Support Vector Machines

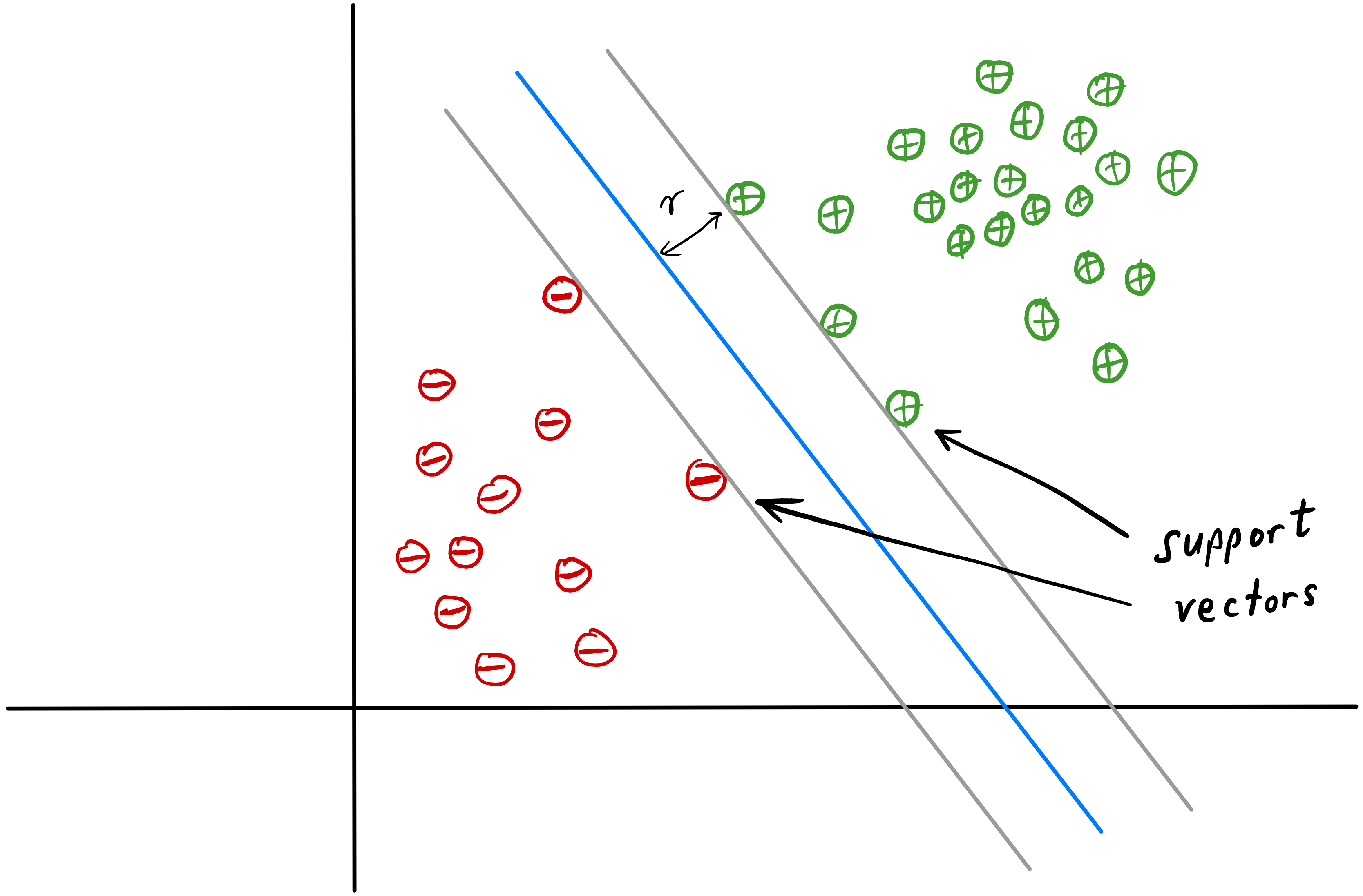
Nishant Mehta

Lecture 9

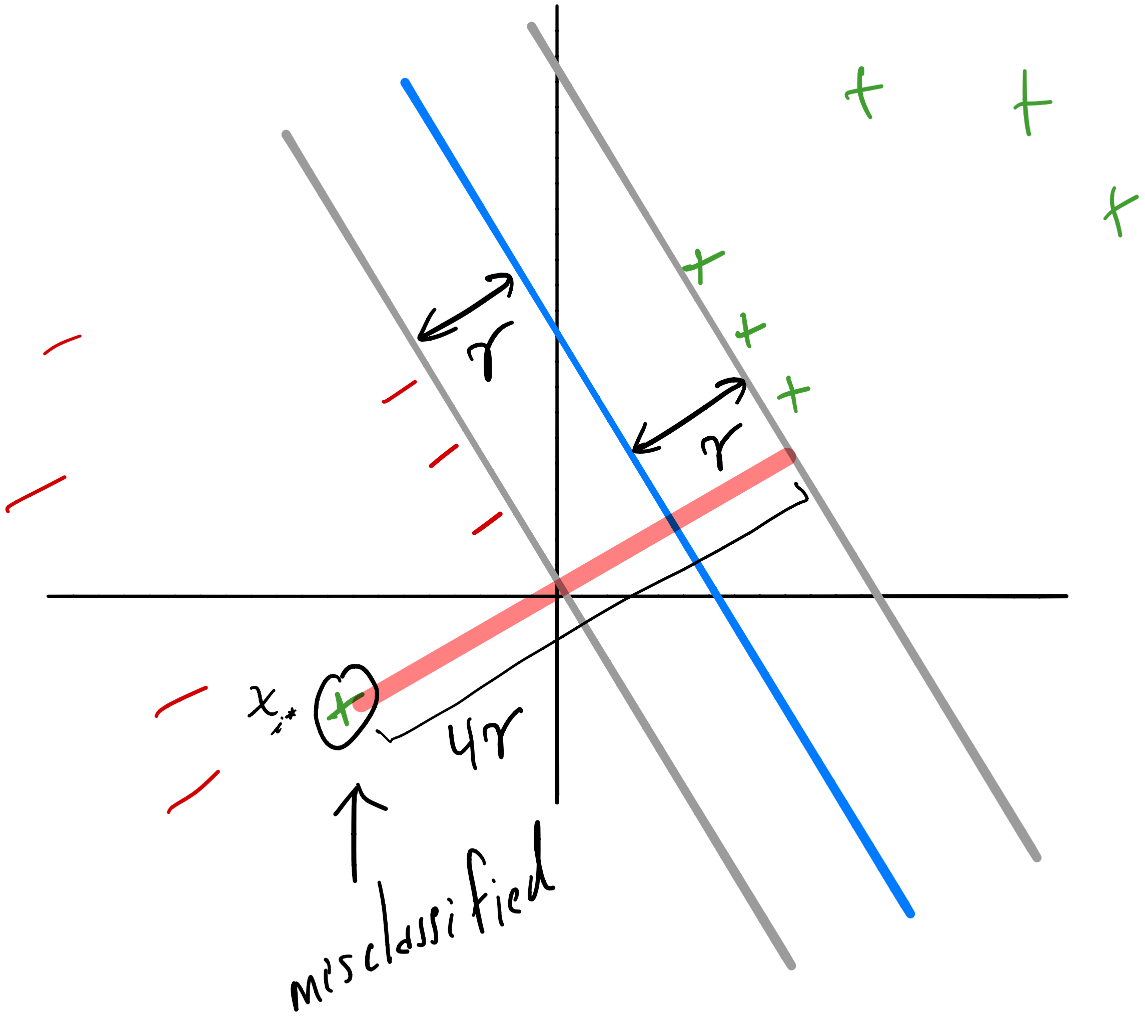
Hard-margin SVM

Hard margin SVM problem

$$\begin{aligned} \underset{w,b}{\text{minimize}} \quad & \|w\|^2 \\ \text{subject to} \quad & y_i\big(\langle w, x_i\rangle + b\big) \geq 1, \;\; i = 1, \ldots, n. \end{aligned}$$



Soft-margin SVM



What if data isn’t linearly separable?

