6 Kernel Methods

Pattern Recognition and Machine Learning

Christopher M. Bishop

CC IRML Journal Club

Jérôme Kunegis, 2008–03–14

Outline

- Motivation 1: Weighted mean prediction
- Motivation 2: Regression with basis Φ
- Definition: Kernel
- Building kernels
- Gaussian kernel
- Regularized linear regression
- Gaussian processes
- Kernel trick
- Symbolic kernels

Motivation 1: Weighted Mean

- Observed vectors $\mathbf{x}_i$
- Prediction by weighted mean

$$y_{\mathrm{new}} = (\Sigma_i\, \mathbf{x}_i \cdot \mathbf{x}_{\mathrm{new}})^{-1}\, \Sigma_i\, \mathbf{x}_i \cdot \mathbf{x}_{\mathrm{new}}$$

$\mathbf{x}_1$	t_1
$\mathbf{x}_2$	t_2
$\mathbf{x}_3$	t_3
$\mathbf{x}_{\mathrm{new}}$	?

- Product $\mathbf{x}_i \cdot \mathbf{x}_j$ acts as similarity (correlation)
- Replace $\mathbf{x}_i \cdot \mathbf{x}_j$ with any function sim($\mathbf{x}_i$, $\mathbf{x}_j$).

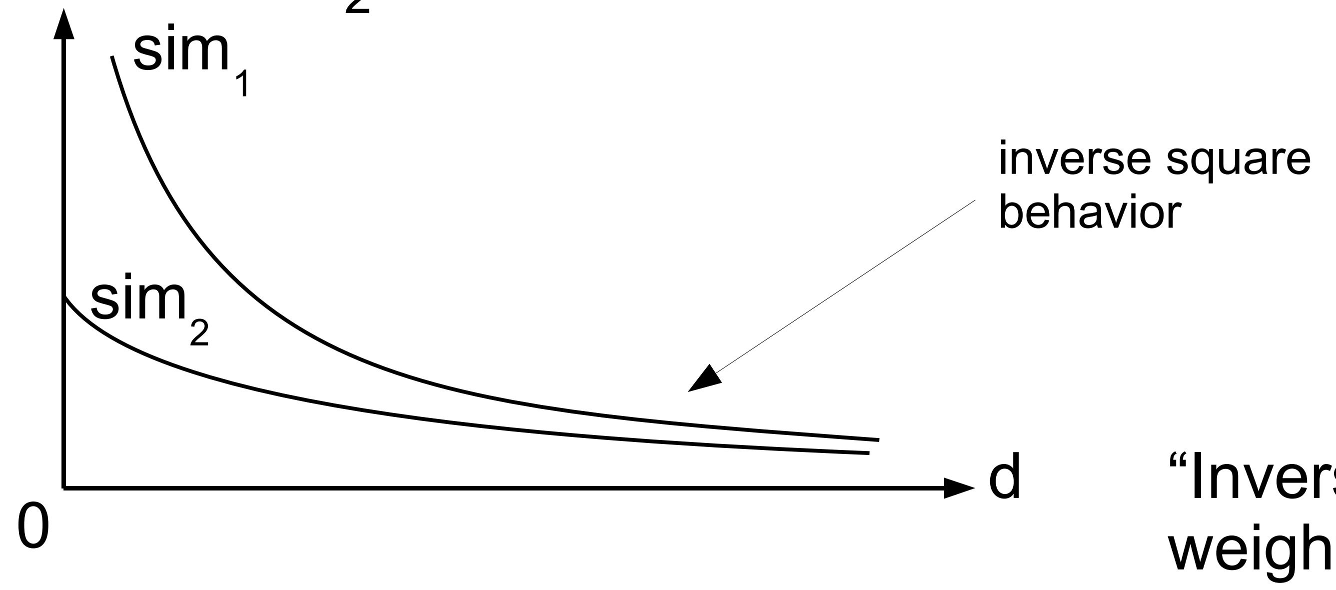
Similarity Function: Euclidian

- Base similarity function on Euclidian distance

$$d^2 = (\mathbf{x} - \mathbf{y})^2$$

$$\mathrm{sim}_1(\mathbf{x}, \mathbf{y}) = 1 \,/\, d^2$$

$$\mathrm{sim}_2(\mathbf{x}, \mathbf{y}) = 1 \,/\, (d^2 + \sigma^2)$$



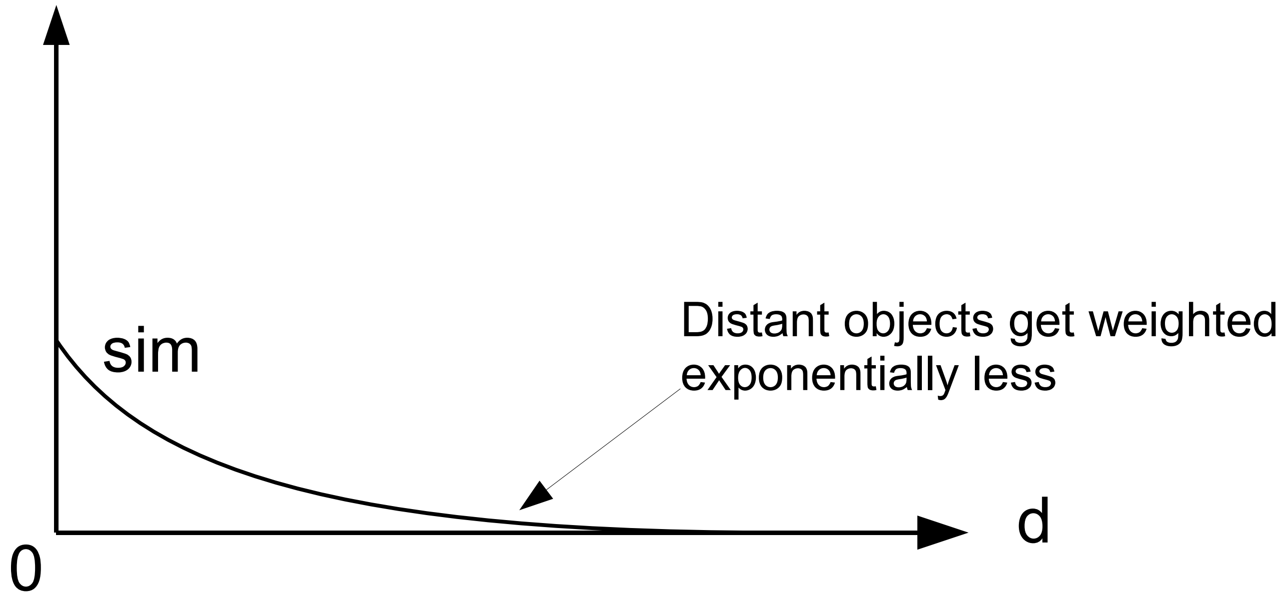
"Inverse distance weighting" (IDW)

