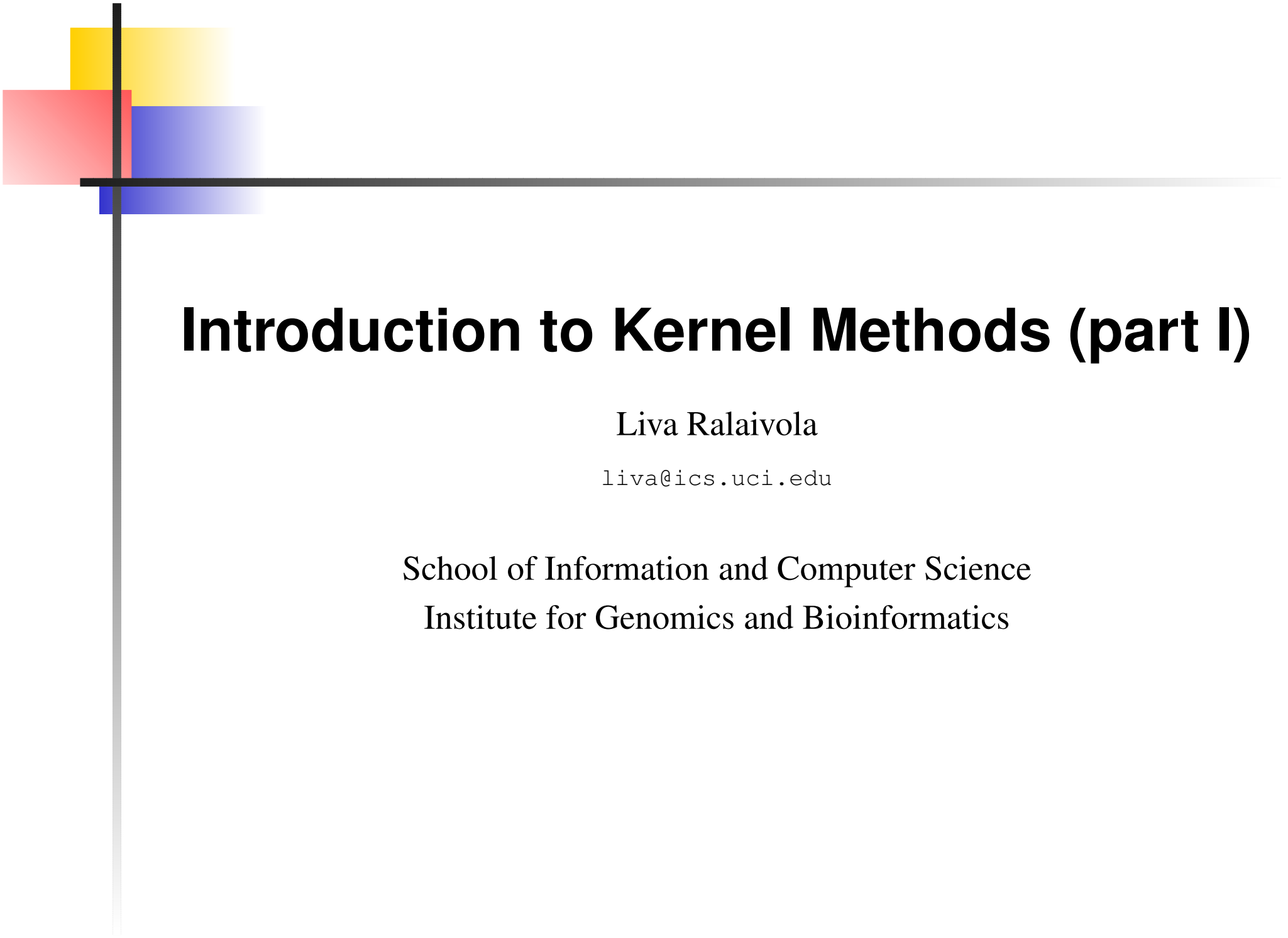
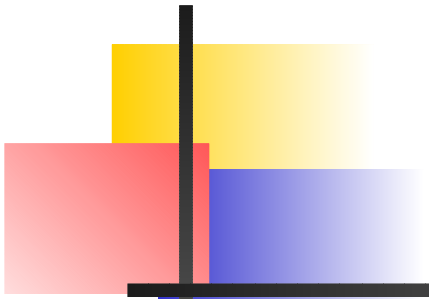


