CSCE 990 Lecture 2: Kernels* CSCE 990 Lecture 2: Kernels* Stephen D. Scott January 12, 2006 - Let $K \subseteq \mathbb{R}^N$. We will consider the afth order product so farmels - The "kernel trick" - The "kernel trick" - Examples of kernels - The "kernel trick" - Examples of kernels - Sections 1.1, 2.1, 2.2.1–2.2.2, 2.2.6–2.2.7, 2.3 (also read Sections 2.2.3–2.2.4, 2.5) - Most figures @2002 MT Press, Bernhard Schölkopt, and Alex Smolu. - Introduction - Remember that a kernel is simply a dot prod- ut under some mapping - We'll go into this more formally later - Dot product \Rightarrow similarity measure - E.G.; $x_1 = (1/\sqrt{2}, 1/\sqrt{2}), x_2 = (1/1, 3, 1/1, 156),$ $x_1 = (1, 0) (x_1 = 1, vi)$ (x_1, x_2) $\approx 1/.1, 3 = (1/2, 1/2) = 0.9958$ (x_1, x_2) $\approx 1/.1, 3 = 1, 1/2, 21 = 0.9958$ (x_1, x_2) $\approx 1/.1, 3 = 1, 1/2, 21 = 0.9958$ (x_1, x_2) $\approx 1/.1, 1, 1, 1, 1, 1, 2.0, 2.1 = 0.9958$ (x_1, x_2) $\approx 1/.1, 1, 1, 1, 1, 2.0, 2.1 = 0.9958$ (x_1, x_2) $\approx 1/.1, 1, 1, 1, 2, 1, 2.2, 1 = 0.9958$ (x_1, x_2) $\approx 1/.1, 1, 1, 1, 1, 1, 2.50, 1, 1, 1, 1, 1, 1, 2.50, 1, 1, 1, 1, 1, 2.50, 1, 1, 1, 1, 1, 1, 1, 1, 2.50, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2.1 = 0.9958$ (x_1, x_2) $\approx 1/.1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2.1 = 0.9958$ (x_1, x_2) $\approx 1/.1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,$		
CSCE 990 Lecture 2: Kernels* Stephen D. Scott January 12, 2006 • Dot products as similarity measures • Example: Product features • Definitions • All kernels are dot products • The "kernel trick" • Examples of kernels • Examples of kernels • Sections 1.1, 2.1, 2.2.1–2.2.2, 2.2.6–2.2.7, 2.3 (also read Sections 2.2.3–2.2.4, 2.5) • Most figures (2002 MIT Press, Bernhard Schökkopf, and Akes Smolu. 1 • Most figures (2002 MIT Press, Bernhard Schökkopf, and Akes Smolu. • Most figures (2002 MIT Press, Bernhard Schökkopf, and Akes Smolu. • Most figures (2002 MIT Press, Bernhard Schökkopf, and Akes Smolu. • The "kernel trick" • Examples of kernels • Sections 1.1, 2.1, 2.2.1–2.2.2, 2.2.6–2.2.7, 2.3 (also read Sections 2.2.3–2.2.4, 2.5) • Product Features • Let $X \subseteq \mathbb{R}^N$. We will consider the dth order products of the entries $[x_1]_0$ of $x \in X^*$. $[x_{1,x_{2}}]_{1} = (1/\sqrt{2}, 1/\sqrt{2}), x_{2} = (1/1.3, 1/1.565),$ $x_{3} = (1, 0)$ ($ x_{1} = 1$ with $(x, x') = cost • So kernel k: X \times X \rightarrow \mathbb{R} gives measures ofsimilarity under its corresponding remapping\Phi: X \rightarrow \mathcal{H}• X is the original input space, where the la-beled training examples x_{1} come from$		Outline
Stephen D. Scott Stephen D. Scott January 12, 2006 • Carter of the series are dot products - The "kernel trick" • Examples of kernels • Carter of the series • Sections 1.1, 2.1, 2.2.1–2.2.2, 2.2.6–2.2.7, 2.3 (also read Sections 2.2.3–2.2.4, 2.5) • Most figures (20002 MIT Press, Bernhard Schölkopf, and Alex Smola. • Most figures (20002 MIT Press, Bernhard Schölkopf, and Alex Smola. • Introduction • Remember that a kernel is simply a dot prod- uct under some mapping • We'll go into this more formally later • Dot product \Rightarrow similarity measure • E.g.: $x_1 = (1/\sqrt{2}, 1/\sqrt{2}), x_2 = (1/1, 3, 1/1.565),$ $x_3 = (1, 0) x = 1 \%$ • Let $x \subseteq \mathbb{R}^N$. We will consider the d th order products of the entries $[z_1]$ of $x \in \mathcal{X}:$ $[x]_1, [x]_2, \cdots [x]_M$ for $j_1, \dots, j_d \in \{1, \dots, N\}$ • These are called product features, and H is the set of all product features. • Popular in image processing: • Let each x correspond to a vector of the pixel intensities of an entire image (smoothed to remove noise) • So kernel $k: X \times X \rightarrow \mathbb{R}$ gives measures of similarity under its corresponding remapping $e: X \rightarrow H$ • X is the original input space, where the la- beled training examples x_i come from	CSCE 990 Lecture 2: Kernels*	 Dot products as similarity measures
