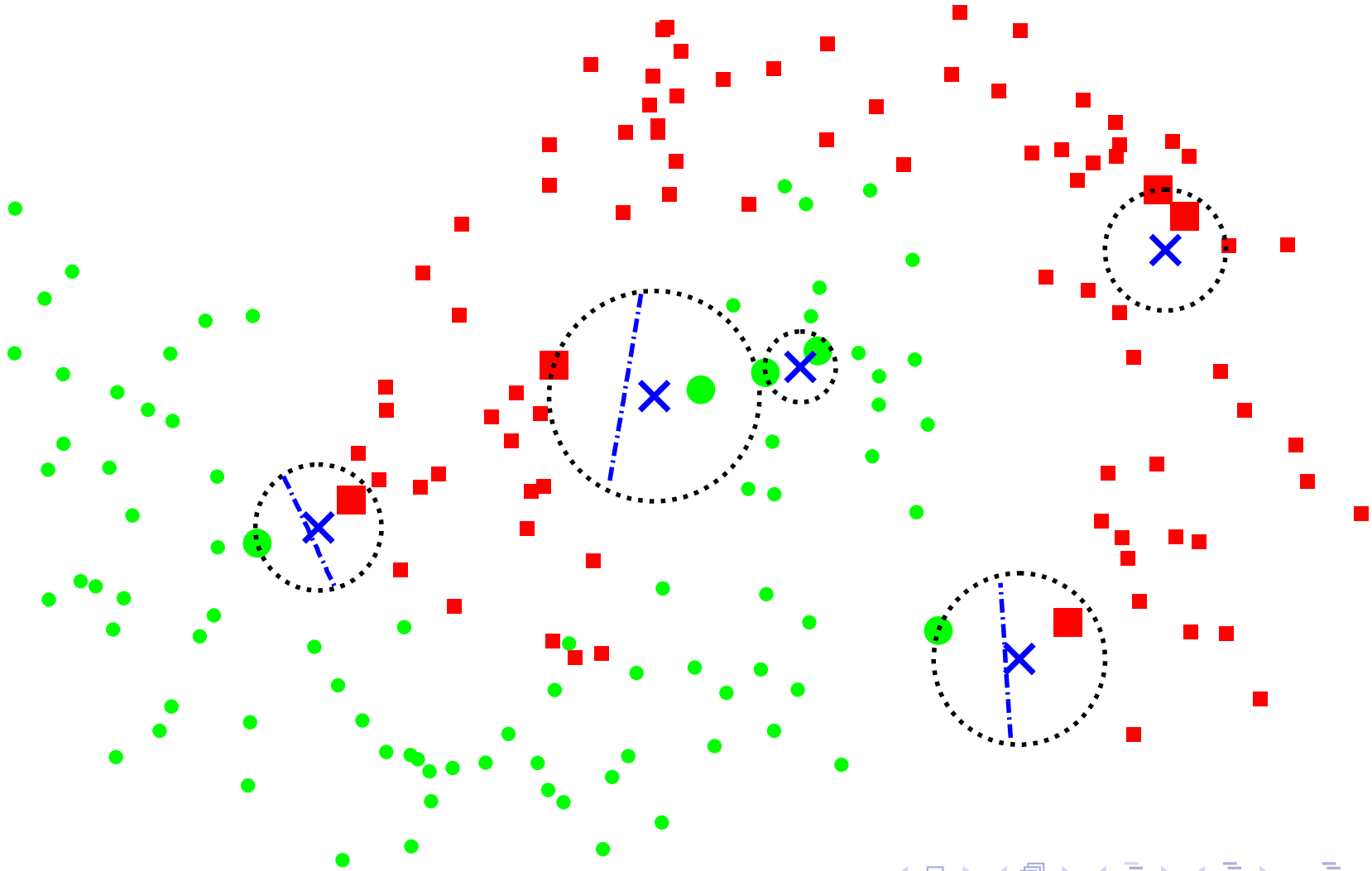
Local SVM and the k-nearest neighbors

- Locally the SVM rule can reduce to the majority rule of kNN



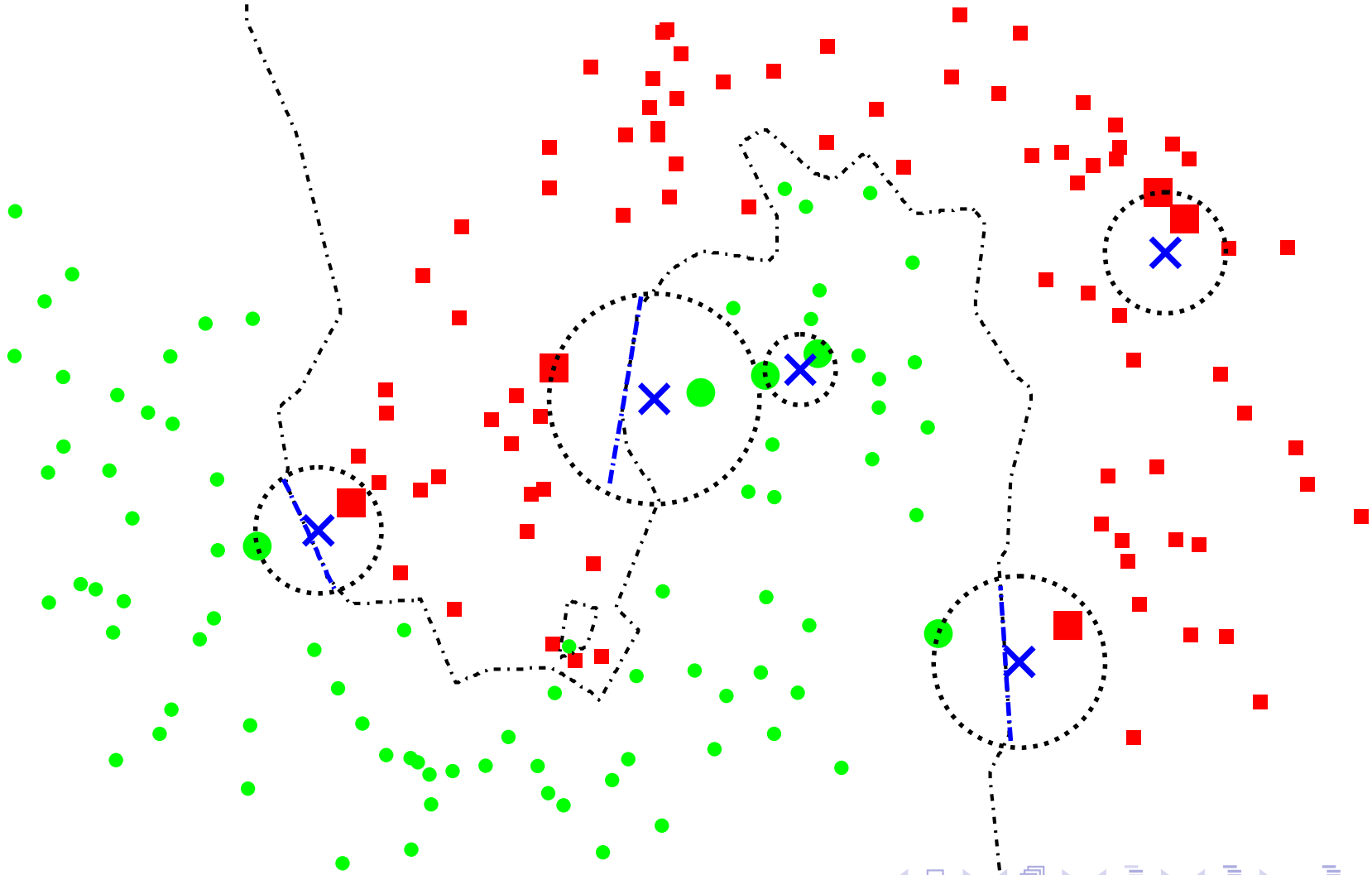
With $k = 2$ kNNSVM is equivalent to 1NN!

- With $k = 2$ kNNSVM is equivalent to 1NN!



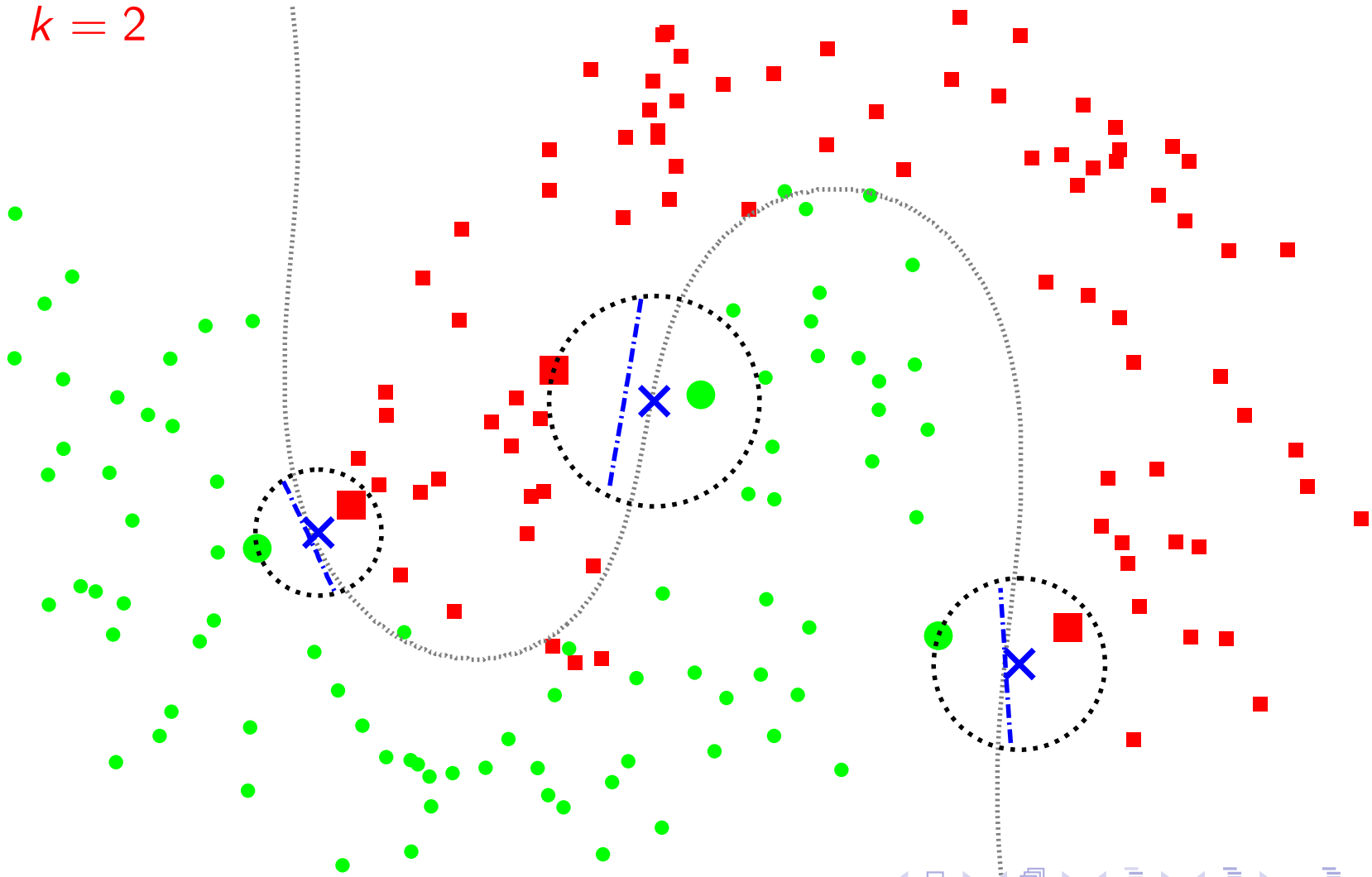
With $k = 2$ kNNSVM is equivalent to 1NN!

- With $k = 2$ kNNSVM is equivalent to 1NN!



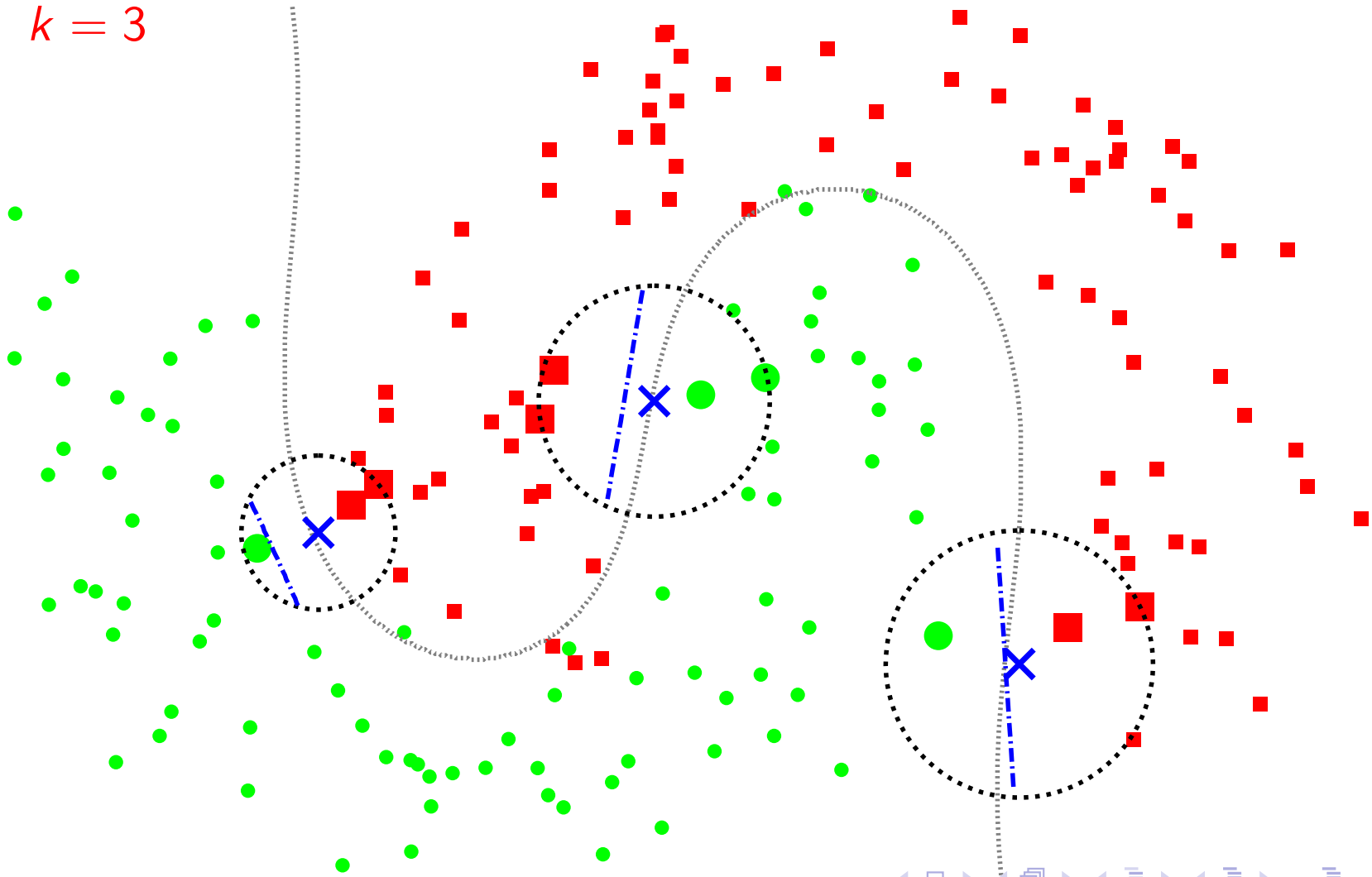
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