Introduction to Kernel Methods (part I)

Liva Ralaivola

liva@ics.uci.edu

School of Information and Computer Science
Institute for Genomics and Bioinformatics



Outline

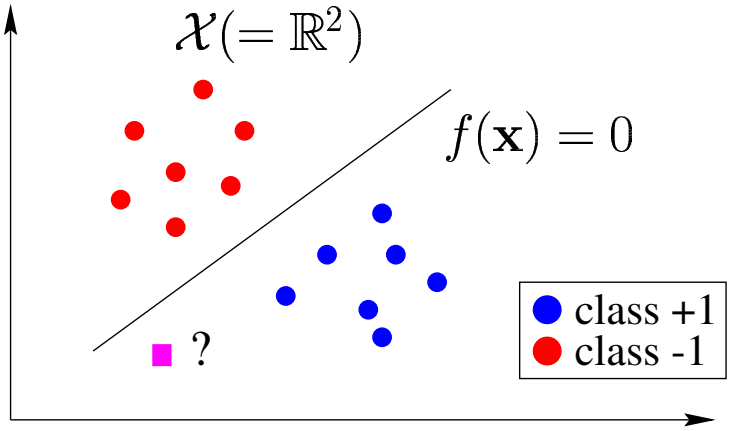
- Learning to classify
- Vectors and inner products
- A simple linear classifier
- *Kernel trick* and linear classifiers in *feature spaces*
- Common kernels

Classifying things (1/2)

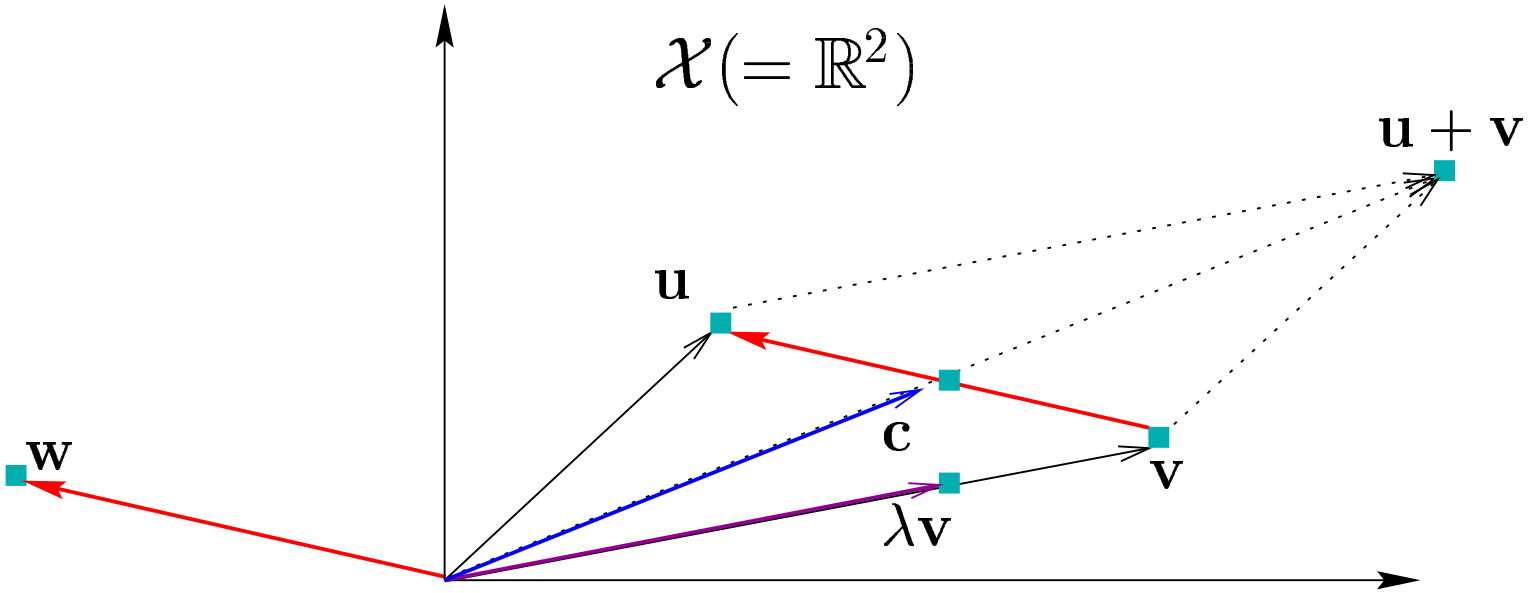
- Classification is a real-world task
  - does some patient have a serious disease?
  - what kind of secondary structure does a sequence of amino acids correspond to?
  - is some molecule toxic/not toxic for some organism?
- Automating this task
  - dealing with large number of items to classify
  - speed
  - cost