Stephen D. Scott January 12, 2006 • Definitions • All kernels are dot products - The "kernel trick" • Examples of kernels • Examples of kernels • Examples of kernels • Sections 1.1, 2.1, 2.2.1–2.2.2, 2.2.6–2.2.7, 2.3 (also read Sections 2.2.3–2.2.4, 2.5) • Most figures @2002 MIT Press, Bernhard Schölkopf, and Alex Smoia. • We'll go into this more formally later • We'll go into this more formally later • We'll go into this more formally later • Dot product \Rightarrow similarity measure • E.g.: $x_1 = (1/\sqrt{2}, 1/\sqrt{2}), x_2 = (1/1, 3, 1/1, 565), x_3 = (1/0) (x_1 = 1 vi)$ • $(x_1, x_2) \approx 1/.838 + 1/2.213 \approx 0.9958$ $(x_1, x_2) \approx 1/.438 + 1/2.213 \approx 0.9958$ $(x_1, x_3) = 1/\sqrt{2} + 0 \approx 0.707$ • If $ x = 1$ and $ x' = 1$, then $(x, x') = co-$ sine of angle between them • So kernel $k : X \times X \rightarrow \mathbb{R}$ gives measures of similarity under its corresponding remapping $0 : X \rightarrow \mathcal{H}$ • X is the original input space, where the la- beled training examples x_1 come from		• Example: Product features
$\begin{aligned} \text{January 12, 2006} \\ & \text{All kernels are dot products} \\ & - \text{The "kernel trick"} \\ & \text{Examples of kernels} \\ & Examples$	Stephen D. Scott	• Definitions
- The "kernel trick" $- The "kernel trick"$ $- The "kernel trick"$ $- Examples of kernels$ $- Examples of kernel$		 All kernels are dot products
 • Examples of kernels • Sections 1.1, 2.1, 2.2.1-2.2.2, 2.2.6-2.2.7, 2.3 (also read Sections 2.2.3-2.2.4, 2.5) • Most figures ©2002 MIT Press, Bernhard Schölkopf, and Alex Smola. 1 • Sections 1.1, 2.1, 2.2.1-2.2.2, 2.2.6-2.2.7, 2.3 (also read Sections 2.2.3-2.2.4, 2.5) • Sections 1.1, 2.1, 2.2.1-2.2.2, 2.2.6-2.2.7, 2.3 (also read Sections 2.2.3-2.2.4, 2.5) • Most figures ©2002 MIT Press, Bernhard Schölkopf, and Alex Smola. • Introduction • Remember that a kernel is simply a dot product features • We'll go into this more formally later • Dot product ⇒ similarity measure • E.g.: x₁ = (1/√2, 1/√2), x₂ = (1/1.3, 1/1.565), x₃ = (1.0) (x₁ = 1 vi) (x₁, x₂) ≈ 1/√2 + 0 ≈ 0.707 • If x₁ = 1 and x' = 1, then ⟨x, x'⟩ = cosine of angle between them • So kernel k : X × X → ℝ gives measures of similarity under its corresponding remapping φ : X → H • X is the original input space, where the labeled training examples x_i come from 	January 12, 2006	– The "kernel trick"
 Most figures @2002 MIT Press, Bernhard Schölkopf, and Alex Smola. Introduction Remember that a kernel is simply a dot product under some mapping We'll go into this more formally later Dot product ⇒ similarity measure E.g.: x₁ = (1/√2, 1/√2), x₂ = (1/1.3, 1/1.565), x₃ = (1,0) (x_i = 1 vi) (x₁, x₂) ≈ 1/√2 + 0 ≈ 0.707 If x = 1 and x' = 1, then (x, x') = cossine of angle between them So kernel k : X × X → R gives measures of similarity under its corresponding remapping Φ : X → H X is the original input space, where the labeled training examples x_i come from Sections 1.1, 2.1, 2.2.1-2.2.2, 2.2.6-2.2.7, 2.3 (also read Sections 2.2.3-2.2.4, 2.5)		 Examples of kernels
 Most figures ©2002 MIT Press, Bernhard Schölkopf, and Alex Smola. 1 Introduction Remember that a kernel is simply a dot product under some mapping We'll go into this more formally later Dot product ⇒ similarity measure E.g.: x₁ = (1/√2, 1/√2), x₂ = (1/1.3, 1/1.565), x₃ = (1, 0) (x_i = 1 ∀i) (x₁, x₂) ≈ 1/1.838 + 1/2.213 ≈ 0.9958 (x₁, x₃) = 1/√2 + 0 ≈ 0.707 If x = 1 and x' = 1, then (x, x') = cosine of angle between them So kernel k : X × X → ℝ gives measures of similarity under its corresponding remapping φ : X → H X is the original input space, where the labeled training examples x_i come from 		 Sections 1.1, 2.1, 2.2.1–2.2.2, 2.2.6–2.2.7, 2.3 (also read Sections 2.2.3–2.2.4, 2.5)