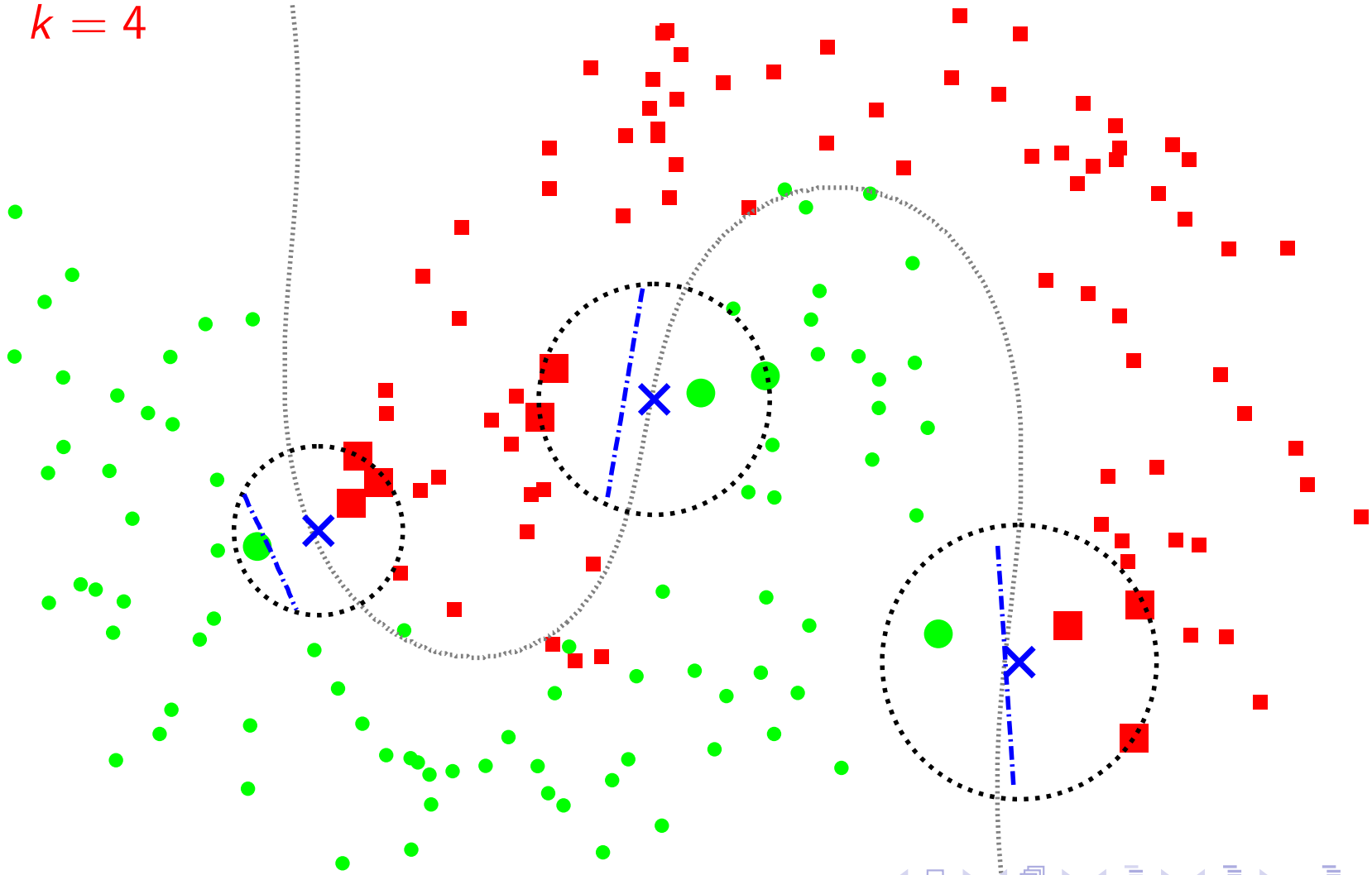
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



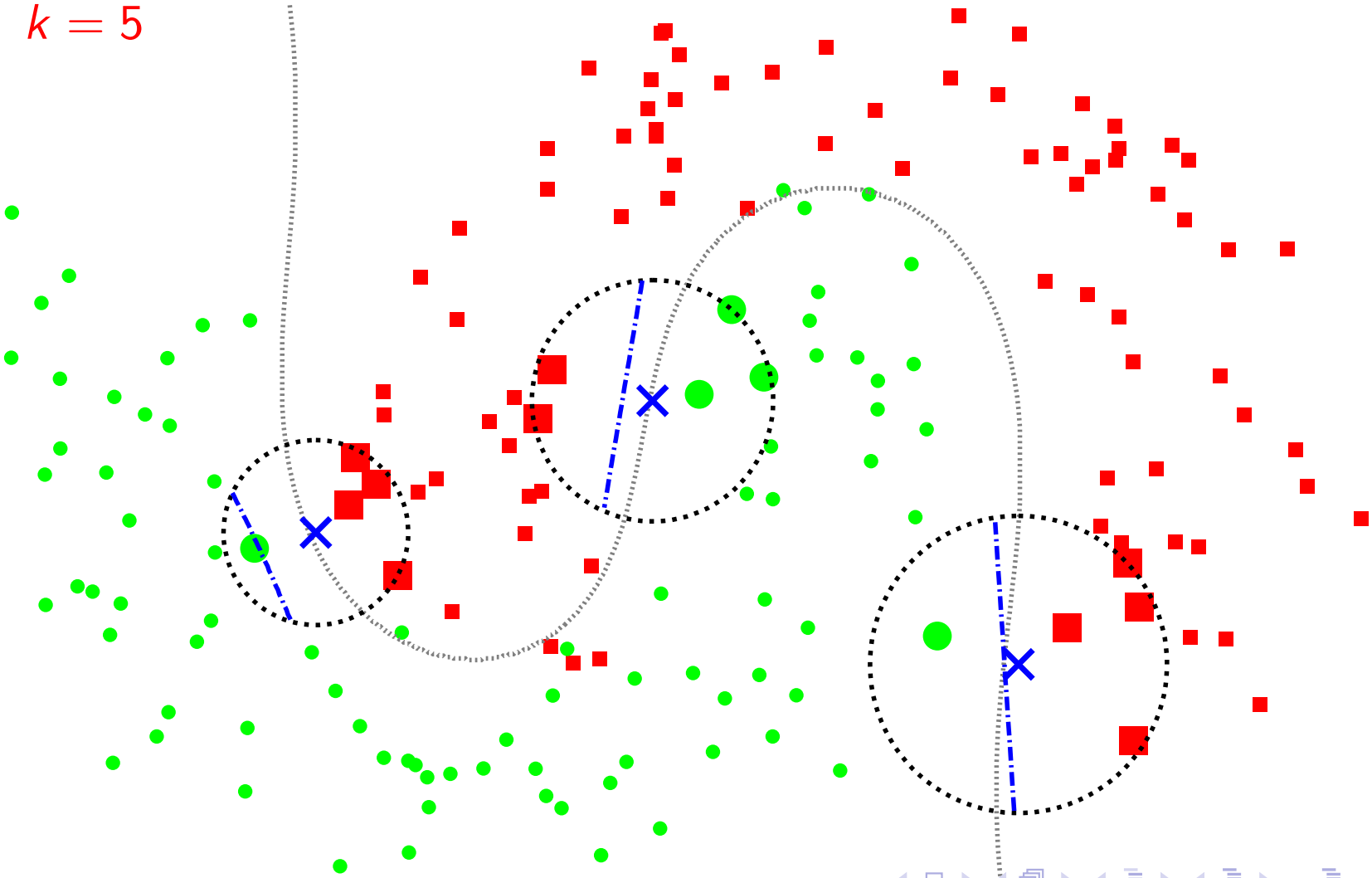
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



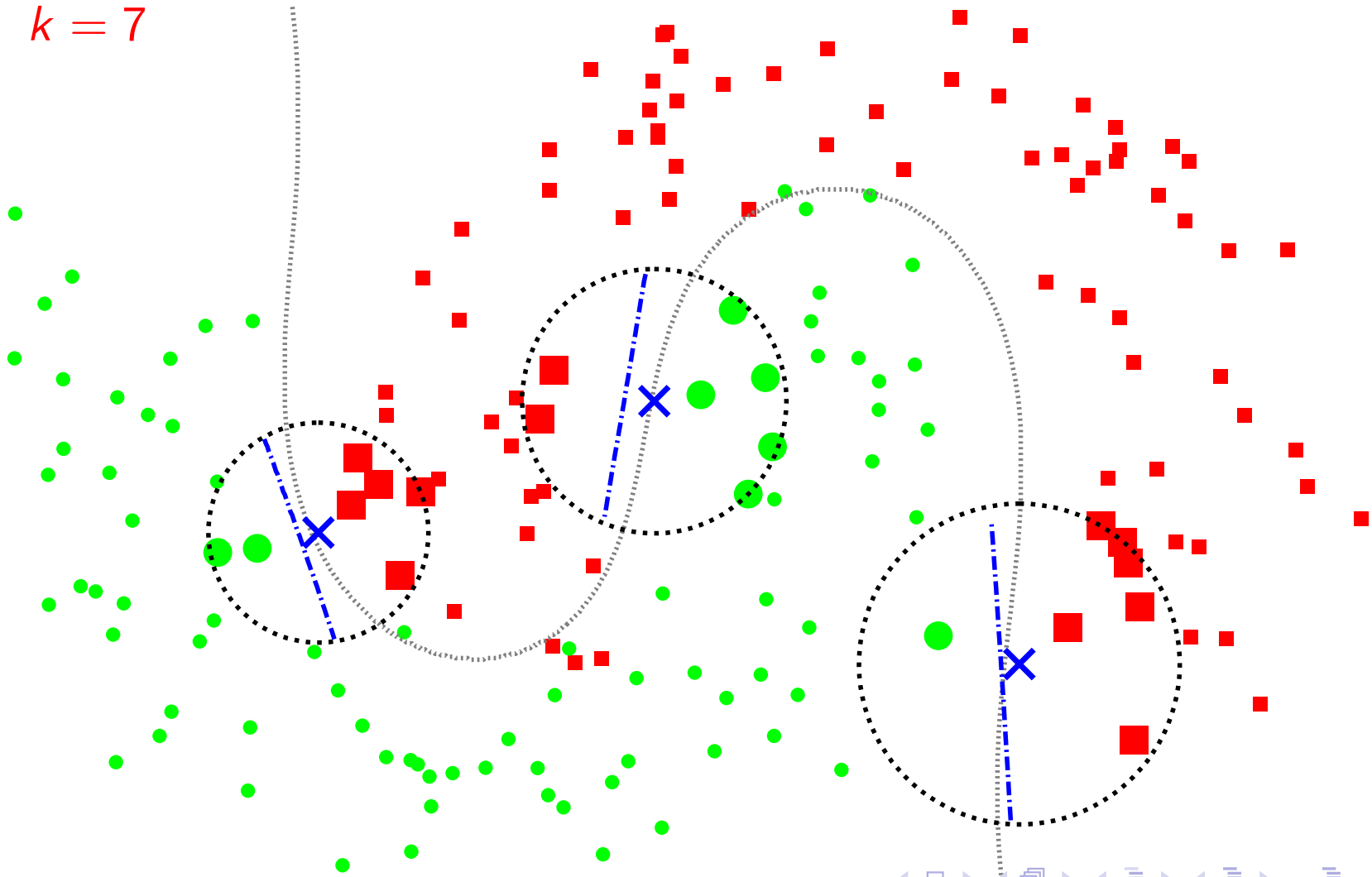
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



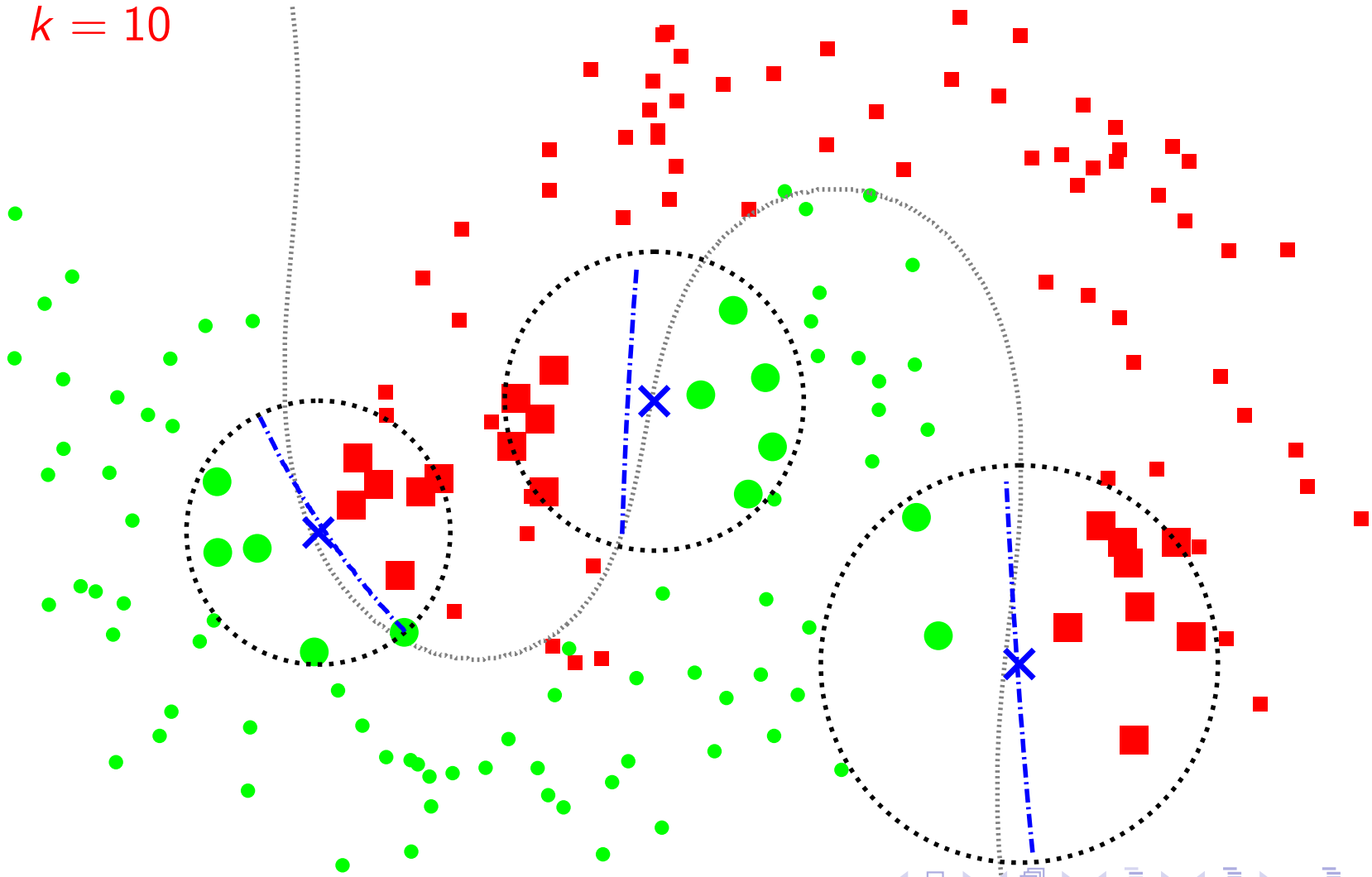
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