Classifying things (2/2)

- Important notions in *learning to classify*
  - limited number of *training* data (patients, sequences, molecules, etc.)
  - learning algorithm (how to build the classifier?)
  - generalization: the classifier should correctly classify *test* data
- Quick formalization
  - $\mathcal{X}$ (e.g. $\mathbb{R}^d, d > 0$) is the space of data, called *input space*
  - $\mathcal{Y}$ (e.g. toxic/not toxic, or $\{-1, +1\}$) is the target space
  - $f : \mathcal{X} \to \mathcal{Y}$ is the classifier



Vectors and inner product (1/3)

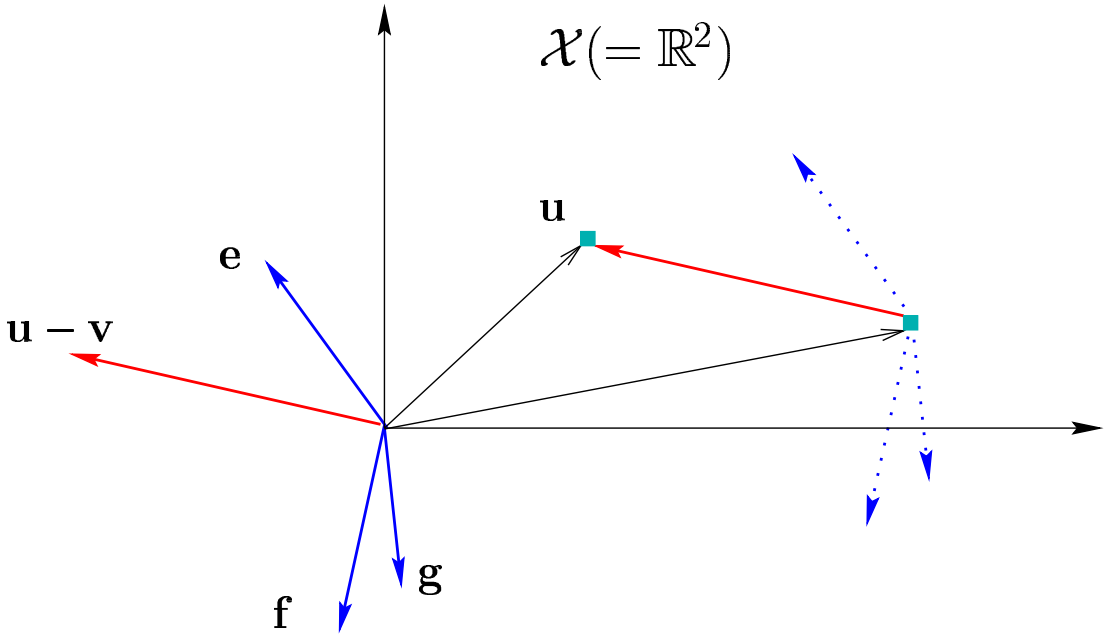


- $\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{c}$ are vectors
- $\mathbf{w} = \mathbf{u} - \mathbf{v}$ (red arrows)
- $\mathbf{c} = \frac{1}{2}(\mathbf{u} + \mathbf{v})$
- Here: $0 < \lambda < 1$