Gaussian Similarity

- Inverse exponential behavior

$$\mathrm{sim}(\mathbf{x}, \mathbf{y}) = \exp\{-d^2 / 2\sigma^2\}$$

"Gaussian" function



Motivation 2: Regression

- Instead of $\mathbf{x}$, use $\Phi(\mathbf{x})$ as basis

$$\mathbf{x} \rightarrow \Phi(\mathbf{x})$$

- Examples:
  - Polynomial fitting: $\Phi(x) = (x^0, x^1, x^2, \ldots, x^k)$
  - Linear regression: $\Phi(\mathbf{x}) = (1, x_1, x_2, \ldots, x_N)$

$$\mathbf{x} \cdot \mathbf{y} \rightarrow \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$$

Definition: Kernel

- sim($\mathbf{x}$, $\mathbf{y}$) is a kernel $k(\mathbf{x}, \mathbf{y})$ when
  - sim($\mathbf{x}$, $\mathbf{y}$) = $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$ for some Φ
  - $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is positive semi-definite

Definition depends on X

- In general, Φ may be infinite dimensional
- K is the Gram[ian] matrix
- Linear kernel: $\Phi(\mathbf{x}) = \mathbf{x}$; $k(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$.

Building Kernels

Assemble kernels [:296]

- $f(\mathbf{x})\, k(\mathbf{x}, \mathbf{y})\, f(\mathbf{y})$ [any function f]
- $p(k(\mathbf{x}, \mathbf{y}))$ [polynomial p]
- $k_1(\mathbf{x}, \mathbf{y}) + k_2(\mathbf{x}, \mathbf{y})$
- $k_1(\mathbf{x}, \mathbf{y})\, k_2(\mathbf{x}, \mathbf{y})$
- $\exp\{k(\mathbf{x}, \mathbf{y})\}$
- $k(\Phi(\mathbf{x}), \Phi(\mathbf{y}))$
- $\mathbf{x}^T A \mathbf{x}$ [A pos. sem.-def.]
- $k_1(\mathbf{x}_1, \mathbf{y}_1) + k_2(\mathbf{x}_2, \mathbf{y}_2)$ [$x = (x_1, x_2)$, etc.]
- $k_1(\mathbf{x}_1, \mathbf{y}_1)\, k_2(\mathbf{x}_2, \mathbf{y}_2)$

Gaussian Kernel

$$k(\mathbf{x}, \mathbf{y}) = \exp\{-(\mathbf{x} - \mathbf{y})^2 / 2\sigma^2\}$$

- Φ is infinite dimensional [ex 6.11]
- $\Phi(\mathbf{x})$ has not to be calculated, only $k(\mathbf{x}, \mathbf{y})$

- Gaussian kernel is positive, inner product gets negative

Regularized Linear Regression

- Regularized linear regression (6.9):

$$t_{\text{new}} = k(\mathrm{X}, \mathbf{x}_{\text{new}})^{\mathrm{T}} \left(k(\mathrm{X}, \mathrm{X}) + \lambda\mathbf{I}\right)^{-1} t$$

- $\Phi(\mathbf{x})$ is not needed, only $k(\mathbf{x}, \mathbf{y})$

(cf. AL)

Gaussian Processes

- Gaussian prior on weight **w** for regression

⇒Function $y(\mathbf{x}, \mathbf{w})$ has Gaussian distribution

⇒Joint distribution on $y(\mathbf{x}_i, \mathbf{w})$:

Gaussian, with covariance K

- Gaussian process, Gaussian random field

- cf. Markov random fields [8.]

'Kernel Trick'

- Algorithm that uses only scalar product $\mathbf{x} \cdot \mathbf{y}$

$$\mathbf{x} \cdot \mathbf{y} \;\rightarrow\; k(\mathbf{x}, \mathbf{y})$$

- Replace scalar product with kernel
- Mapping into *higher dimension* makes data easier separable for classification, etc.
- Kernel variants of algorithms: kernel regression, kernel PCA, …

Symbolic Kernels

- Graph kernels: **x** is a node in a graph, e.g. *resistance distance*.
- e.g. $\exp\{-d^2 / 2\sigma\}$ for a distance d.

Conclusion

- Not covered: Nadaraya–Watson model, (Gaussian processes)
- Current IRML research:
  - AL: kernels for filter agent ensembles
  - JK: graph kernels for collaborative filtering
- Next chapters:
  - 7 Support vector machines (Leo)
  - 8 Graphical models (Winfried)
- Other kernels: $\{x \mid Ax = 0\}$