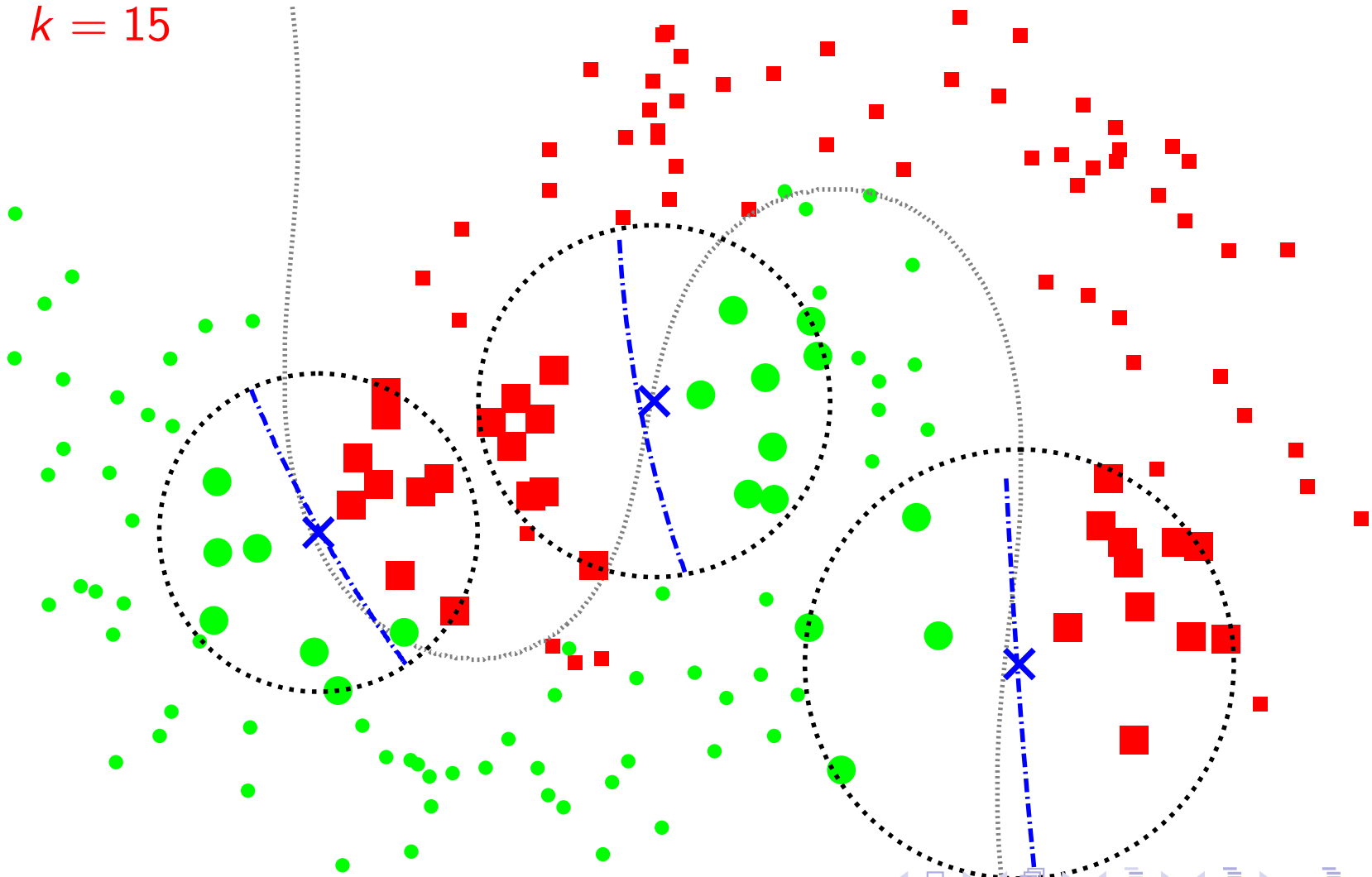
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



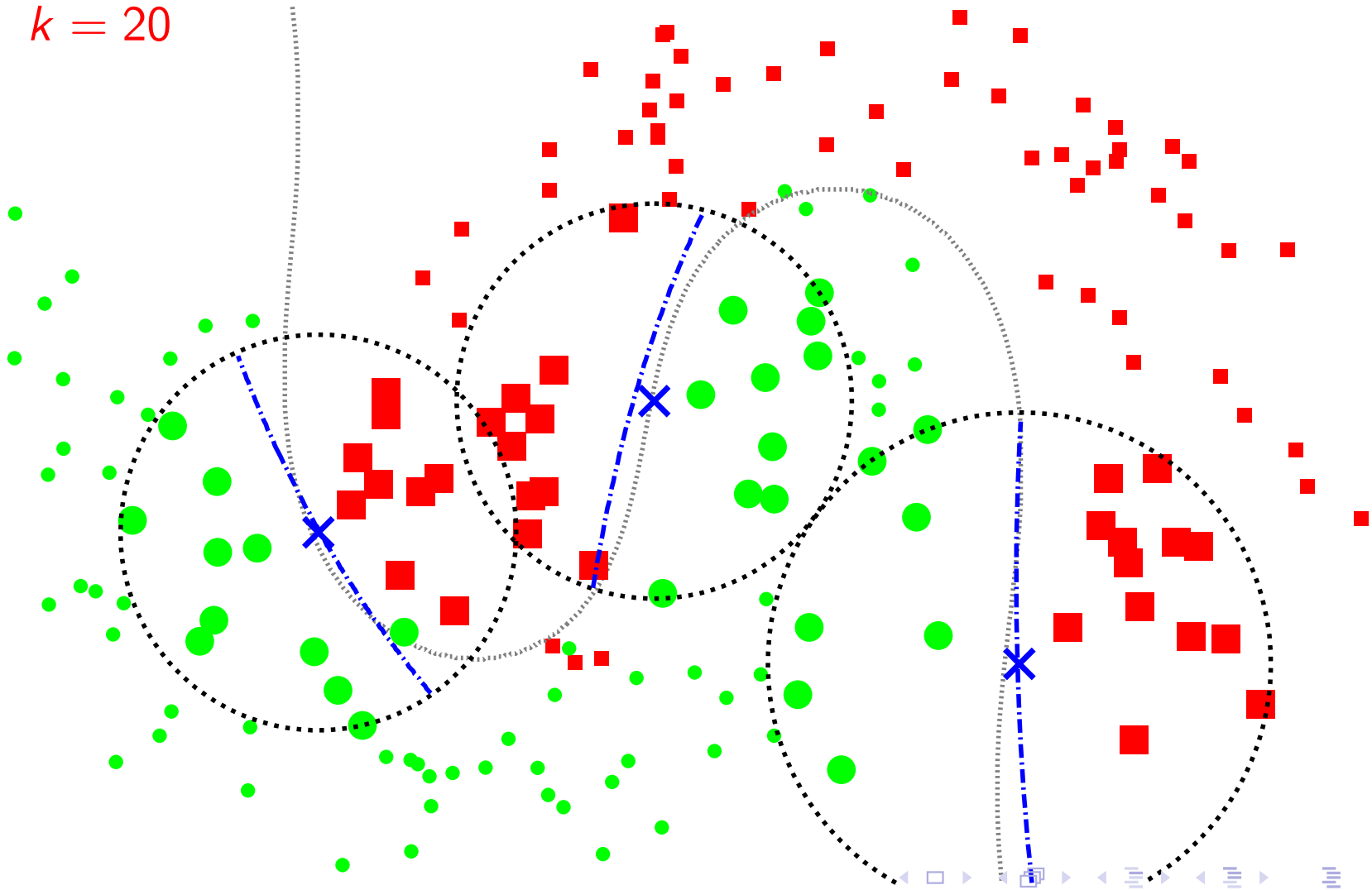
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



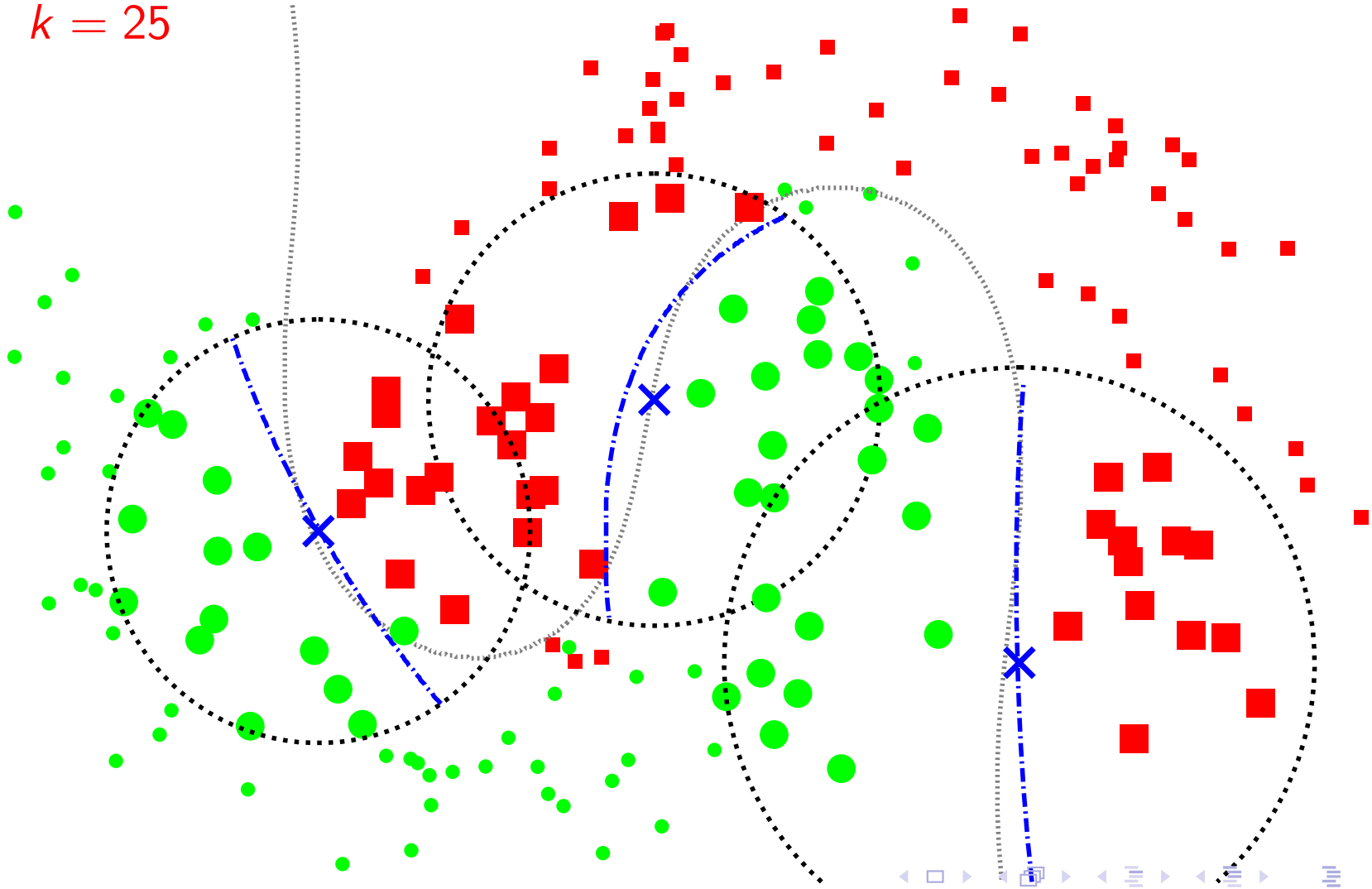
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



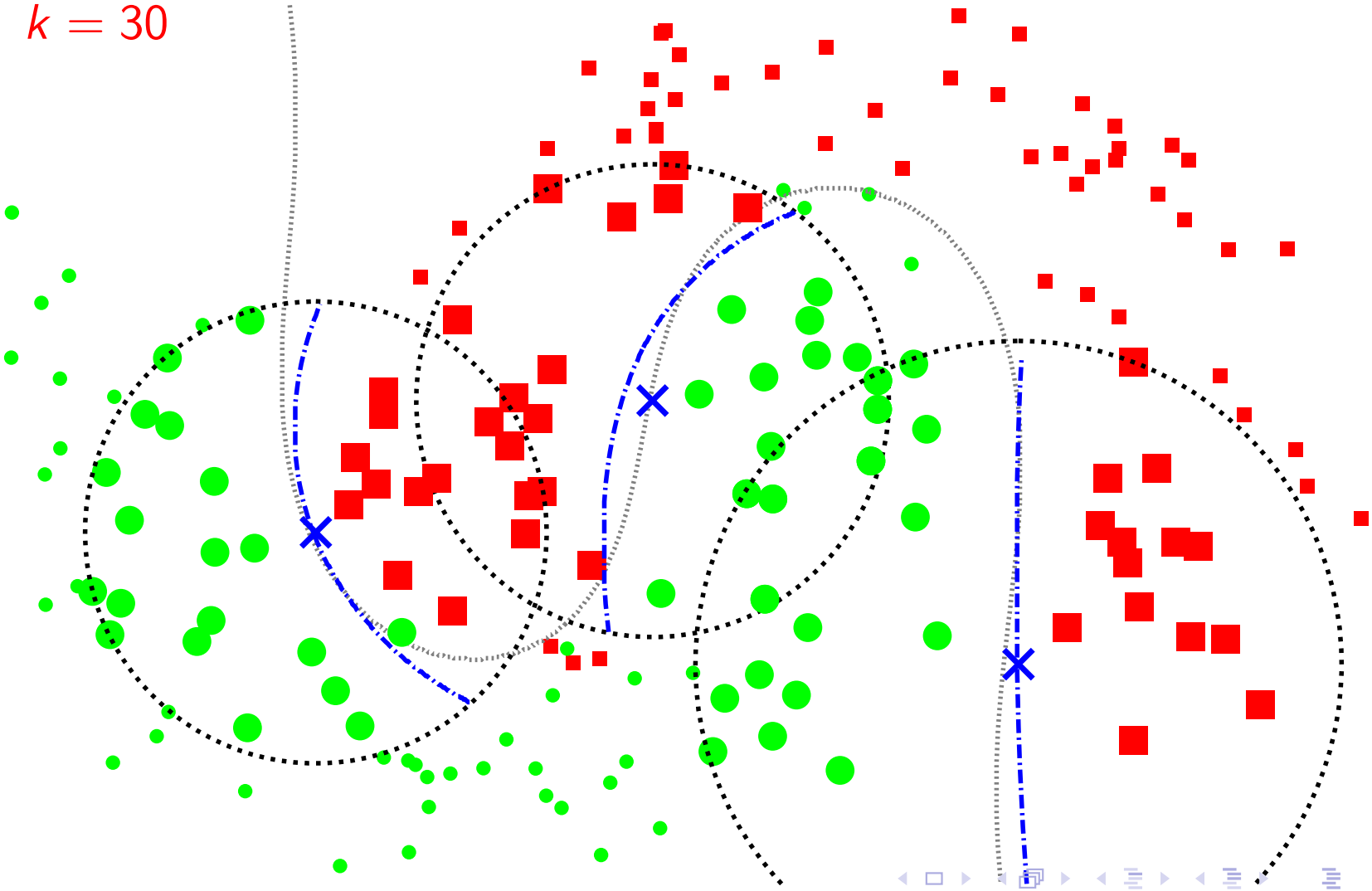
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