1 2 Introduction Froduct Features • Remember that a kernel is simply a dot product under some mapping • Let X ⊆ ℝ ^N . We will consider the dth order products of the entries [x] _j of x ∈ X: • We'll go into this more formally later • Let X ⊆ ℝ ^N . We will consider the dth order products of the entries [x] _j of x ∈ X: • Dot product ⇒ similarity measure • Let X ⊆ ℝ ^N . We will consider the dth order products of the entries [x] _j of x ∈ X: • (1/√2, 1/√2), x2 = (1/1.3, 1/1.565), x3 = (1,0) (x_i = 1 vi) • These are called product features, and H is the set of all products of d entries • 1/√2 + 0 ≈ 0.707 • So kernel k : X × X → ℝ gives measures of similarity under its corresponding remapping e: X → H • Let each x correspond to a vector of the pixel intensities of an entire image (smoothed to remove noise). • So kernel k : X × X → ℝ gives measures of similarity under its corresponding remapping e: X → H • Let each x correspond to a subset (up to size d) of pixels from the image x.	*Most figures $\textcircled{O}2002$ MIT Press, Bernhard Schölkopf, and Alex Smola.	
Introduction • Remember that a kernel is simply a dot product under some mapping – We'll go into this more formally later • Dot product \Rightarrow similarity measure – E.g.: $x_1 = (1/\sqrt{2}, 1/\sqrt{2}), x_2 = (1/1.3, 1/1.565), x_3 = (1, 0) (x_i = 1 \forall i)$ $\langle x_1, x_2 \rangle \approx 1/1.838 + 1/2.213 \approx 0.9958$ $\langle x_1, x_3 \rangle = 1/\sqrt{2} + 0 \approx 0.707$ – If $ x = 1$ and $ x' = 1$, then $\langle x, x' \rangle = \text{cossine of angle between them}$ • So kernel $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ gives measures of similarity under its corresponding remapping $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ - \mathcal{X} is the original input space, where the labeled training examples x_i come from	1	2
 Remember that a kernel is simply a dot product under some mapping We'll go into this more formally later Dot product ⇒ similarity measure E.g.: x₁ = (1/√2, 1/√2), x₂ = (1/1.3, 1/1.565), x₃ = (1,0) (x_i = 1 ∀i) ⟨x₁, x₂⟩ ≈ 1/1.838 + 1/2.213 ≈ 0.9958 ⟨x₁, x₃⟩ = 1/√2 + 0 ≈ 0.707 If x = 1 and x' = 1, then ⟨x, x'⟩ = cosimilarity under its corresponding remapping Φ : X → H X is the original input space, where the labeled training examples x_i come from Let X ⊆ ℝ^N. We will consider the dth order products of the entries [x]_j of x ∈ X: [x]_{j1} · [x]_{j2} ··· [x]_{jd} for j₁,, j_d ∈ {1,,N} These are called product features, and H is the set of all products of d entries Popular in image processing: Let each x correspond to a vector of the pixel intensities of an entire image (smoothed to remove noise) 		
- <i>H</i> is the <u>feature space</u> , which is where we'll search for a separating hyperplane	Introduction	Product Features
	 Introduction Remember that a kernel is simply a dot product under some mapping We'll go into this more formally later Dot product ⇒ similarity measure E.g.: x₁ = (1/√2, 1/√2), x₂ = (1/1.3, 1/1.565), x₃ = (1,0) (x_i = 1 ∀i) (x₁, x₂) ≈ 1/1.838 + 1/2.213 ≈ 0.9958 (x₁, x₃) = 1/√2 + 0 ≈ 0.707 If x = 1 and x' = 1, then ⟨x, x'⟩ = cosine of angle between them So kernel k : X × X → ℝ gives measures of similarity under its corresponding remapping φ : X → H X is the original input space, where the labeled training examples x_i come from H is the feature space, which is where we'll search for a separating hyperplane 	 Product Features Let X ⊆ ℝ^N. We will consider the dth order products of the entries [x]_j of x ∈ X: [x]_{j1} · [x]_{j2} ··· [x]_{jd} for j1,, jd ∈ {1,,N} These are called product features, and H is the set of all products of d entries Popular in image processing: Let each x correspond to a vector of the pixel intensities of an entire image (smoothed to remove noise) Each product feature in Φ(x) is related to a logical "and" of a subset (up to size d) of pixels from the image x