Vectors and inner product (2/3)

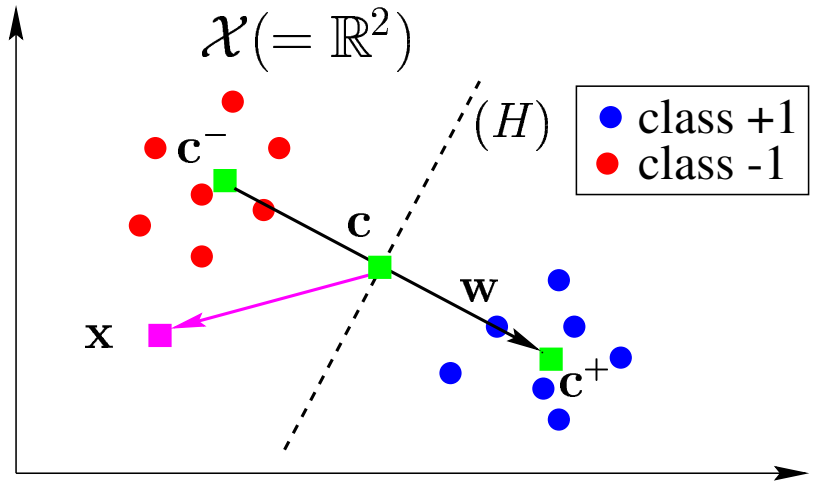
- Inner product $\langle \cdot, \cdot \rangle : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$:
  - symmetric: $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$
  - bilinear: $\langle \lambda \mathbf{u}_1 + \gamma \mathbf{u}_2 \rangle = \lambda \langle \mathbf{u}_1, \mathbf{v} \rangle + \gamma \langle \mathbf{u}_2, \mathbf{v} \rangle$
  - positive: $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$
  - definite: $\langle \mathbf{u}, \mathbf{u} \rangle = 0 \Rightarrow \mathbf{u} = 0$
- An inner product
  - provides $\mathcal{X}$ with a structure
  - can be viewed as a 'similarity'
  - defines a norm $\| \cdot \|$ on $\mathcal{X}$: $\|\mathbf{u}\| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$
- Example in $\mathbb{R}^2$
  - $\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$: $\langle \mathbf{u}, \mathbf{v} \rangle = u_1 v_1 + u_2 v_2$

Vectors and inner product (3/3)



- $\langle \mathbf{u}-\mathbf{v}, \mathbf{e} \rangle > 0$: $\mathbf{u}-\mathbf{v}$ and $\mathbf{e}$ point to the 'same direction'
- $\langle \mathbf{u}-\mathbf{v}, \mathbf{f} \rangle = 0$: $\mathbf{u}-\mathbf{v}$ and $\mathbf{f}$ are orthogonal
- $\langle \mathbf{u}-\mathbf{v}, \mathbf{g} \rangle < 0$: $\mathbf{u}-\mathbf{v}$ and $\mathbf{g}$ point to 'opposite directions'