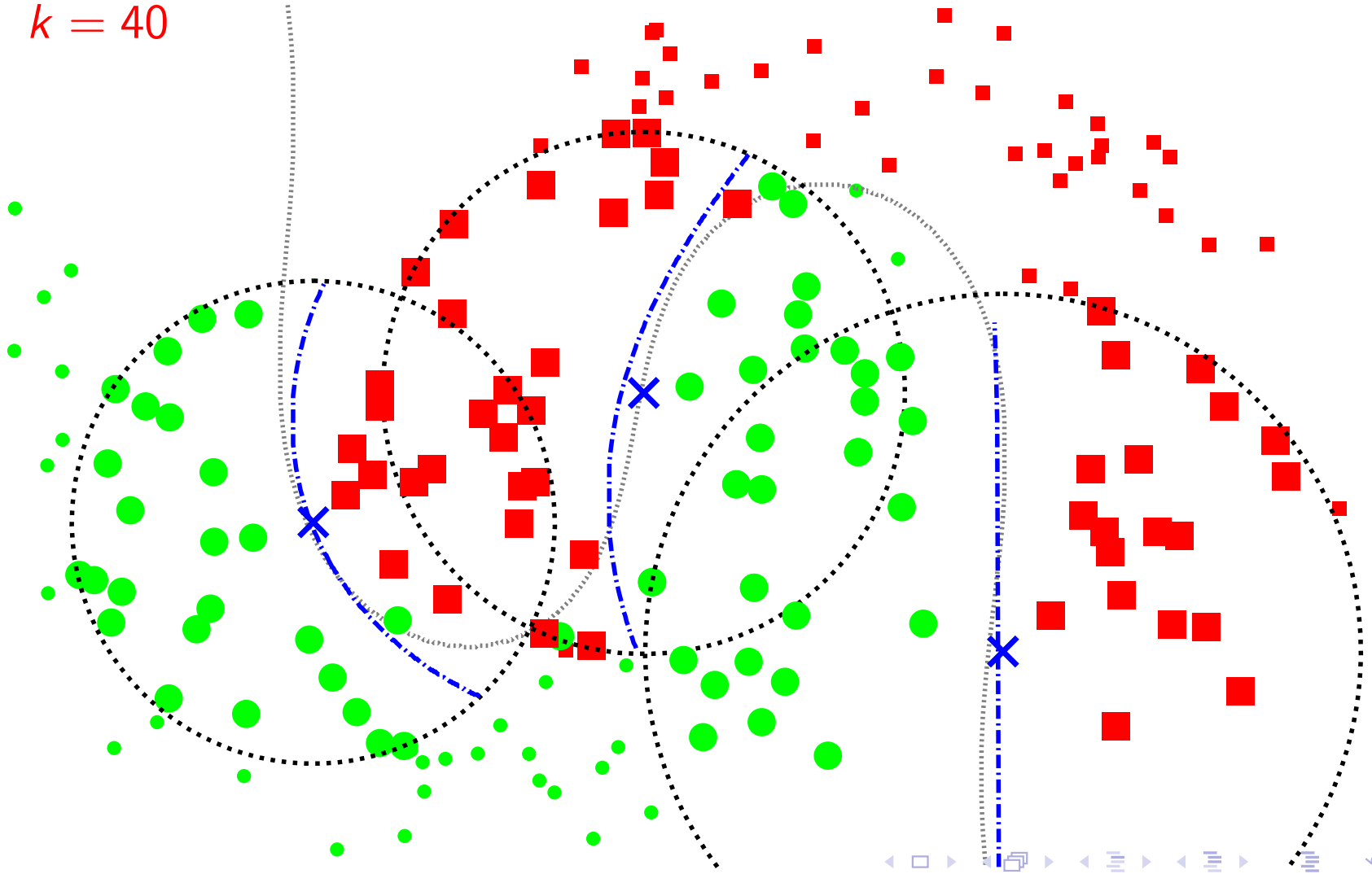
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



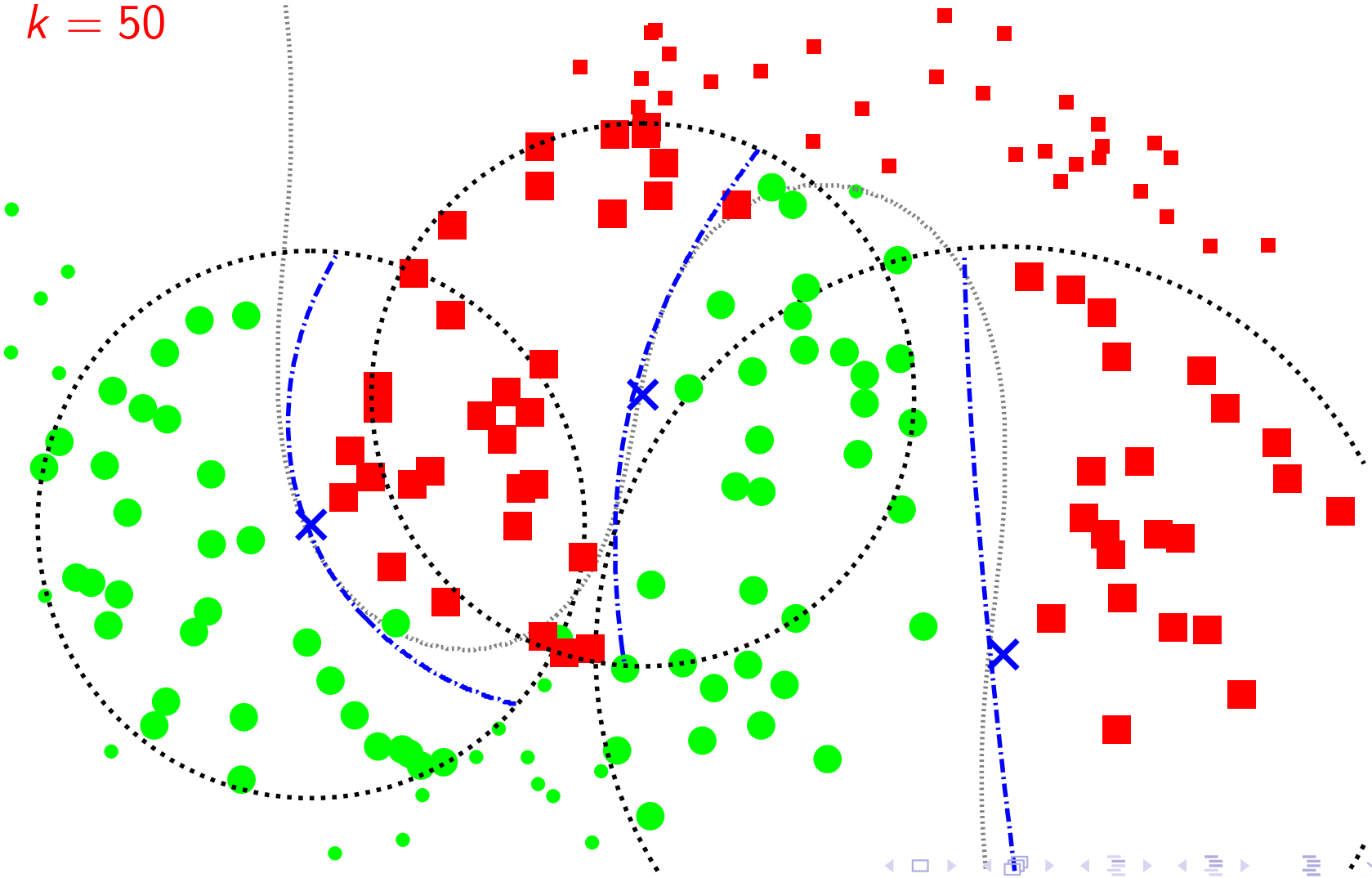
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



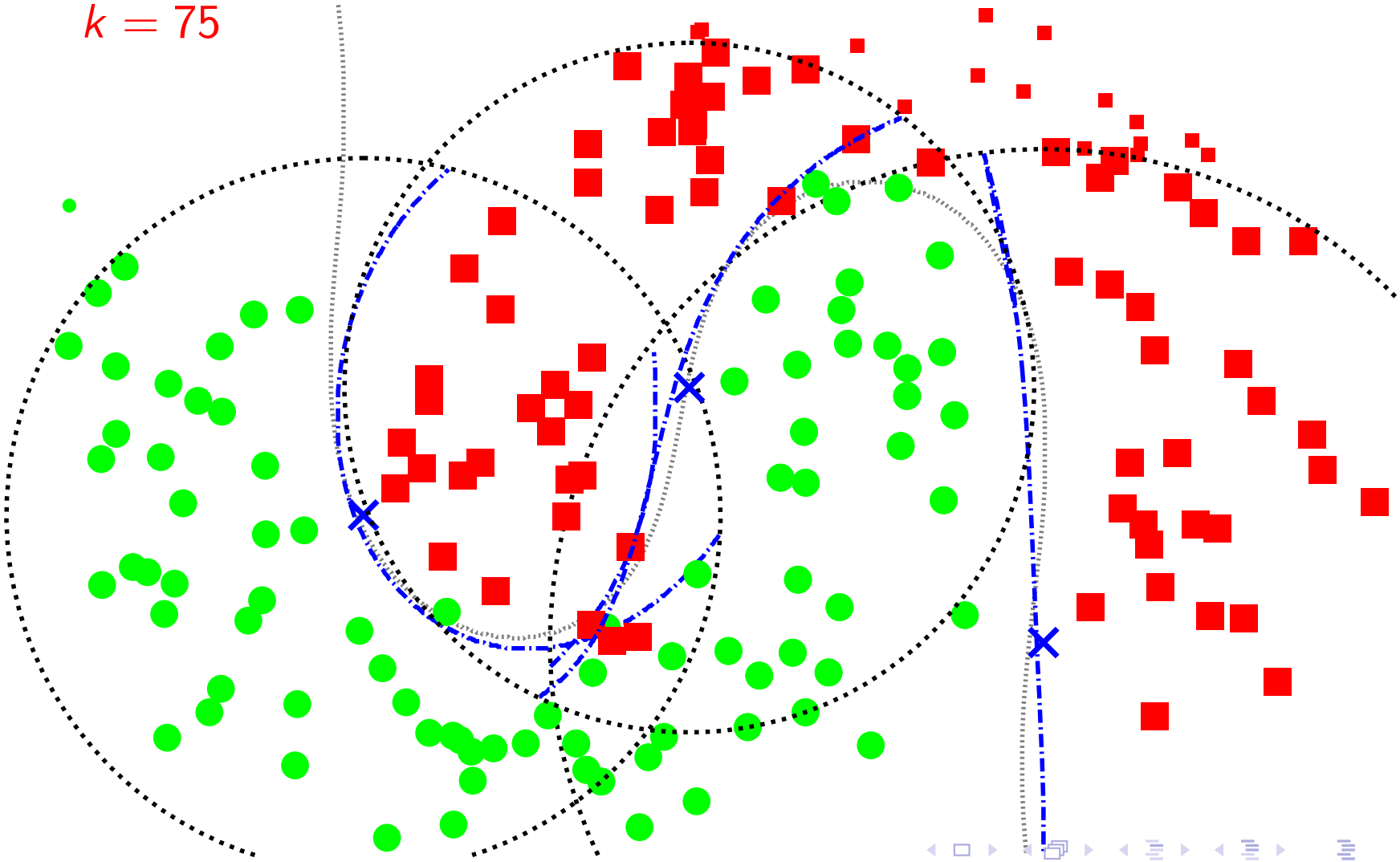
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



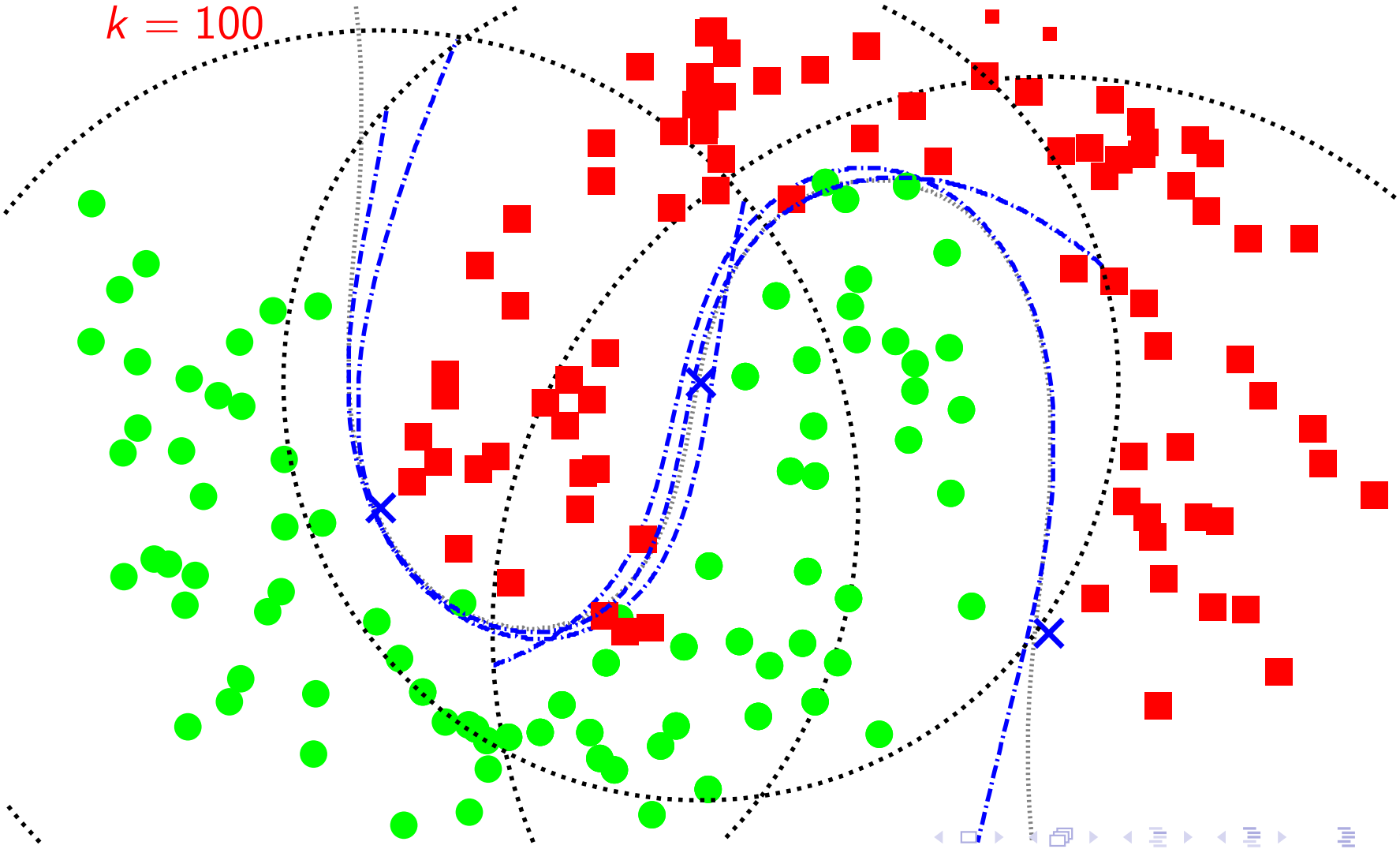
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