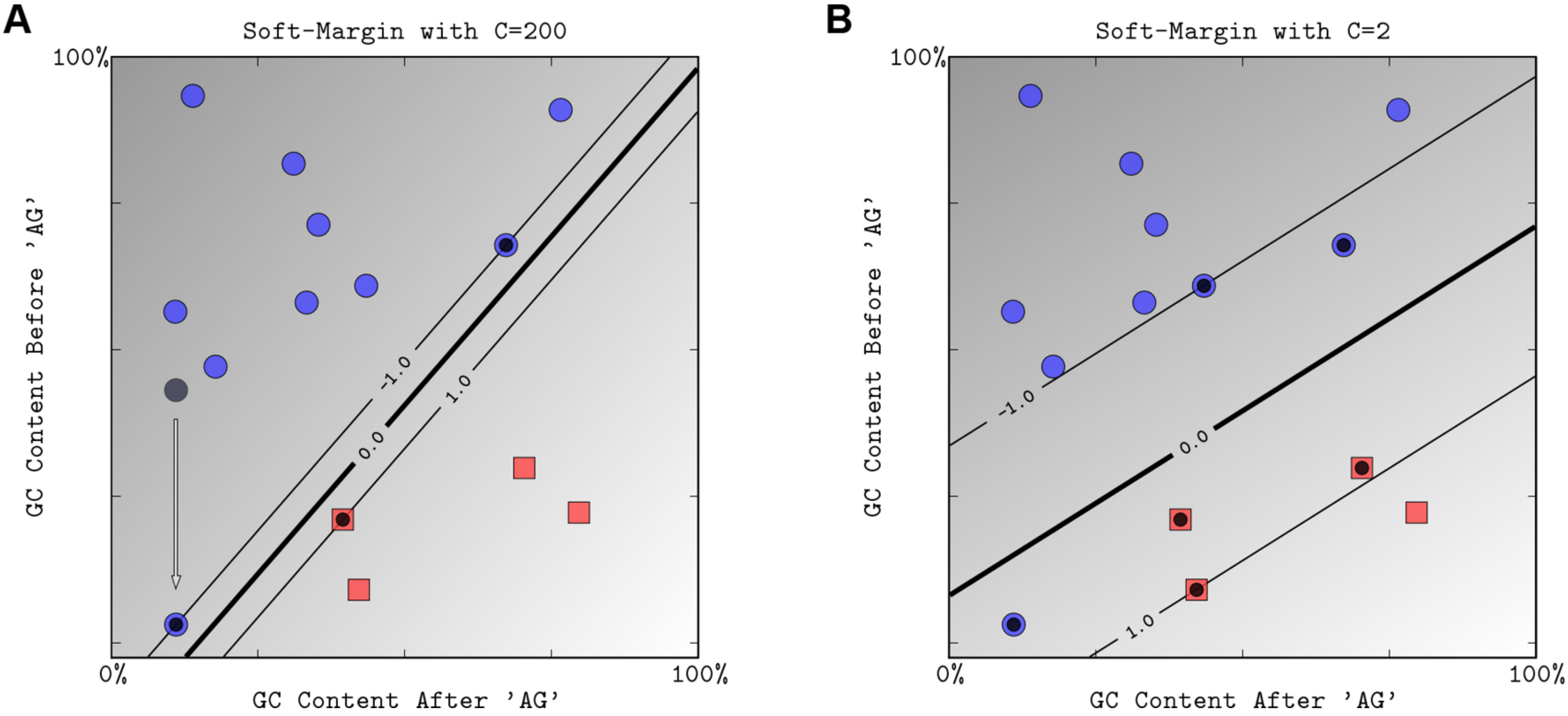
Or, most of the data is separable with large margin, and some only with very low margin?

Soft-margin SVM problem

$$\underset{\substack{w\in\mathbb{R}^n, b\in\mathbb{R} \\ \xi\in\mathbb{R}^n_+}}{\text{minimize}} \quad \|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \quad y_i\big(\langle w, x_i\rangle + b\big) \geq 1 - \xi_i,\ \ i = 1, \ldots, n$$