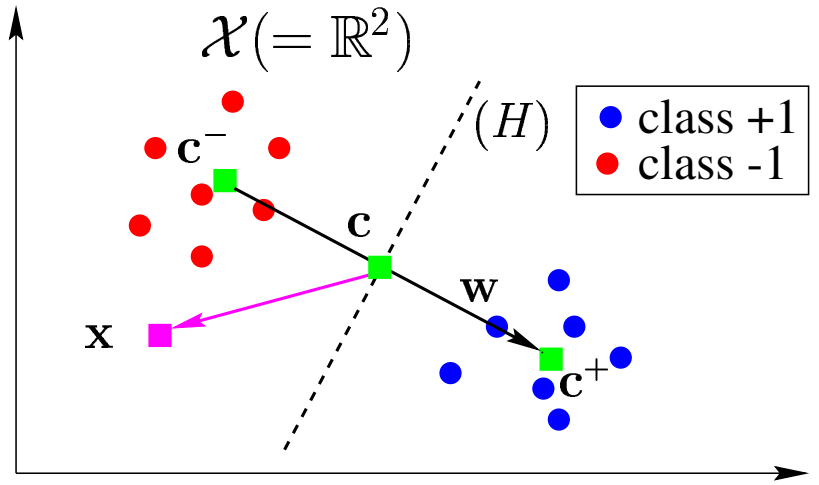
A simple linear classifier



- $\mathbf{c}^+ = \frac{1}{m^+} \sum_{\{i:y_i=+1\}} \mathbf{x}_i$
- $\mathbf{c}^- = \frac{1}{m^-} \sum_{\{i:y_i=-1\}} \mathbf{x}_i$
- $\mathbf{c} = \frac{1}{2}(c^+ + c^-)$
- $\mathbf{w} = c^+ - c^-$

- Idea (see [Schölkopf and Smola, 2002] for details): classify points $\mathbf{x}$ according to which of the two class means $\mathbf{c}^+$ or $\mathbf{c}^-$ is closer:
  - for $\mathbf{x} \in \mathcal{X}$, it is sufficient to take the sign of the inner product between $\mathbf{w}$ and $\mathbf{x} - \mathbf{c}$
  - if $h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} - \mathbf{c} \rangle$, we have the classifier $f(\mathbf{x}) = \text{ sign } (h(\mathbf{x}))$
  - the (dotted) hyperplane (H), of normal vector $\mathbf{w}$, is the decision surface

A simple linear classifier

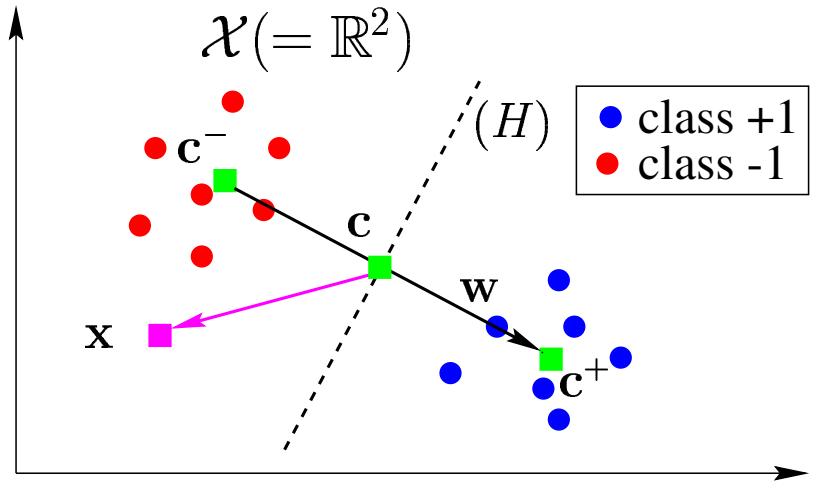


- $\mathbf{c}^+ = \frac{1}{m^+} \sum_{\{i:y_i=+1\}} \mathbf{x}_i$
- $\mathbf{c}^- = \frac{1}{m^-} \sum_{\{i:y_i=-1\}} \mathbf{x}_i$
- $\mathbf{c} = \frac{1}{2}(c^+ + c^-)$
- $\mathbf{w} = c^+ - c^-$