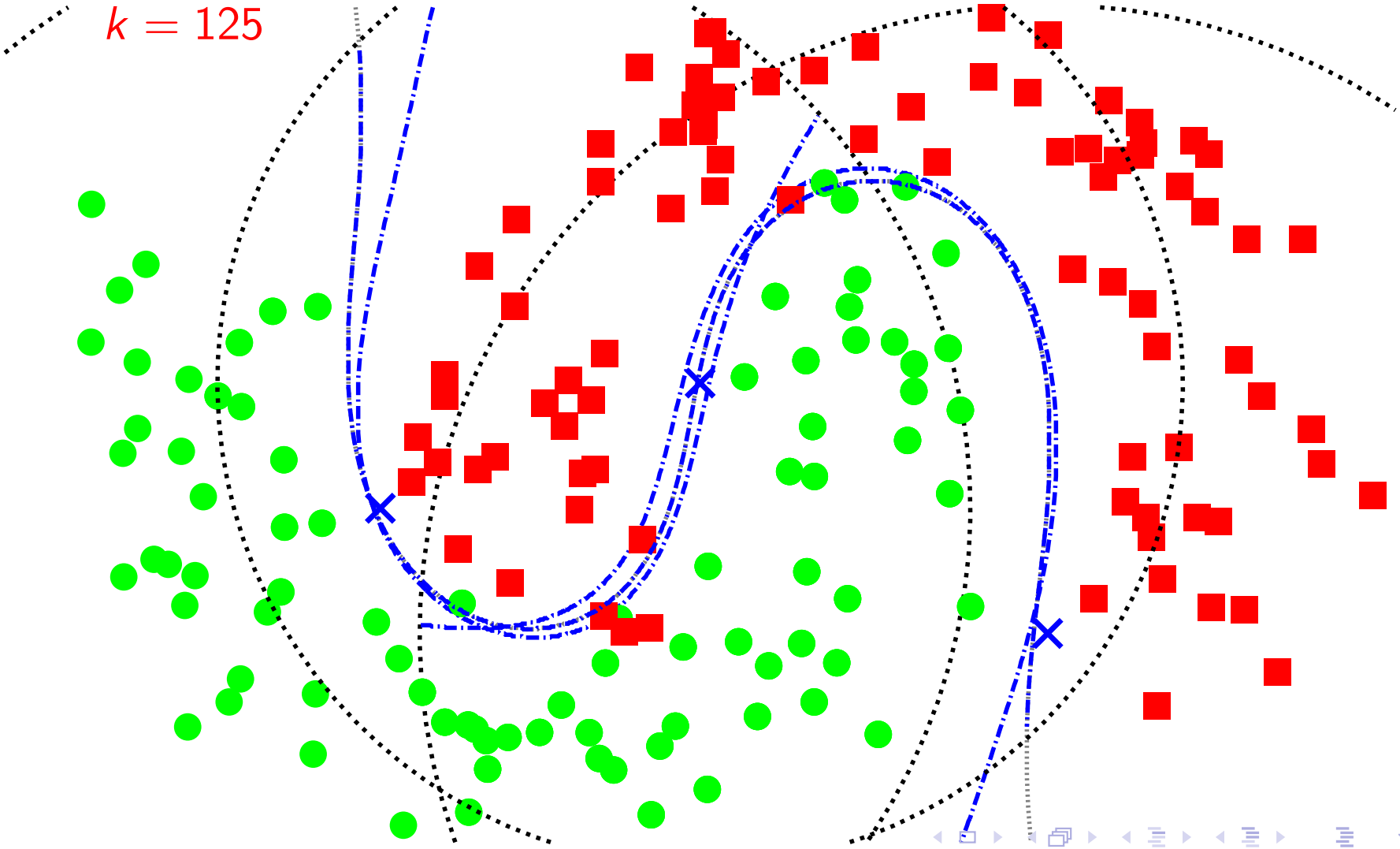
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



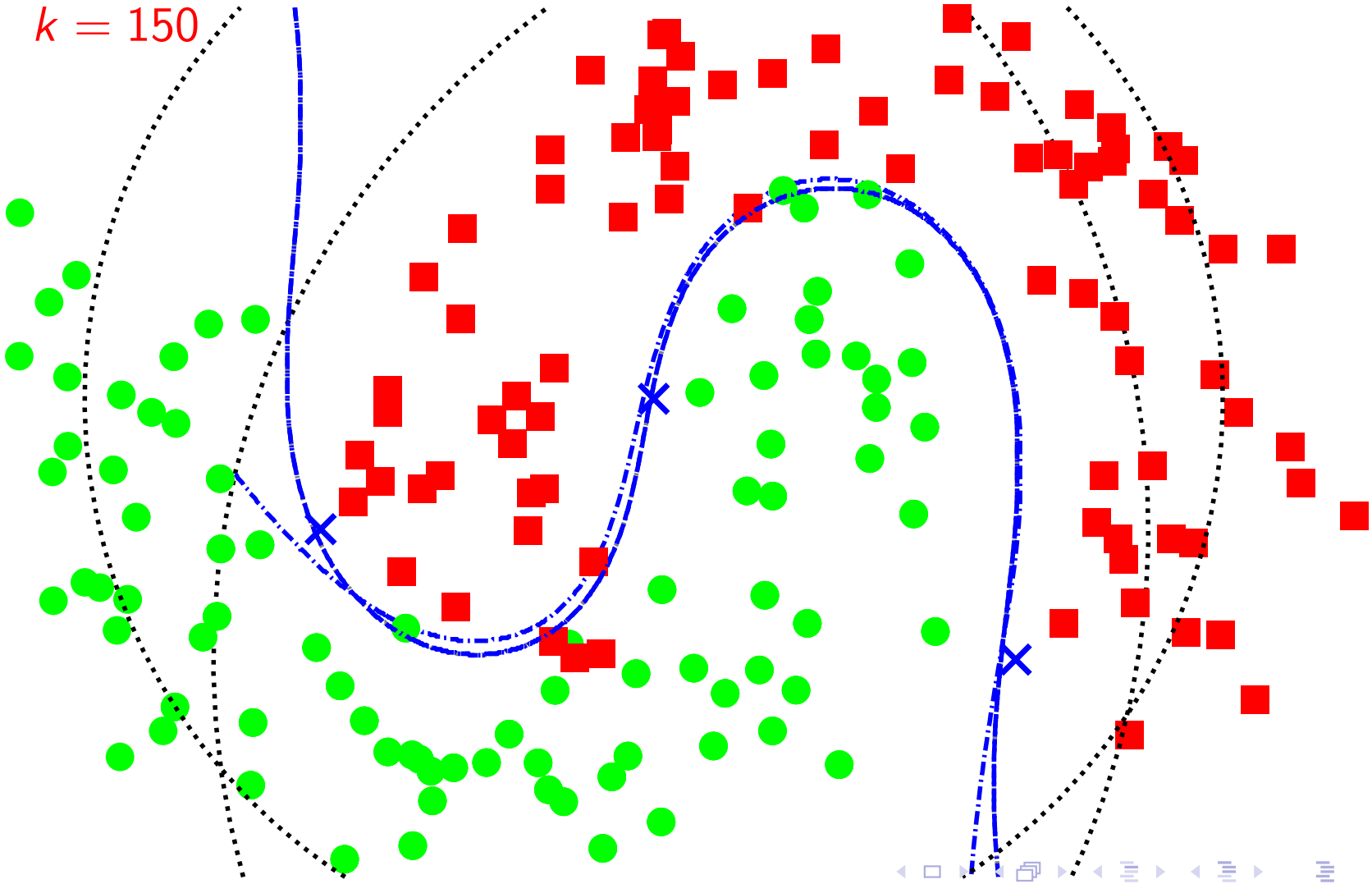
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



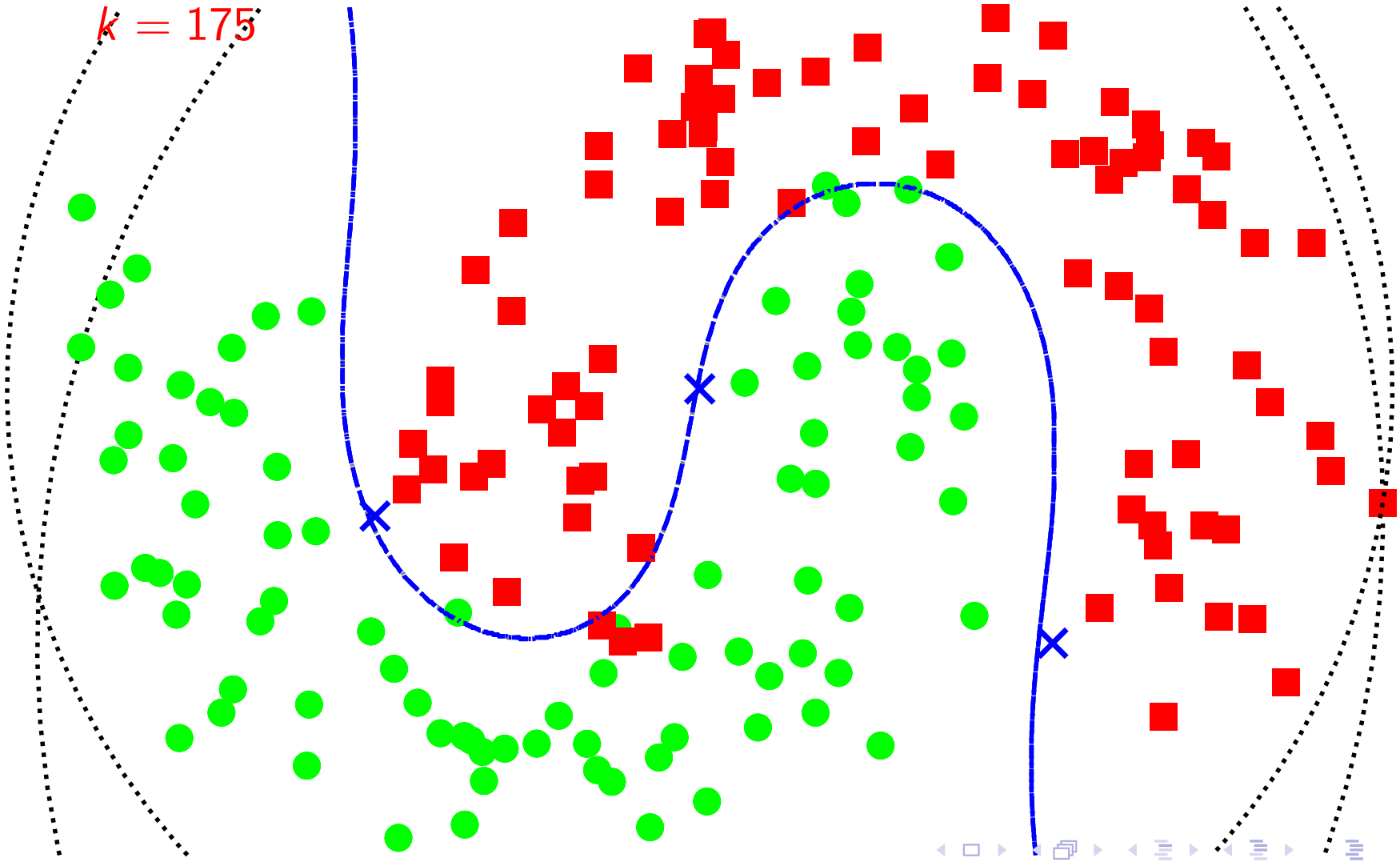
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



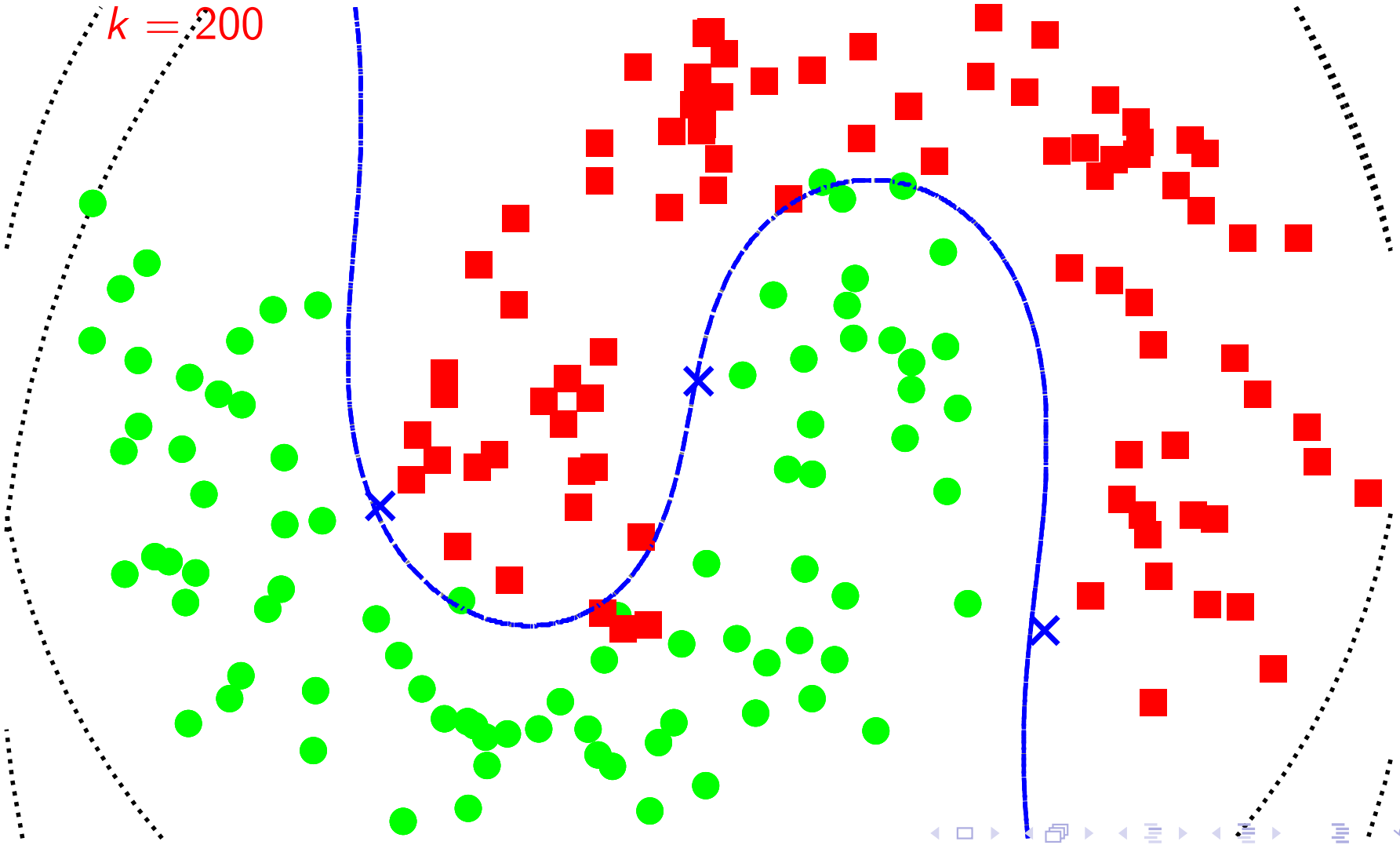
SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



SVM as a special case of Local SVM

- For $k \longrightarrow N$ kNNSVM is equivalent to SVM for each testing point



k-Nearest Neighbors and Support Vector Machines

Binary classification problem with samples (x_i, y_i), $i = 1 \dots N$, $x_i \in \mathbb{R}^p$, $y_i \in \{+1, -1\}$. $x \in \mathbb{R}^p$ is an unseen (testing) sample.

kNN decision rule

$$k\text{NN}(x) = \text{sign}\left(\sum_{i=1}^{k} y_{r_x(i)}\right).$$

Linear SVM decision rule

$$\text{Linear SVM}(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i \langle x_i, x\rangle + b\right).$$

where the Lagrange multipliers α_i and the constant b come from the dual optimization SVM problem

Accessing the feature-space through the kernel

Non-linear SVM decision rule

$$\text{SVM}(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right).$$

Common positive-definite kernel functions

$$\begin{array}{rclrcl} K^{lin}(x,x') & = & \langle x, x' \rangle & K^{rbf}(x,x') & = & \exp \frac{\|x-x'\|^2}{\sigma} \\ K^{hpol}(x,x') & = & \langle x, x' \rangle^{\delta} & K^{ipol}(x,x') & = & (\langle x, x' \rangle + 1)^{\delta} \end{array}$$

Feature-space distance

$$\begin{aligned} ||\Phi(x) - \Phi(x')||^2 &= \langle \Phi(x), \Phi(x) \rangle_{\mathcal{F}} + \langle \Phi(x'), \Phi(x') \rangle_{\mathcal{F}} - 2\langle \Phi(x), \Phi(x') \rangle_{\mathcal{F}} = \\ &= K(x,x) + K(x',x') - 2K(x,x'). \end{aligned}$$

kNNSVM: the algorithm for Local SVM

kNNSVM decision rule

$$kNNSVM(x) = \text{sign}\left(\sum_{i=1}^{k} \alpha_{r_x(i)} y_{r_x(i)} K(x_{r_x(i)}, x) + b\right)$$

where

- $r_{x'}$: $\{1, \ldots, N\} \rightarrow \{1, \ldots, N\}$ is a function that reorders the training set w.r.t x using the feature space distance $\|\Phi(x_i) - \Phi(x')\|$
- $\alpha_{r_x(i)}$ and b come from the dual optimization SVM problem using the neighborhood points as training set

Theoretical properties

Theoretically Local SVM can perform better than SVM because:

- "Selecting a non trivial value for the locality parameter β might reduce the generalization error induced by the unavoidable inaccuracy of the parameter α (the model)" [Vapnik and Bottou, 1993]. Arguments based on Local Structural Risk Minimization.
- Since Local SVM can find a lower Radious/Margin Bound for some k, "an appropriate choice of k can lead to improved generalization with respect to the SVM" [Blanzieri and Melgani, 2006].