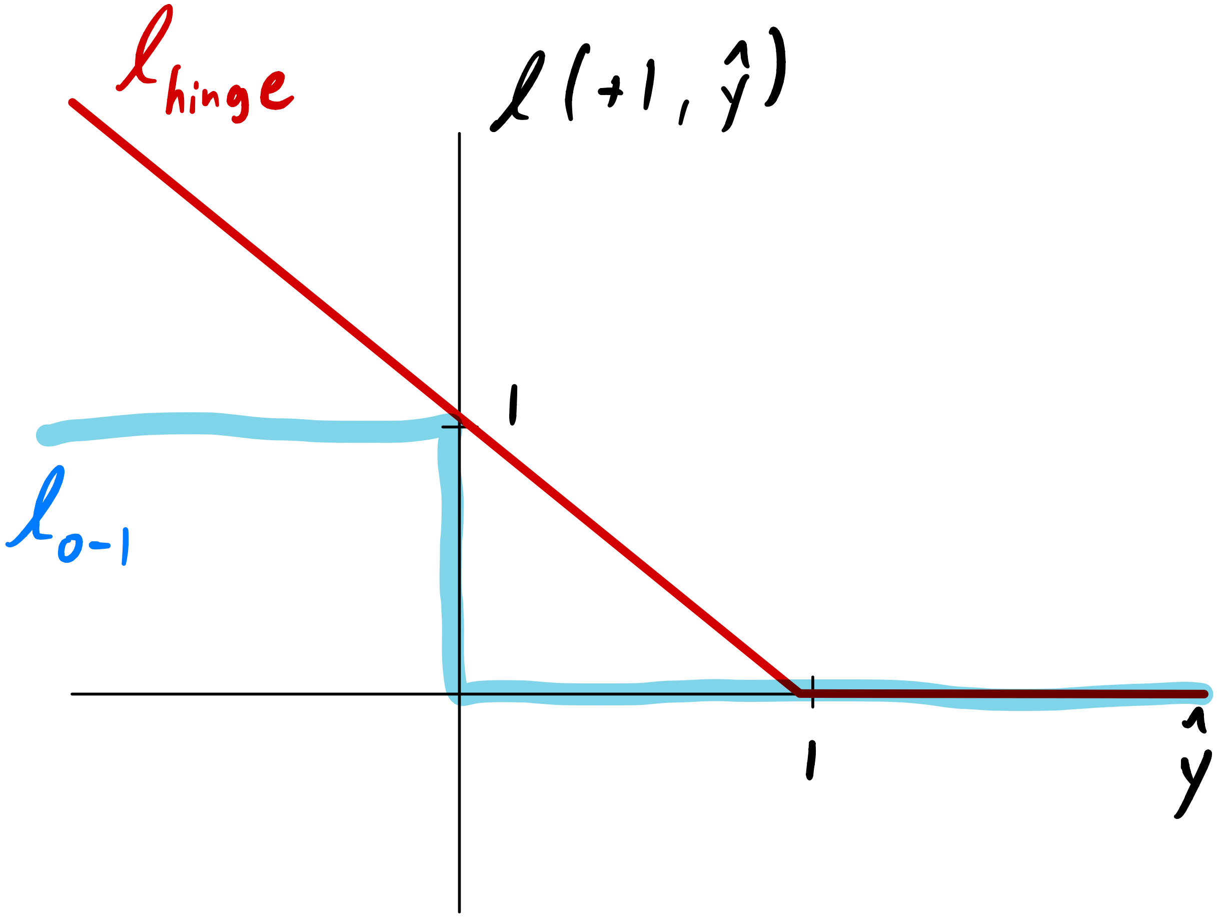
Varying C (linear kernel)



From "Support Vector Machines and Kernels for Computational Biology" (Ben-Hur et al., 2008)

Soft-margin SVM - Hinge Loss

$$\underset{w\in\mathbb{R}^n, b\in\mathbb{R}}{\text{minimize}} \quad \|w\|^2 + C\sum_{i=1}^{n} \max\Big\{0, 1 - y_i\big(\langle w, x_i\rangle + b\big)\Big\}$$