Product Features

(cont'd)

- E.g. $\Phi(([x]_1, [x]_2)) = ([x]_1^2, [x]_2^2, [x]_1[x]_2)$
- Problem: If x has N dimensions, then for orderd products, the dimension of Φ(x) is

$$N_{\mathcal{H}} = {\binom{d+N-1}{d}} \ge {\left(\frac{d+N-1}{d}\right)^d}$$

- E.g. images that are 16 \times 16 (N= 256) and d= 5 yield $N_{\mathcal{H}}\approx$ 10 10
- But if we're only concerned about the dot products, then we can define $\Phi_d(x)$ such that

$$\left\langle \Phi_d(x), \Phi_d(x') \right\rangle = \left\langle x, x' \right\rangle^d = k(x, x') ,$$

which is easy to compute

- E.g.
$$\Phi_2(x) = ([x]_1^2, [x]_2^2, \sqrt{2}[x]_1[x]_2)$$

- Can also use $k(x,x') = (\langle x,x'\rangle + 1)^d$ to get terms of degree $\leq d$

5

Definitions

- Up to now, we assumed $\mathcal{X} \subseteq \mathbb{R}^N$. For the rest of this course, \mathcal{X} can be arbitrary, e.g. sequences of letters from some alphabet (such as protein sequences)
- \bullet We will require the range of kernels to be $\mathbb{R},$ even though the book allows it to be complex
- **D2.3** Given a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and patterns $x_1, \ldots, x_m \in \mathcal{X}$, the $m \times m$ matrix K (where $K_{ij} = k(x_i, x_j)$) is called the <u>Gram matrix</u> or <u>kernel matrix</u> of k wrt x_1, \ldots, x_m
- **D2.4** A real, symmetric $m \times m$ matrix K that satisfies $\langle \mathbf{x}, K\mathbf{x} \rangle \geq 0$ for all $\mathbf{x} \in \mathcal{H}$ is positive definite. Equivalently, K is PD if it is symmetric and satisfies $\sum_{i,j} c_i c_j K_{ij} \geq 0$ for all $c_i, c_j \in \mathbb{R}$
 - PD \Leftrightarrow all eigenvalues ≥ 0