The need for an empirical assessment of the approach

Nice properties and theoretical arguments but...

What about classification ability of Local SVM in practice?

Is Local SVM better than SVM?
Are further developments of the Local SVM approach promising?

- No extensive empirical analysis performed yet
- no direct comparison between local learning and SVM with local kernels performed yet
- new approaches and developments based on locality should rely also on empirical analysis

Empirical comparison between Local SVM and SVM

Our empirical analysis has the purpose to compare the generalization capability of Local SVM and SVM on different datasets and kernels

The empirical assessment of Local SVM

The experimental procedure:

- comparison between kNNSVM and SVM on 13 real datasets
- evaluation performed using the 10-fold cross validation (CV) classification accuracies
- four kernel function used: K^{lin}, K^{ipol}, K^{hpol} and K^{rbf}
- model selection (on each fold) with 10-fold CV grid search
  - C of SVM chosen in $\{1, 5, 10, 25, 50, 75, 100, 150, 300, 500\}$
  - σ of the RBF kernel among $\{2^{-10}, 2^{-9}, \ldots, 2^{9}, 2^{10}\}$
  - δ of polynomial kernels is bounded to 5.
  - k of kNNSVM in $\{1, 3, 5, 7, 9, 11, 15, 23, 39, 71, 135, 263, 519, |training\_set|\}$.
- one-against-all strategy for multi-class classification problems
- statistical significance assessment using the two-tailed paired t-test ($\alpha = 0.05$) on the two sets of fold accuracies

The 13 general real datasets used for the comparison

NAME	SOURCE	#CLASSES	#TR SAMPLES	#FEAURES
IRIS	UCI	3	150	4
WINE	UCI	3	178	13
LEUKEMIA	TG99	2	38	7129
LIVER	UCI	2	345	6
SVMGUIDE2	CWH03A	3	391	20
VEHICLE	STATLOG	4	846	18
VOWEL	UCI	11	528	10
BREAST	UCI	2	683	10
FOURCLASS	TKH96A	2	862	2
GLASS	UCI	6	214	9
HEART	STATLOG	2	270	13
IONOSPHERE	UCI	2	351	34
SONAR	UCI	2	208	60

Best results for each dataset

DATASET	SVM				*k*NNSVM			
	K^{lin}	K^{ipol}	K^{hlin}	K^{rbf}	K^{lin}	K^{ipol}	K^{hlin}	K^{rbf}
IRIS	0.967	**0.973**	**0.973**	0.947	0.960	0.967	0.960	0.960
WINE	0.966	0.966	0.966	**0.994**	0.983	**0.994**	0.989	0.989
LEUKEMIA	**0.950**	**0.950**	**0.950**	0.708	0.925	0.925	0.925	0.925
LIVER	0.681	0.701	0.713	0.722	**0.739**	0.733	**0.739**	0.728
SVMGUIDE2	0.816	0.826	0.816	0.836	**0.859**	0.857	0.841	0.844
VEHICLE	0.799	0.847	0.837	0.849	**0.861**	0.848	0.857	0.840
VOWEL	0.837	0.989	0.979	0.992	**0.998**	**0.998**	**0.998**	**0.998**
BREAST	0.968	0.968	0.968	0.968	0.966	0.962	0.965	**0.971**
FOURCLASS	0.768	0.998	0.811	0.999	**1.000**	**1.000**	**1.000**	**1.000**
GLASS	0.622	0.701	0.720	0.687	0.692	0.706	**0.720**	0.674
HEART	0.826	0.822	0.822	**0.830**	0.822	0.822	0.822	0.819
IONOSPHERE	0.869	0.912	0.892	**0.937**	0.929	0.929	0.929	0.935
SONAR	0.779	0.875	0.880	0.894	0.875	0.890	0.890	**0.904**

Comparison with Linear Kernel $K^{lin}(x, x') = \langle x, x' \rangle$

DATASET	SVM	kNNSVM	DIFF	$p < 0.05$
IRIS	0.967	0.960	−0.007	
WINE	0.966	0.983	+0.017	
LEUKEMIA	0.950	0.925	−0.025	
LIVER	0.681	0.739	+0.058	√
SVMGUIDE2	0.816	0.859	+0.043	√
VEHICLE	0.799	0.861	+0.061	√
VOWEL	0.837	0.998	+0.161	√
BREAST	0.968	0.966	−0.001	
FOURCLASS	0.768	1.000	+0.232	√
GLASS	0.622	0.692	+0.071	√
HEART	0.826	0.822	−0.004	
IONOSPHERE	0.869	0.929	+0.060	√
SONAR	0.779	0.875	+0.096	√

Comparison with IPol Kernel $K^{ipol}(x, x') = (\langle x, x' \rangle + 1)^{\delta}$

DATASET	SVM	kNNSVM	DIFF	$p < 0.05$
IRIS	0.973	0.967	−0.007	
WINE	0.966	0.994	+0.028	√
LEUKEMIA	0.950	0.925	−0.025	
LIVER	0.701	0.733	+0.032	√
SVMGUIDE2	0.826	0.857	+0.031	√
VEHICLE	0.847	0.848	+0.001	
VOWEL	0.989	0.998	+0.009	√
BREAST	0.968	0.962	−0.006	
FOURCLASS	0.998	1.000	+0.002	
GLASS	0.701	0.706	+0.006	
HEART	0.822	0.822	0.000	
IONOSPHERE	0.912	0.929	+0.017	
SONAR	0.875	0.890	+0.015	

Comparison with HPol Kernel $K^{hpol}(x, x') = \langle x, x' \rangle^{\delta}$

DATASET	SVM	kNNSVM	DIFF	$p < 0.05$
IRIS	0.973	0.960	−0.013	
WINE	0.966	0.989	+0.023	√
LEUKEMIA	0.950	0.925	−0.025	
LIVER	0.713	0.739	+0.026	√
SVMGUIDE2	0.816	0.841	+0.026	
VEHICLE	0.837	0.857	+0.020	√
VOWEL	0.979	0.998	+0.019	√
BREAST	0.968	0.965	−0.003	
FOURCLASS	0.811	1.000	+0.189	√
GLASS	0.720	0.720	+0.001	
HEART	0.822	0.822	0.000	
IONOSPHERE	0.892	0.929	+0.037	√
SONAR	0.880	0.890	+0.010	

Comparison with RBF kernel $K^{rbf}(x, x') = \exp \|x - x'\|^2/\sigma$

DATASET	SVM	kNNSVM	DIFF	$p < 0.05$
IRIS	0.947	0.960	+0.013	
WINE	0.994	0.989	−0.006	
LEUKEMIA	0.708	0.925	+0.217	$\surd$
LIVER	0.722	0.728	+0.006	
SVMGUIDE2	0.836	0.844	+0.008	
VEHICLE	0.849	0.840	−0.008	
VOWEL	0.992	0.998	+0.006	
BREAST	0.968	0.971	+0.003	
FOURCLASS	0.999	1.000	+0.001	
GLASS	0.687	0.674	−0.013	
HEART	0.830	0.819	−0.011	
IONOSPHERE	0.937	0.935	−0.003	
SONAR	0.894	0.904	+0.010	

Results of the comparison on real datasets

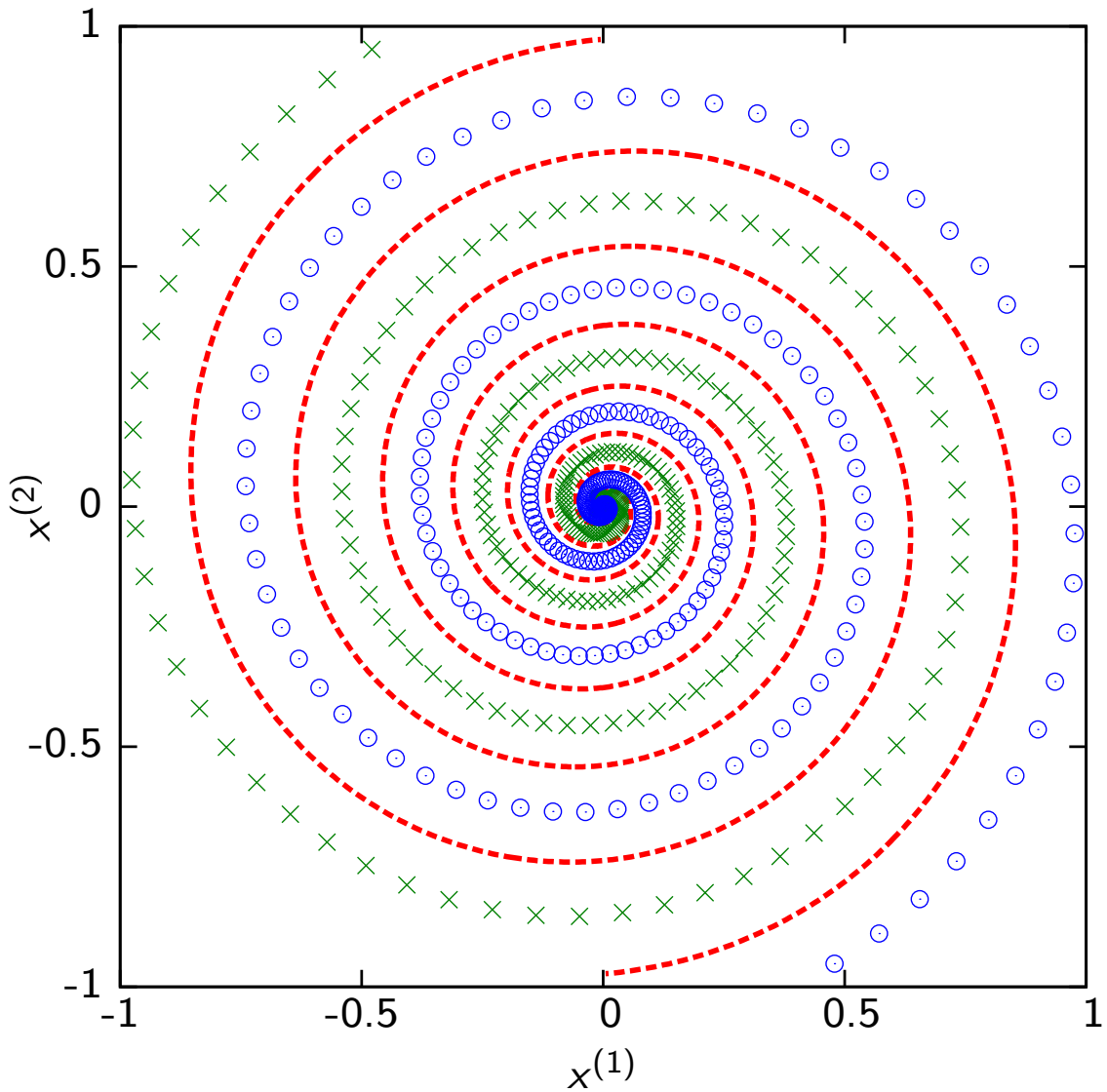
1. Local Linear SVM is more accurate than Linear SVM.
2. Local SVM is never statistically worse than SVM regardless to the kernel function
3. Local SVM with polynomial kernels is statistically more accurate than SVM with the same kernels in a number of datasets
4. It is not clear if Local SVM is more accurate than SVM with the RBF kernel for general real datasets

Local SVM can have advantages over SVM with RBF kernel?

We designed new experiments with artificial data because:

- we can control the level of noise
- we can qualitatively and graphically detect the different behaviour of the approaches

The two-spirals dataset



Parametric definition:

$$\begin{cases} x^{(1)}(t) = c \cdot t^d \cdot sin(t) \\ x^{(2)}(t) = c \cdot t^d \cdot cos(t) \end{cases}$$