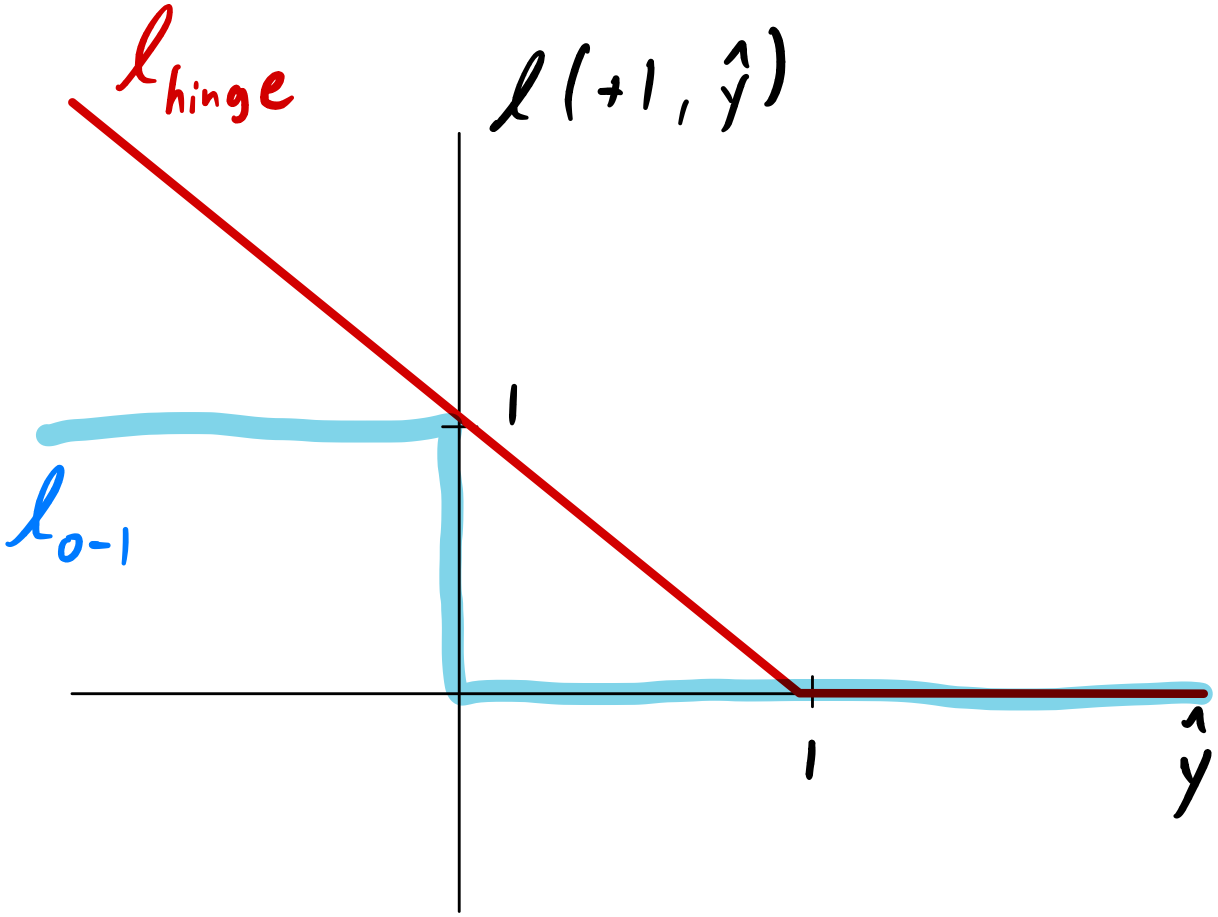


hinge loss

$$\ell_{\text{hinge}}(y, \hat{y}) = \max\{0, 1 - y\hat{y}\}$$

Soft-margin SVM - Hinge Loss

$$\underset{w\in\mathbb{R}^n, b\in\mathbb{R}}{\text{minimize}} \quad \|w\|^2 + C\sum_{i=1}^{n} \max\Big\{0, 1 - y_i\big(\langle w, x_i\rangle + b\big)\Big\}$$



hinge loss

$$\ell_{\text{hinge}}(y, \hat{y}) = \max\{0, 1 - y\hat{y}\}$$

$$\underset{w\in\mathbb{R}^n, b\in\mathbb{R}}{\text{minimize}} \quad \|w\|^2 + C\sum_{i=1}^{n} \ell_{\text{hinge}}\big(y_i, f_{w,b}(x_i)\big)$$

SVM - Regularization viewpoint

SVM can be viewed as minimizing regularized training error under hinge loss

$$\underset{w \in \mathbb{R}^n, b \in \mathbb{R}}{\text{minimize}} \quad \|w\|^2 + C \sum_{i=1}^{n} \ell_{\text{hinge}}\big(y_i, f_{w,b}(x_i)\big)$$

$$\Updownarrow$$

Equivalent

$$\lambda = \frac{1}{C}$$

$$\underset{w \in \mathbb{R}^n, b \in \mathbb{R}}{\text{minimize}} \quad \sum_{i=1}^{n} \ell_{\text{hinge}}\big(y_i, f_{w,b}(x_i)\big) + \lambda \|w\|^2$$