- On evaluating $h(\mathbf{x})$

$$
\begin{aligned}
h(\mathbf{x}) &= \langle \mathbf{w}, \mathbf{x} - \mathbf{c} \rangle = \langle \mathbf{w}, \mathbf{x} \rangle - \langle \mathbf{w}, \mathbf{c} \rangle \\
&= \langle \mathbf{c}^+, \mathbf{x} \rangle - \langle \mathbf{c}^-, \mathbf{x} \rangle - \langle \mathbf{c}^+, \mathbf{c} \rangle + \langle \mathbf{c}^-, \mathbf{c} \rangle \\
&= \sum_{i=1,\ldots,m} \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b, \quad \text{with } b \text{ a real constant}
\end{aligned}
$$

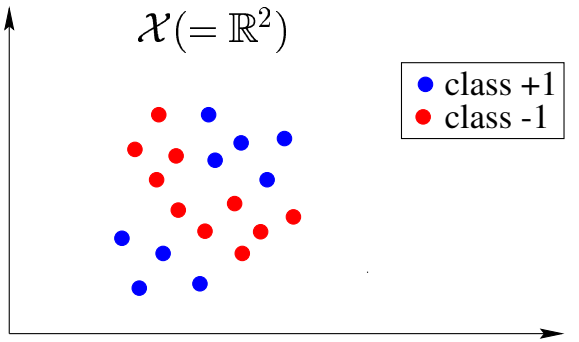
A simple linear classifier



- $\mathbf{c}^+ = \frac{1}{m^+} \sum_{\{i:y_i=+1\}} \mathbf{x}_i$
- $\mathbf{c}^- = \frac{1}{m^-} \sum_{\{i:y_i=-1\}} \mathbf{x}_i$
- $\mathbf{c} = \frac{1}{2}(c^+ + c^-)$
- $\mathbf{w} = c^+ - c^-$

- To summarize: $h(\mathbf{x}) = \sum_{i=1,\ldots,m} \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b$

- Question: what if the dataset is not linearly separable, i.e. (H) fails to separate red and blue disks?



The kernel trick (1/4)

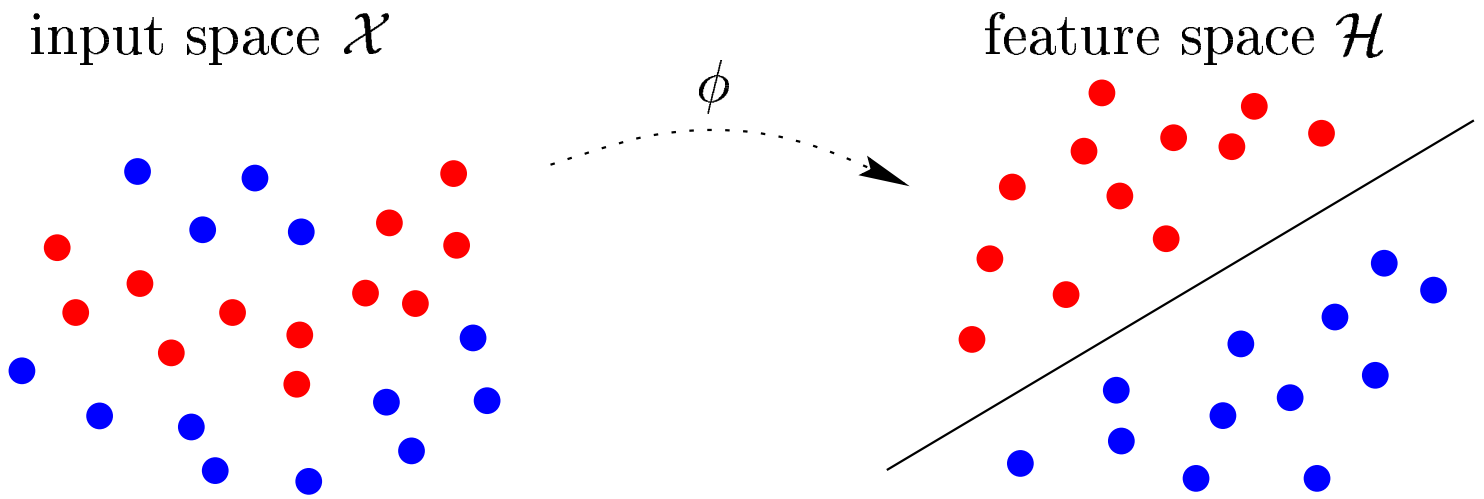
- Context: nonlinearly separable dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$
- Idea to learn a nonlinear classifier
  - choose a (nonlinear) mapping ϕ