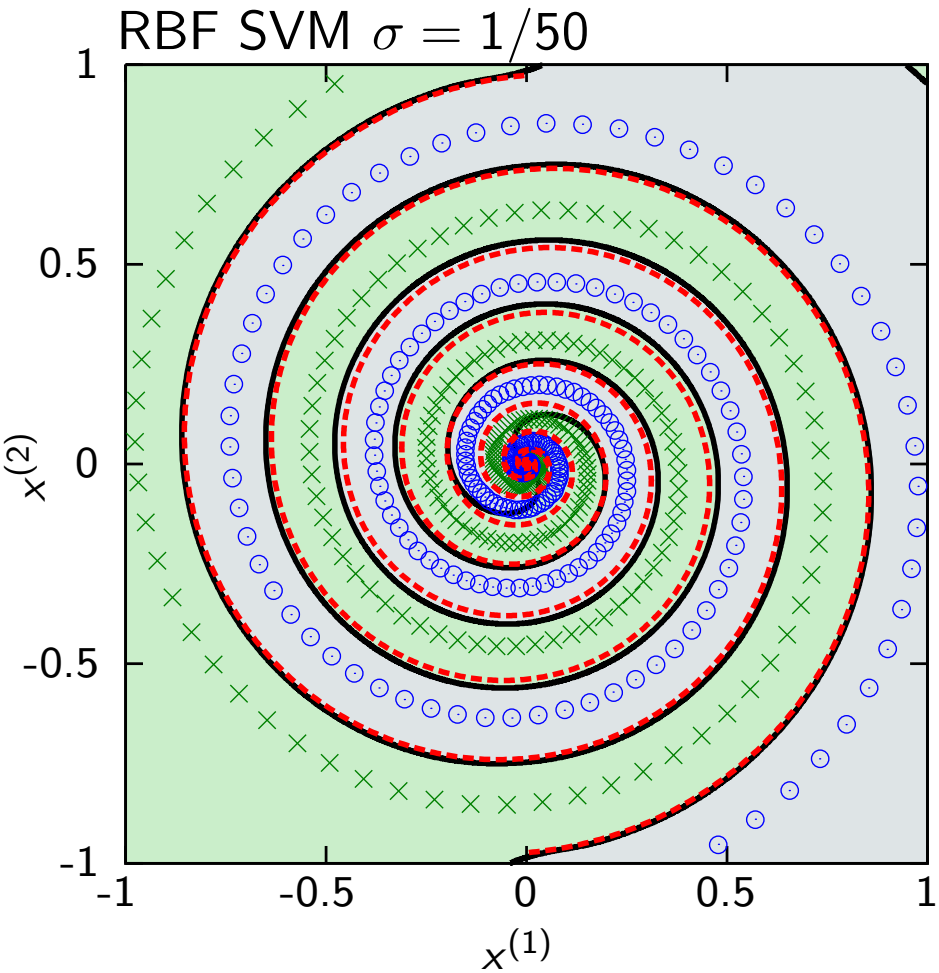
$$d = 2.5,$$

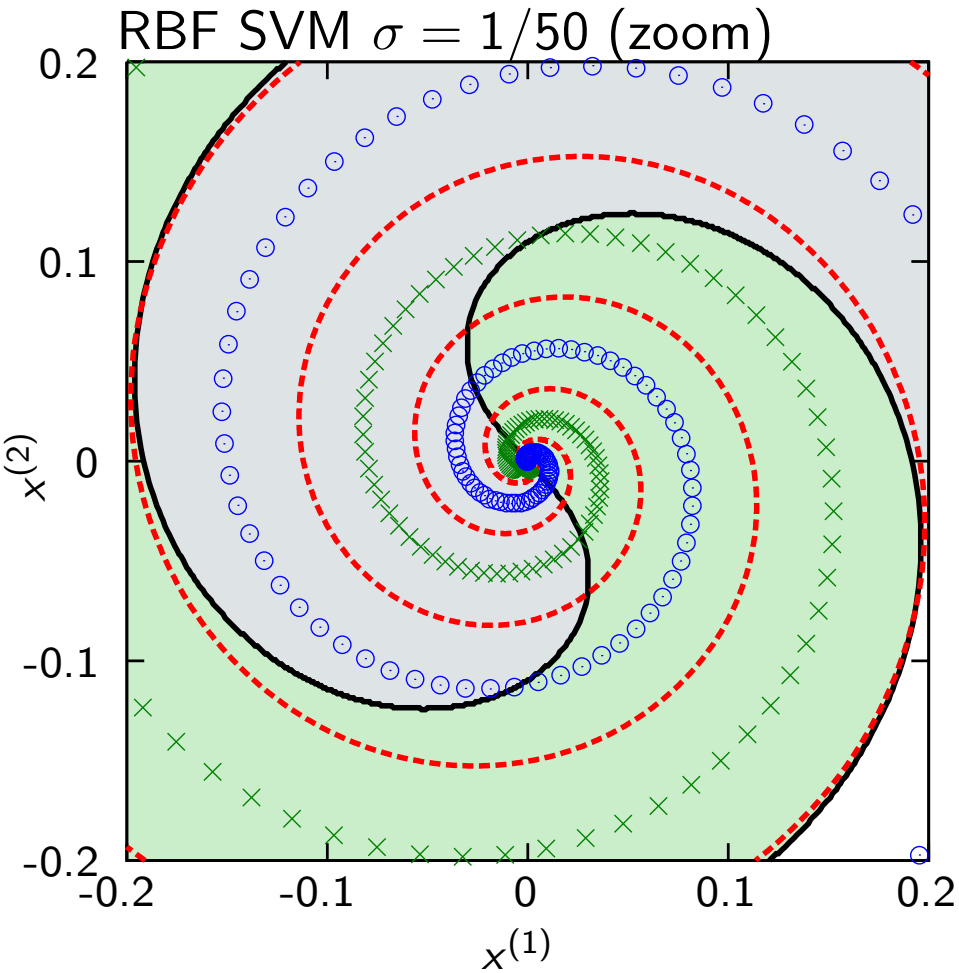
$$c = y_i/500$$

$$t \in [0, 10\pi]$$

sampling rate $t \cdot \pi/30$

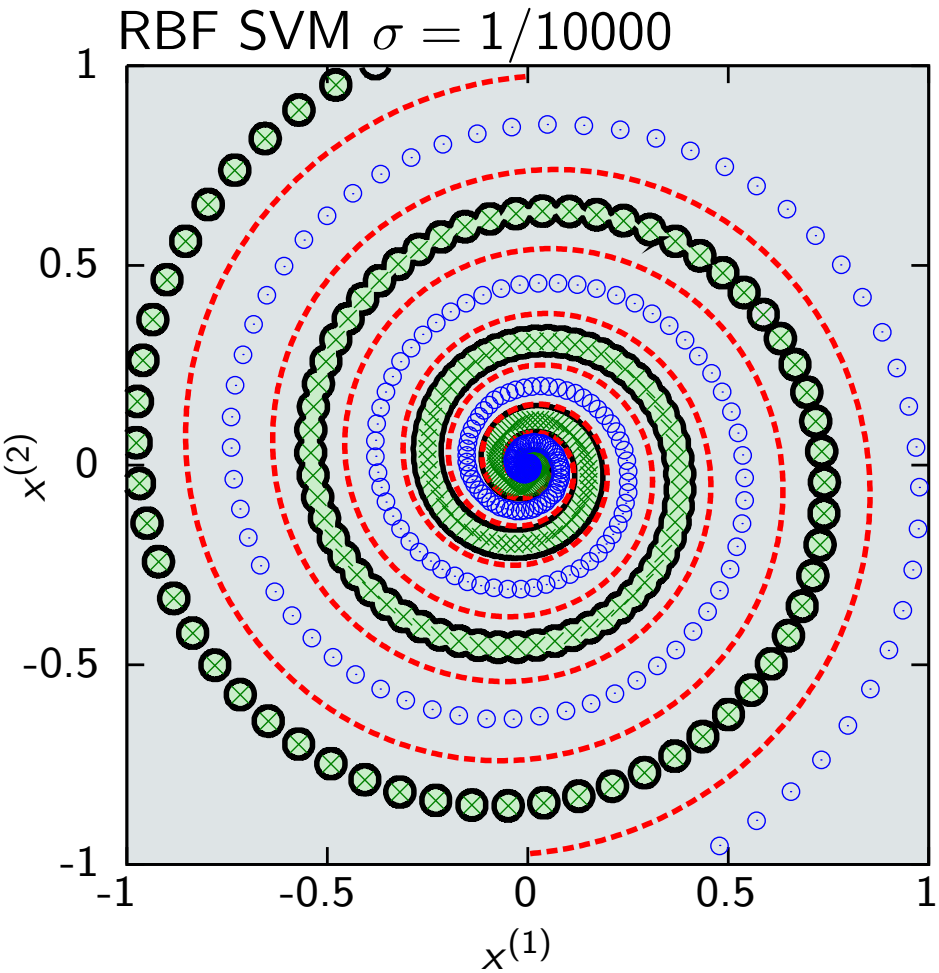
SVM with RBF kernel on the two-spirals dataset

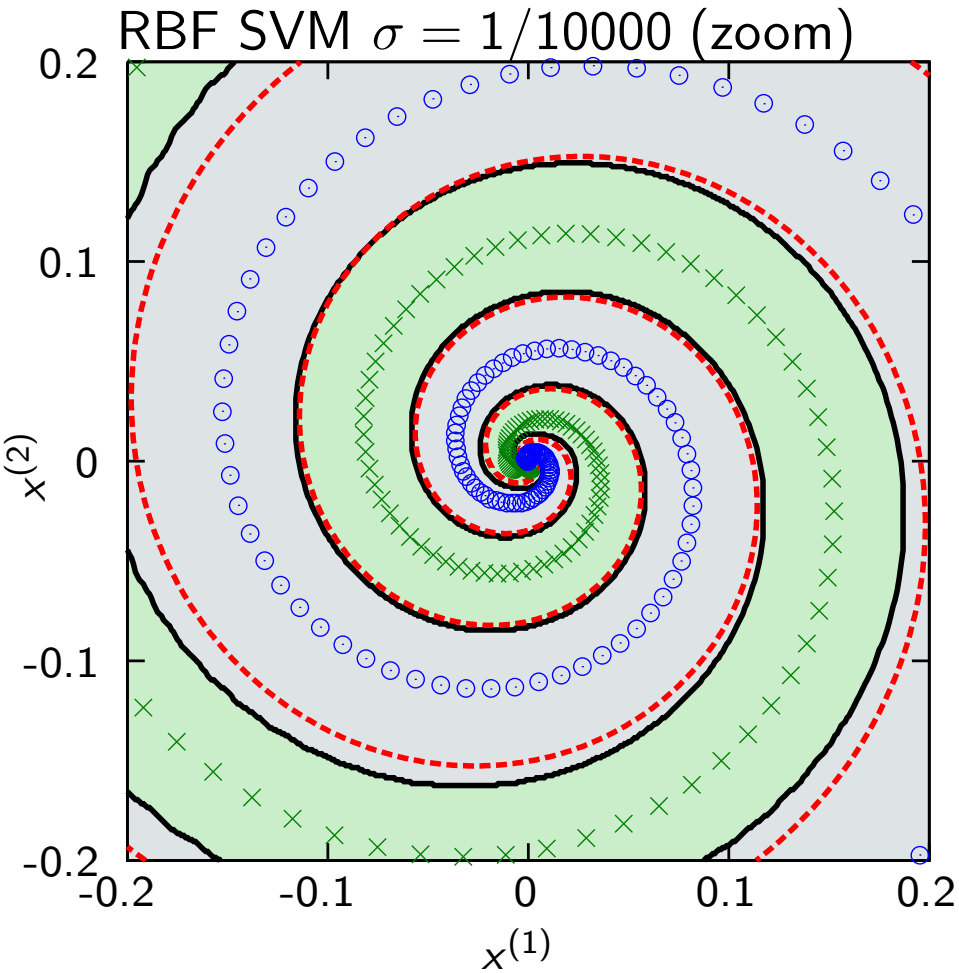




Under-fitting...

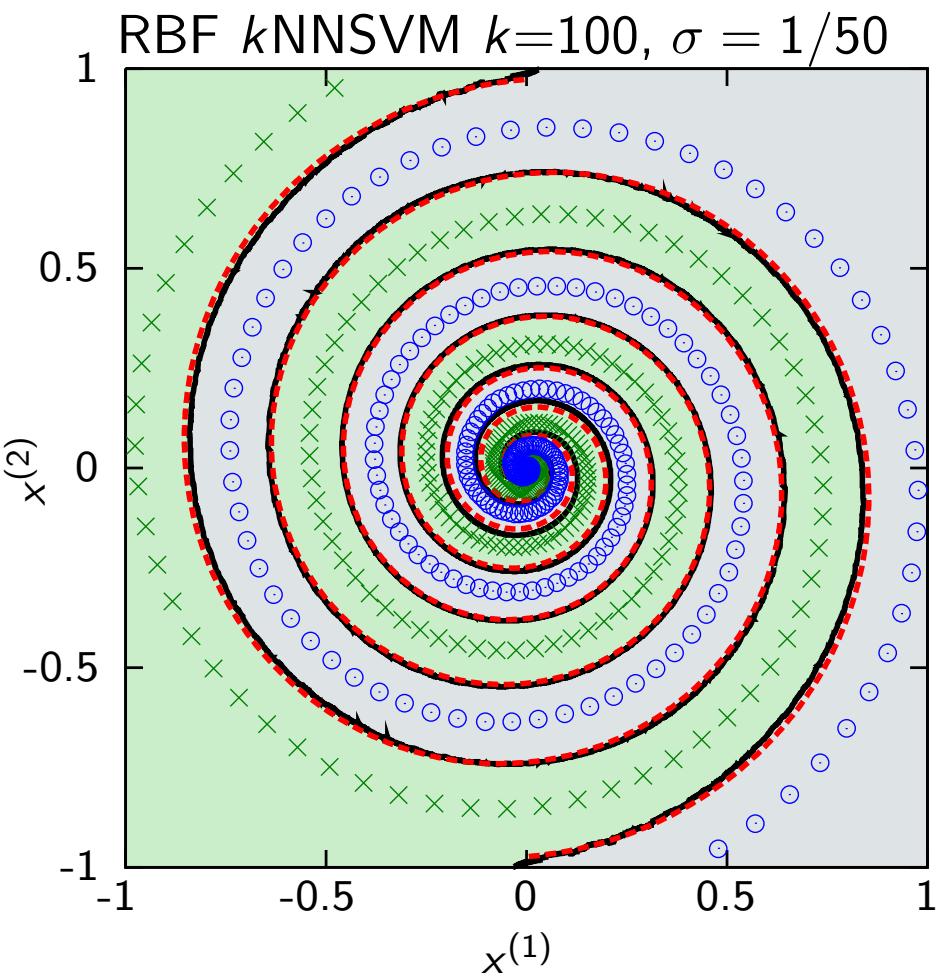
SVM with RBF kernel on the two-spirals dataset

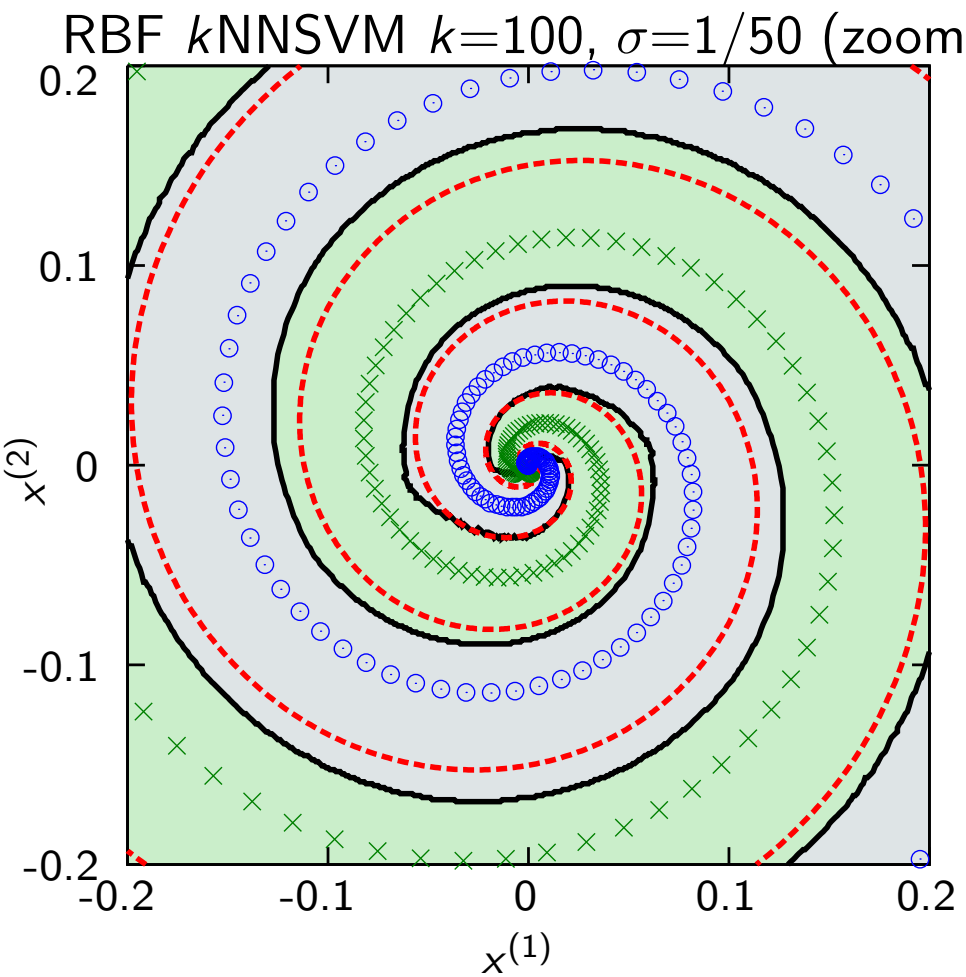




Over-fitting...

kNNSVM with RBF kernel on the two-spirals dataset



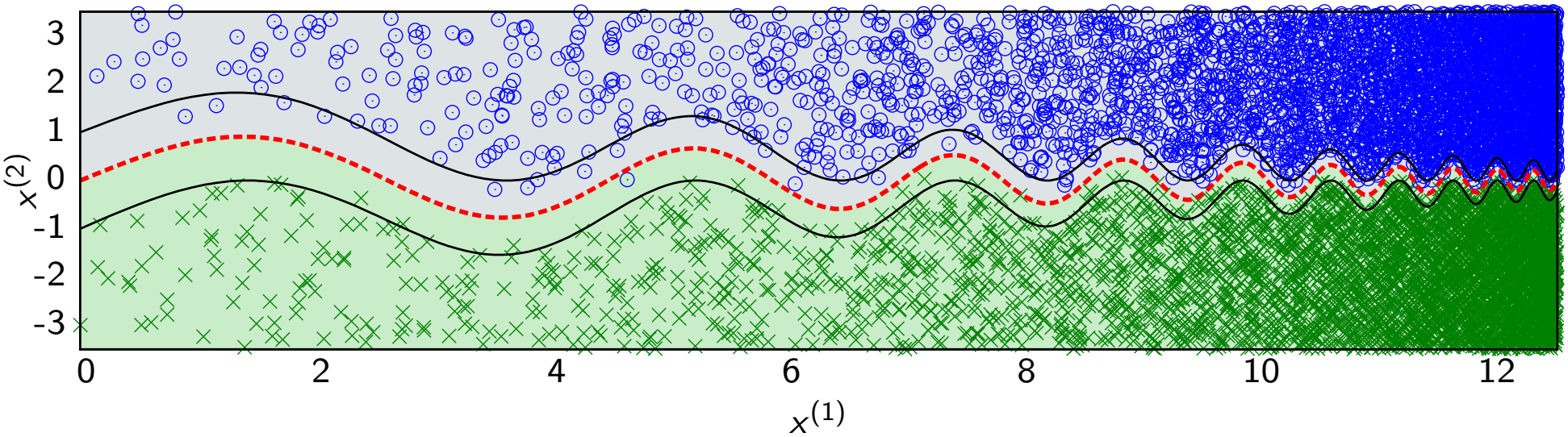


No Over- or Under-fitting!