SVM dual problem

$$\begin{aligned} \underset{\alpha \in \mathbb{R}^n}{\text{maximize}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \\ \text{subject to} \quad & \sum_{i=1}^{n} y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C, \ i = 1, \ldots, n \end{aligned}$$

How to get w and b from this?

$$w = \sum_{i=1}^{n} y_i \alpha_i x_i$$

$$b = y_i - \sum_{j=1}^{n} y_j \alpha_j \langle x_i, x_j \rangle \quad \text{for any } i \text{ satisfying } 0 < \alpha_i < C$$

How to predict?

$$f_{w,b}(x_{\text{test}}) = \langle w, x_{\text{test}} \rangle + b = \sum_{i=1}^{n} y_i \alpha_i \langle x_i, x_{\text{test}} \rangle + b$$

SVM dual problem - Inner products only

$$\begin{aligned}
\underset{\alpha\in\mathbb{R}^n}{\text{maximize}} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle \\
\text{subject to} \quad & \sum_{i=1}^n y_i \alpha_i = 0 \\
& 0 \leq \alpha_i \leq C,\ \ i = 1, \ldots, n
\end{aligned}$$

How to predict? $f_{w,b}(x_{\text{test}}) = \langle w, x_{\text{test}}\rangle + b = \sum_{i=1}^n y_i \alpha_i \langle x_i, x_{\text{test}}\rangle + b$

Dual SVM only needs inner products between input examples!

How can we achieve nonlinear classifiers?