$$\begin{array}{rccc} \phi: & \mathcal{X} & \rightarrow & \mathcal{H} \\ & \mathbf{x} & \mapsto & \phi(\mathbf{x}) \end{array}$$

  where $\mathcal{H}$ is an inner product space (inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$), called *feature space*
  - find a linear classifier (i.e. a separating hyperplane) in $\mathcal{H}$ to classify $\{(\phi(\mathbf{x}_1), y_1), \ldots, (\phi(\mathbf{x}_m), y_m)\}$

The kernel trick (2/4)

- Linearly classifying in *feature space*



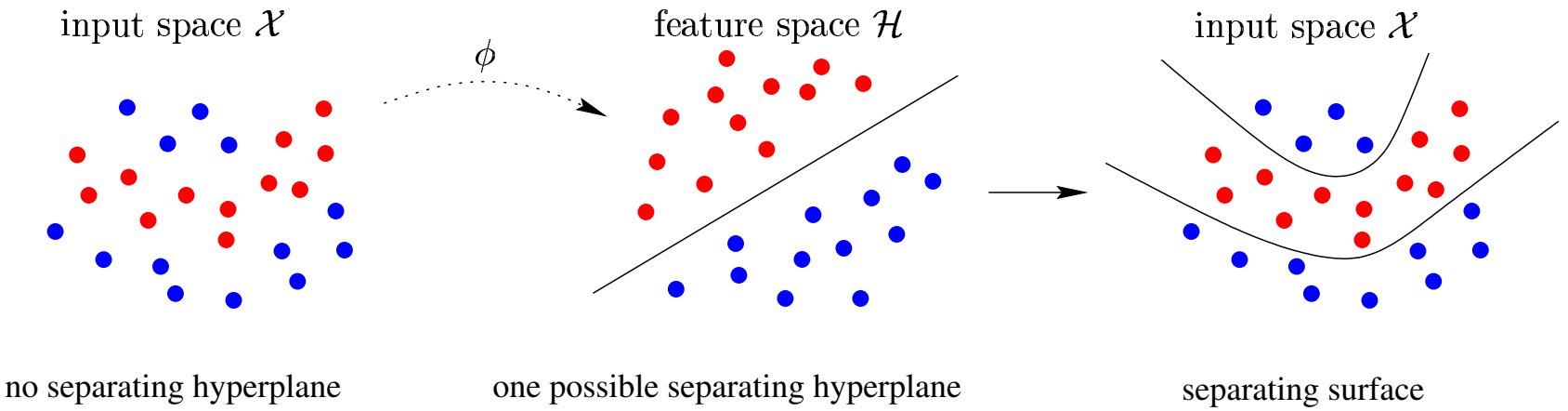
- Taking the previous linear algorithm and implementing it in $\mathcal{H}$:

$$h(\mathbf{x}) = \sum_{i=1,\ldots,m} \alpha_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle_{\mathcal{H}} + b$$

The kernel trick (3/4)

- The kernel trick can be applied if there is a function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ such that: $k(\mathbf{u}, \mathbf{v}) = \langle \phi(\mathbf{u}), \phi(\mathbf{v}) \rangle_{\mathcal{H}}$
  If so, all occurrences of $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle_{\mathcal{H}}$ are replaced by $k(\mathbf{x}_i, \mathbf{x})$
- Keypoint: the 'focus' is sometimes only on k and not on ϕ
- Kernels must verify Mercer's property to be valid kernels
  - ensures that there exist a space $\mathcal{H}$ and a mapping $\phi : \mathcal{X} \rightarrow \mathcal{H}$ such that $k(\mathbf{u}, \mathbf{v}) = \langle \phi(\mathbf{u}), \phi(\mathbf{v}) \rangle_{\mathcal{H}}$
  - however non valid kernels have been used with success
  - and, research is in progress on using non semi-definite kernels
- k might be viewed as a similarity measure