Definitions (cont'd)

- **D2.5** A function k on $\mathcal{X} \times \mathcal{X}$ which for all positive integers m and all $x_1, \ldots, x_m \in \mathcal{X}$ yields a PD Gram matrix is called a positive definite kernel, aka kernel, reproducing kernel, Mercer kernel, admissible kernel, support vector kernel, nonnegative definite kernel, positive semidefinite kernel, covariance function
 - Properties of PD kernels:
 - 1. If Φ maps \mathcal{X} to \mathcal{H} , then $\langle \Phi(x), \Phi(x') \rangle$ is a PD kernel on $\mathcal{X} \times \mathcal{X}$
 - 2. $k(x,x) \ge 0$ for all $x \in \mathcal{X}$
 - 3. Cauchy-Schwarz: $k(x, x')^2 \leq k(x, x)k(x', x')$
 - 4. k(x,x) = 0 for all $x \in \mathcal{X}$ implies k(x,x') = 0 for all $x, x' \in \mathcal{X}$

7

8

All Kernels are Dot Products

- \bullet All kernels are dot products in some feature space ${\cal H}$
- Consider a kernel k and some $x \in \mathcal{X}$
- Then $\Phi(x)(\cdot) = k(\cdot, x)$ is a function that measures similarity of all $x' \in \mathcal{X}$ to x
 - I.e. $\Phi(x)(x') = k(x', x)$
 - One such function for each $x \in \mathcal{X}$



• Can now think of each $x \in \mathcal{X}$ as a function over \mathcal{X}

The "Kernel Trick"

• Thus we see that any algorithm formulated in terms of a PD kernel k can be changed by

replacing k with another PD kernel k'

• Holds for any algorithm, not just SVMs

All Kernels are Dot Products (cont'd)

- We can turn the set of functions $\Phi(\mathcal{X})$ into a linear space
- Let m, m' be positive ints, $\alpha_i, \beta_j \in \mathbb{R}$, and $x_1, \ldots, x_m, x'_1, \ldots, x'_{m'} \in \mathcal{X}$ be arbitrary
- Let

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) \qquad g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x'_j)$$

and define the dot product as

$$\langle f,g\rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j) \tag{1}$$

• Can show that (1) is a valid dot product and that

$$\langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x')$$
,

which implies

$$\langle \Phi(x), \Phi(x') \rangle = k(x, x')$$

10

Examples of Kernels

- Polynomial: $k(x,x') = \left(\langle x,x' \rangle + c
 ight)^d$ for $c \geq 0$
 - When c = 0, then k is invariant under all rotations and mirroring operations of \mathcal{X}
- Gaussian radial basis function (Gaussian RBF):

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

with $\sigma > 0$

- Invariant under rotations and translations
- Its remapping has $\|\Phi(x)\| = 1$ for all $x \in \mathcal{X}$
- Sigmoid: $k(x, x') = \tanh(\kappa \langle x, x' \rangle + \vartheta)$ with $\kappa > 0$ and $\vartheta < 0$
 - Invariant under rotations
 - Not PD, but still used in practice

9

	Empirical Kernel Map
Examples of Kernels (cont'd)	• Given a kernel k and a data set $Z = \{z_1, \dots, z_n\}$, can define an <u>empirical kernel map</u> $\Phi_m(x) = (k(z_1, x), \dots, k(z_n, x))^\top$
• Can make new kernels from other kernels: if k_1 and k_2 are PD kernels, then so are $\Rightarrow \alpha k_1$ for all $\alpha \ge 0$ $\Rightarrow k_1 + k_2$ $\Rightarrow k_1 k_2$ $\Rightarrow k(A, B) := \sum_{x \in A, x' \in B} k_1(x, x')$, where $A, B \subseteq \mathcal{X}$ - More on this later	 I.e. remap x to a new representation based on its similarities to the patterns in Z Can then use each Φ_m(x) as training patterns in an SVM, etc. Can feed pairs of Φ_m(x) into a different kernel k' If k' is a straight dot product, then this is the same as squaring K, k's Gram matrix This remapping is valid even if k is not PD!
13	14
<text></text>	