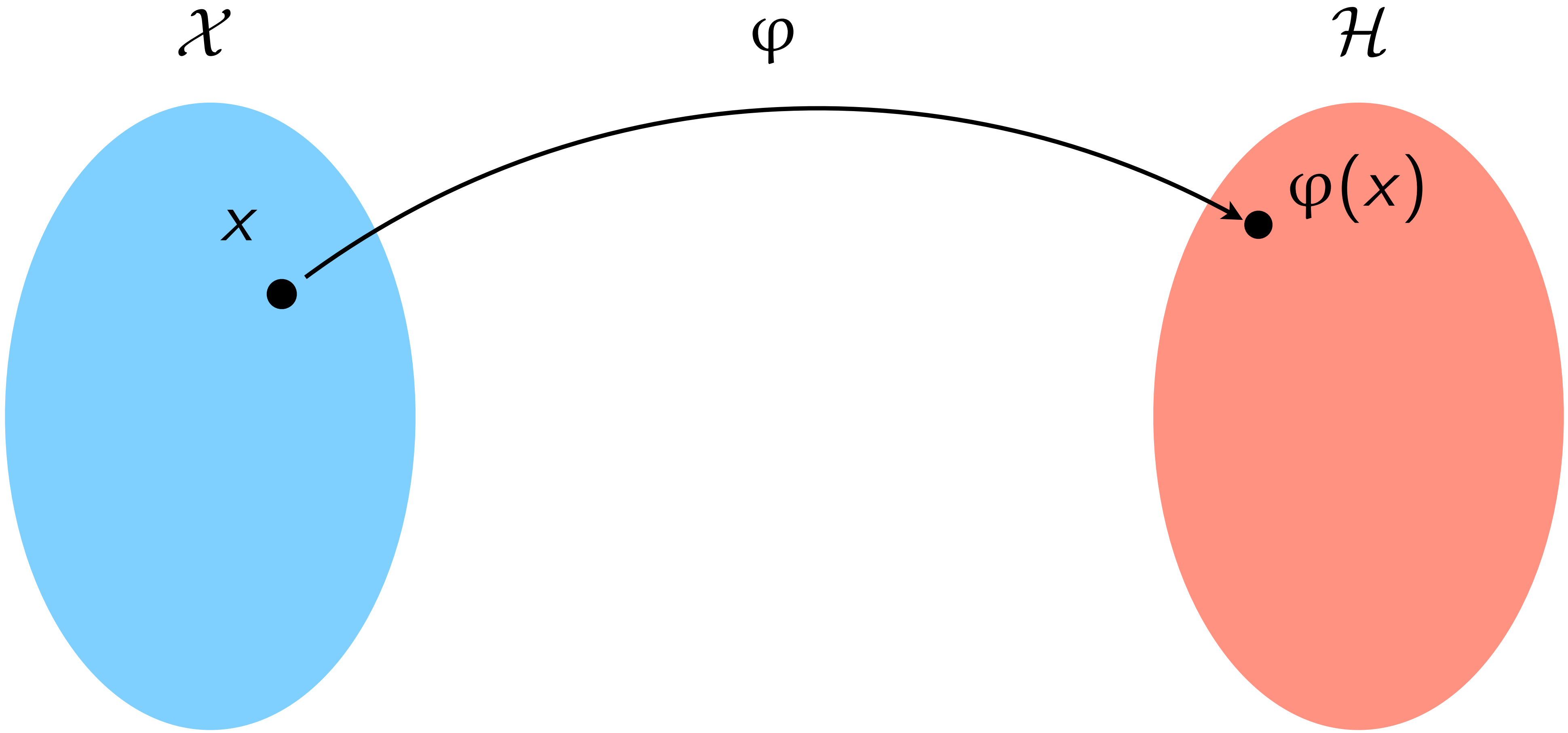
Idea: feature map

Classification in original space

Classification in feature space

Idea: feature map

Use a feature map: $\varphi(x) : \mathcal{X} \rightarrow \mathcal{H}$



Kernel trick

Question: Can we compute inner product between input examples x and z in feature space without explicitly computing $\varphi(x)$ and $\varphi(z)$?

In many cases, yes! We use a *kernel function*:

$$k(x, z) = \langle \varphi(x), \varphi(y) \rangle$$

Equal to inner product… but we won’t compute it this way!

Example 1: Warm-up exercise

Example 2: Polynomial kernel, one dimension

The *polynomial kernel* (one dimension):

$$k(x, z) = (xz + a)^r$$

What is the feature space?

Example 3: Polynomial kernel, general dimension

The *polynomial kernel* (general dimension):

$$k(x, z) = \left(\langle x, z \rangle + a\right)^r$$

$\varphi(x)$ has one feature for each monomial up to degree *r*

How many features are there in the feature space?

Example 3: Polynomial kernel, general dimension

The *polynomial kernel*:

$$k(x, z) = (\langle x, z \rangle + a)^r$$

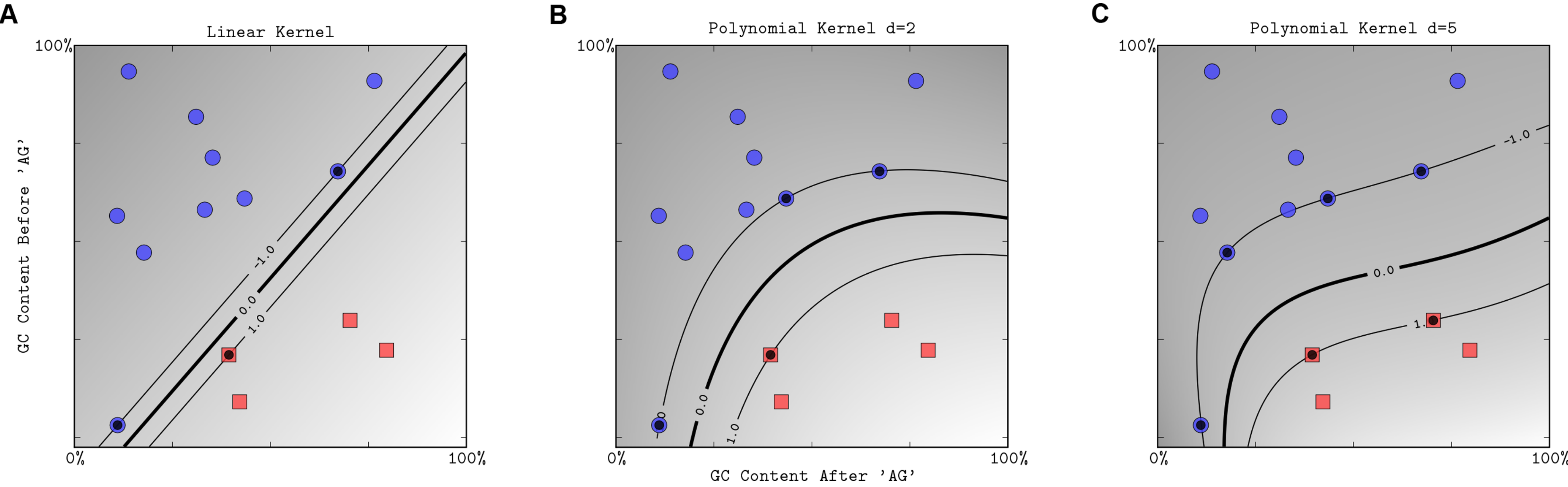
$\varphi(x)$ has one feature for each monomial up to degree r