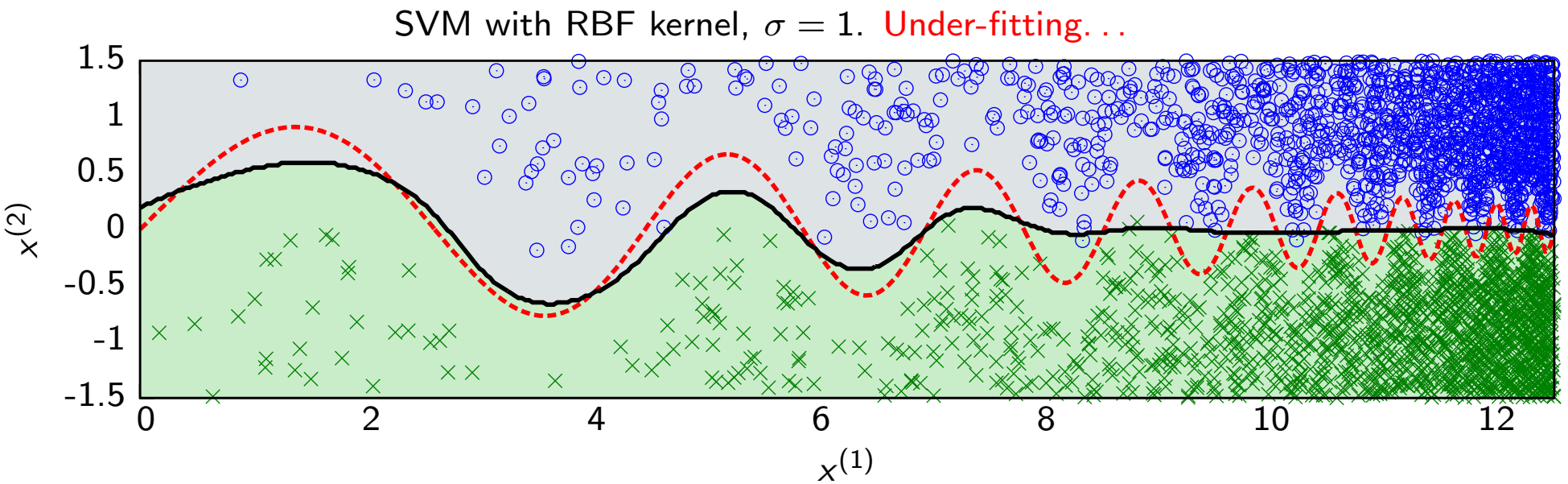
The DECSIN dataset

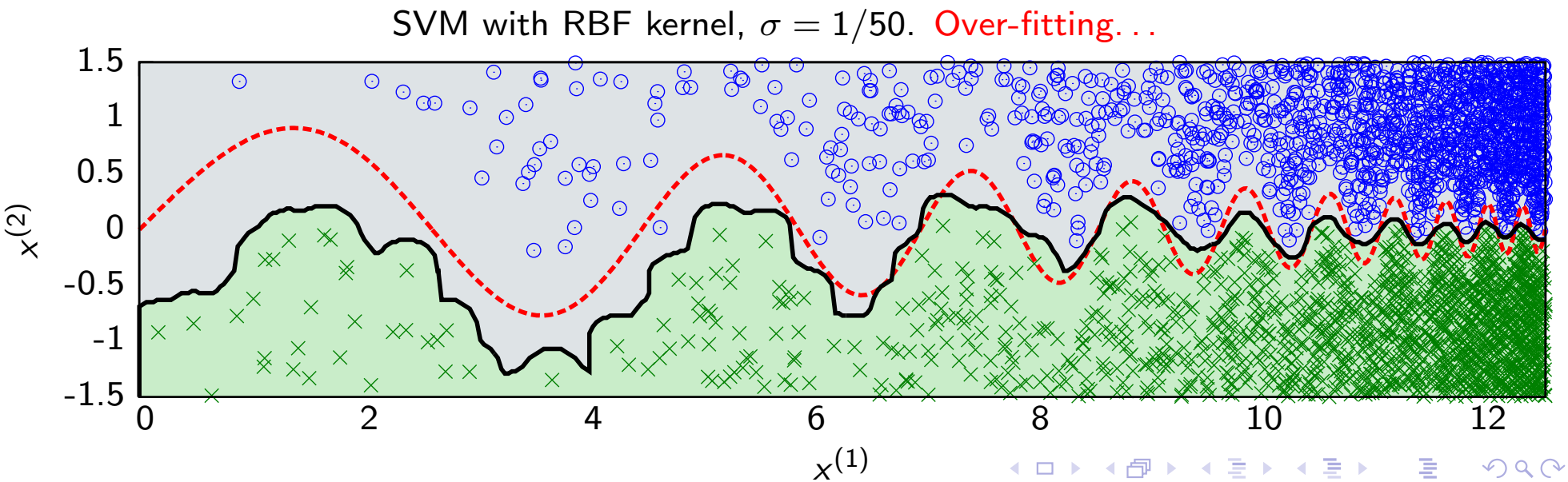
Parametric definition:

$$\begin{cases} u(t) = \dfrac{t}{1 + c \cdot t} \\ v(t) = \dfrac{sin(t)}{1 + c \cdot t} \end{cases} \qquad c = \frac{1}{5 \cdot \pi}, \quad t \in [0, 20\pi]$$



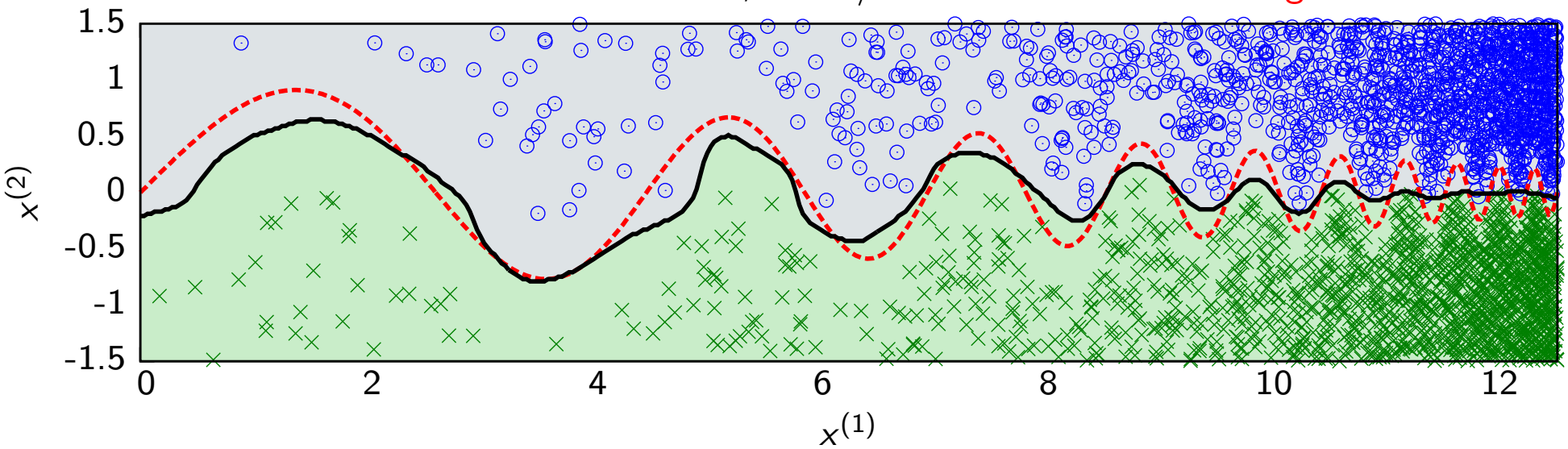
SVM with RBF kernel on the DECSIN dataset



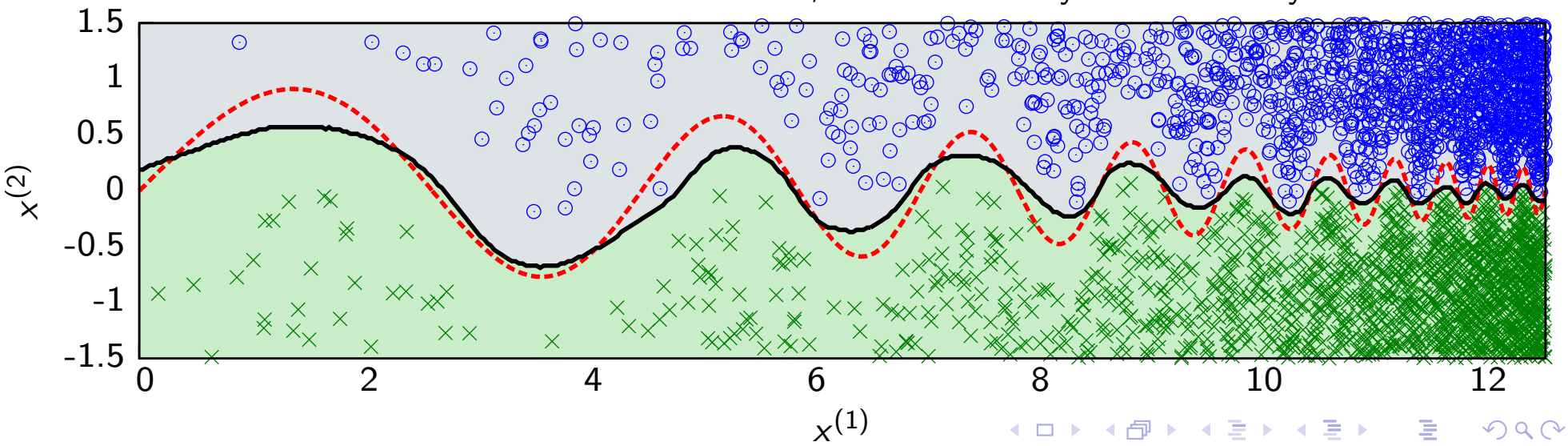


SVM and *k*NNSVM with RBF on the DECSIN dataset

SVM with RBF kernel, $\sigma = 1/10$. Over- and under-fitting...



*k*NNSVM with RBF kernel, σ automatically chosen locally.



Conclusions: Local SVM charactheristics

The Local SVM approach can be seen as

1. a generalization of SVM
2. a way of using SVM with a lazy learning setting
3. a modified k-NN in which the majority rule is substituted with the SVM decision function
4. a local learning algorithm that locally applies the maximal margin principle
5. a strategy to attach complex datasets with simpler classes of decision functions
6. a strategy to handle datasets with very uneven distributions
7. a development of SVM to enhance the classification accuracies

Conclusions: the empirical analysis

1. Local SVM is better than SVM for non local kernels (linear and polynomials) with statistical significance
2. Local SVM can achieve better classification results w.r.t. SVM if we can select the kernel
3. Although is not clear if Local SVM is more accurate than SVM with the RBF kernel for general real datasets, there are cases in which Local SVM with RBF kernel performs better than SVM with RBF kernel

Locality enhances the classification capability of SVM

This motivates us:

- to find approximations of the approach for tackling large datasets efficiently
  - is locality more important for large datasets?
- to apply the approach for other tasks

Related and outgoing work

FaLK-SVM: Fast Local Kernel Machines for Large Datasets

- adoption of Cover Trees for fast neighborhood retrieval
- pre-computations of the local models during training phase
- minimization of the # of local models covering the training set

FaLK-SVM scalable to very large datasets $\mathcal{O}(n \log n)$, faster and more accurate than SVM for non very high-dimensional data. Preliminary version in [Segata,Blanzieri - MLDM09]

Local SVM for noise reduction

- application of probabilistic kNNSVM in a LOO setting
- removal of training points with prediction-label discordance

FkNNSVM-nr favourable benchmark w.r.t. traditional noise reduction [Segata,Blanzieri,Delany,Cunningham - DISI TR]

FaLKNR a fast and scalable variant of FkNNSVM based on FaLK-SVM [Segata,Blanzieri,Cunningham - ICCBR09]

FaLKM-lib v. 1.0

FaLKM-lib v1.0 is a library for fast local kernel machine implemented in C++. It contains the following modules:

FkNN	a (kernel-space) kNN implementation using Cover Trees
FkNNSVM	the kNNSVM algorithm of this work with computational (non-approximated) improvements
FkNNSVM-nr	a noise reduction algorithm based on kNNSVM
FaLK-SVM	very fast and scalable learning with local kernel machines
FaLKNR	a fast and scalable noise reduction algorithm

The modules share also tools for model selection, efficient local model selection, performance assessment...

FaLKM-lib is freely available for research purposes

You can download the code, datasets, benchmark, additional infos and examples at `http://disi.unitn.it/~segata/FaLKM-lib`
Any comments/suggestions are welcome!

Empirical Assessment of Classification Accuracy of Local SVM

Nicola Segata Enrico Blanzieri

Department of Engineering and Computer Science (DISI)
University of Trento, Italy.
segata@disi.unitn.it

18th Annual Belgian-Dutch Conference on Machine Learning

Tilburg University May 19, 2009

Related works: FaLK-SVM, a Fast Local Kernel Machine

Enhancing computational performances:

- use a supporting data-structure for efficiently handling neighborhood retrievals (like Cover Tree)
- pre-compute local models during training phase
- reduce the number of local models required to cover the entire training set space

FaLK-SVM: Fast Local Kernel Machines for Large Datasets

FaLK-SVM is a scalable approach $\mathcal{O}(n \log n)$ applicable to very large datasets (up to some millions training points) faster and more accurate than SVM for non very high-dimensional data. Preliminary version [Segata,Blanzieri - MLDM09]

Related works: Local SVM for noise reduction

- some local and instance-based classification approaches are not noise-tolerant (es. nearest neighbor)
- noise reduction strategies detect and remove noisy samples that would cause classification errors

Local SVM for noise reduction

- applies a probabilistic variant of kNNSVM for each training point, and remove it if the probability of misclassification is higher than a threshold [Segata,Blanzieri,Delany,Cunningham - DISI TR]
- improves the generalization ability of NN (better than state-of-the-art noise reduction techniques)
- can be made scalable for large datasets [Segata,Blanzieri,Cunningham - ICCBR09]