The kernel trick (4/4)



- *Kernel trick* recipe
  - consider a nonlinear classification problem on $\mathcal{X} \times \mathcal{Y}$
  - choose a linear classification algorithm (expr. in terms $\langle \cdot, \cdot \rangle$)
  - replace all occurrences of $\langle \cdot, \cdot \rangle$ by a kernel $k(\cdot, \cdot)$
- Obtained classifier: $$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1,\ldots,m} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b \right)$$

Common kernels (1/2)

- Gaussian kernel
  - $k(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}\right), \quad \sigma^2 > 0$
  - the corresponding $\mathcal{H}$ is of infinite dimension
- Polynomial kernel
  - $k(\mathbf{u}, \mathbf{v}) = (\langle \mathbf{u}, \mathbf{v} \rangle + c)^d, \quad c \in \mathbb{R}, d \in \mathbb{N}$
  - a corresponding analytic ϕ may be constructed (see below)
- Tangent kernel (it is *not* a Mercer kernel)
  - $k(\mathbf{u}, \mathbf{v}) = \tanh(a\langle \mathbf{u}, \mathbf{v} \rangle + c), \quad a, c \in \mathbb{R}$

Common kernels (2/2)

- Let $k = \langle \mathbf{u}, \mathbf{v} \rangle^2_{\mathbb{R}^2}$ (polynomial kernel with $c = 0$ and $d = 2$) defined on $\mathbb{R}^2 \times \mathbb{R}^2$
- Consider the mapping:

$$\begin{array}{rccc} \phi: & \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ & \mathbf{x} = [x_1, x_2]^\top & \mapsto & \phi(\mathbf{x}) = \left[x_1^2, \sqrt{2}x_1x_2, x_2^2\right]^\top \end{array}$$

- We have, for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$:

$$\begin{aligned} \langle \phi(\mathbf{u}), \phi(\mathbf{v}) \rangle_{\mathbb{R}^3} &= \langle [u_1^2, \sqrt{2}u_1u_2, u_2^2]^\top, [v_1^2, \sqrt{2}v_1v_2, v_2^2]^\top \rangle \\ &= (u_1v_1 + u_2v_2)^2 \\ &= \langle \mathbf{u}, \mathbf{v} \rangle^2_{\mathbb{R}^2} \\ &= k(\mathbf{u}, \mathbf{v}) \end{aligned}$$

What to take home ?

- Classification
  - appears in many real world problems
  - can be done automatically
  - generalization is important
- Kernels
  - are wonderful !
  - field of active, thrilling research
  - classification of structured objects might be envisioned:
    - sequences: DNA strings, amino-acid strings, texts
      $\Rightarrow$ *Kernels and appl. to sequences*
    - graphs: structure of a molecule, disulfide bonds, GO
      $\Rightarrow$ *Kernels and appl. to graphs and molecular structures*
  - fusion of heterogeneous information
    $\Rightarrow$ *Combining Classifiers and Combining Kernels*

Miscellaneous information

- Contact information
  - Liva Ralaivola
  - Office CS/E 307
  - Webpage (not really up to date): http://www.ics.uci.edu/~liva/
- Useful places/resources/links
  - Kernel Club: every Friday starting Oct. 1^{st}, 2004, 11:00am, CS 432
  - Kernel Machines: http://www.kernel-machines.org/
  - Learning with Kernels: http://www.learning-with-kernels.org/
  - SVM applet: http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml

References

[Schölkopf and Smola, 2002] Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels, Support Vector Machines, Regularization, Optimization and Beyond*. MIT University Press.

Other references on my web page (soon)