How many features are there in the feature space?

$$\binom{r+d}{d}$$

But the kernel can be computed in only $O(d)$

Polynomial kernels of increasing degree



Gaussian kernel

The *Gaussian kernel* is based on the distance between two examples

$$k(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$$

bandwidth parameter

The Gaussian kernel is a type of similarity measure, taking values between 0 and 1

What is the corresponding feature map $\varphi(x)$?

Gaussian kernel

The *Gaussian kernel* is based on the distance between two examples

$$k(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$$

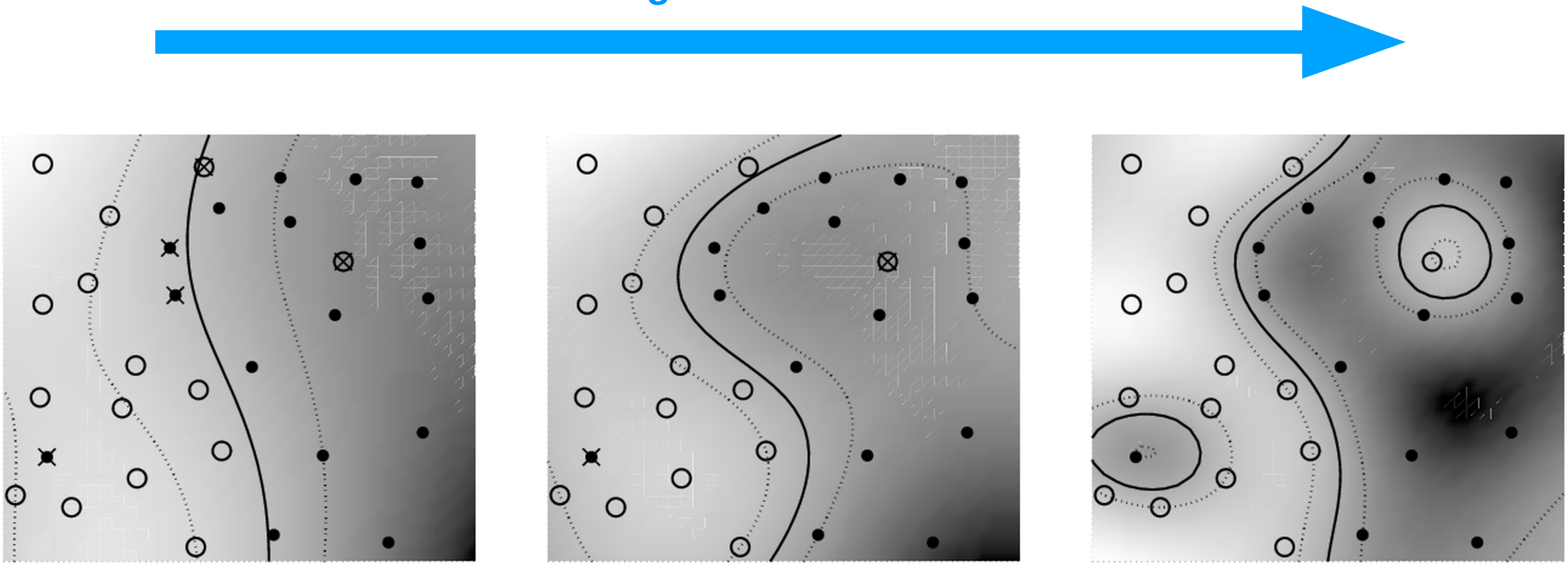
bandwidth parameter

The Gaussian kernel is a type of similarity measure,
taking values between 0 and 1

What is the corresponding feature map $\varphi(x)$?

It’s infinite dimensional!

Varying Gaussian kernel bandwidth (C kept constant)



Decreasing kernel bandwidth

From the book “Learning with Kernels” (Schölkopf and Smola, 2001)