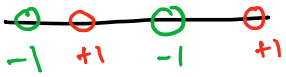
Ch 16 Kernel Methods

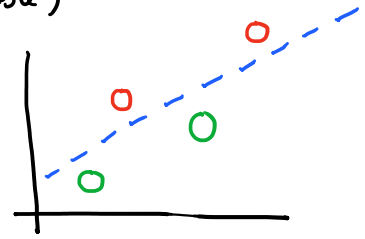
Friday, March 27, 2020 3:05 PM

A common trick is to **embed** data x into a higher dimension $\psi(x)$, especially because if the embedding ψ is nonlinear, then a linear classifier in high-dimensional space corresponds to a non-linear classifier in the original space (but the non-linearity is done in pre-processing, conceptually at least, so it doesn't hurt things like convexity)

Example 1D points (not linearly separable)



embed into 2D via $\psi: \mathbb{R}^1 \to \mathbb{R}^2$



... now in 2D, the dataset is linearly separable

putting in #'s to make a real example:

Consider a dataset w/ "x" as $\{-10, -9, \ldots, 9, 10\}$

let labels y be defined as $y = \begin{cases} +1 & |x| > 2 \\ -1 & |x| \leq 2 \end{cases}$

so not lin. separable

but define $\psi: \mathbb{R}^1 \to \mathbb{R}^2$ as $\psi(x) = (x, x^2)$, and now it is lin. sep. in $\mathbb{R}^2$

So, in theory, it's simple:

$\psi: X \to F$ ← new feature space,

preprocess data. Our distribution D over $X \times Y$ is now extended to D^{ψ} over $F \times Y$

in the obvious way, so $L_{D^\psi}(h) = L_D(h \circ \psi)$

we've even seen this trick before: we never discussed quadratic or polynomial regression because we can treat them as a special case of linear regression (after adding extra features)

For most learning algorithms to work, we'll need F to at least be a Hilbert space.

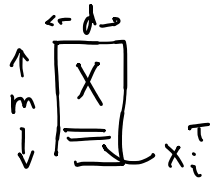
The kernel trick

Often it's unclear what ψ to use, and/or we don't want to work in F (why? it's larger-dim, even ∞-dim!)

The kernel trick is a way around this, applicable to many (not all) algorithms.

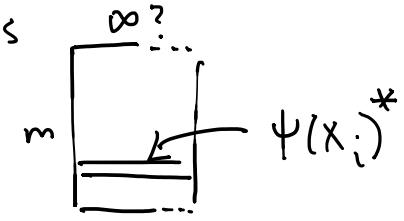
Ex Kernel Ridge Regression (Saunders et al., ICML '98)
(my simplified derivation w/o duality)

$$\min_w \frac{1}{2}\|Xw - y\|^2 + \lambda\|w\|^2$$



Solution is $w = (X^TX + \lambda I)^{-1} \cdot \underbrace{X^Ty}_{b = \sum_{i=1}^m y_i X_i}$

Now, suppose X is



our sol'n is still valid in any Hilbert space, but

$$X^TX = \sum_{i=1}^m \psi(x_i)\psi(x_i)^*$$

isn't easy to compute

Trick:

$$(X^TX + \lambda I)^{-1} = I - X^T(I + XX^T)^{-1}X$$

choose $\lambda = 1$ for simplicity

(matrix inversion lemma)

How does that help? $X^TX = \sum_i \psi(x_i)\psi(x_i)^*$ is tricky → a $\tilde{d} \times \tilde{d}$ matrix if $\psi: \mathbb{R}^d \to \mathbb{R}^{\tilde{d}}$

$$(XX^T)_{ij} = \psi(x_i)^*\psi(x_j) = \langle \psi(x_i), \psi(x_j) \rangle$$

$K(x, \tilde{x}) = \langle \psi(x), \psi(\tilde{x}) \rangle$ is a kernel

So, ψ defines K, but...

you can define K without needing to know what ψ is!

In particular, let the kernel matrix be K such that ← aka Gram matrix

$$(K)_{ij} = k(x_i, x_j) = \langle \psi(x_i), \psi(x_j) \rangle, \quad K \in \mathbb{R}^{m \times m}$$

So, back to ridge regression,

$$w = (I - X^T(I + K)^{-1}X) \cdot b, \qquad b = X^Ty = \sum y_i \psi(x_i)$$

$$= b - X^T\tilde{K}Xb, \qquad \tilde{K} = (I+K)^{-1} \in \mathbb{R}^{m \times m}$$

↑ w/o knowing ψ, this isn't computable

$\tilde{K}$ computable

but... we don't actually want w, we want to apply it, i.e., $h(x) = \langle w, x \rangle$

So, can we compute $h(x) = \langle w, x \rangle$?

or, since we're embedding x to $\psi(x)$, can we compute $h(x) = \langle w, \psi(x) \rangle$?

yes! $w = b - X^T\tilde{K}Xb$

↳ $\langle b, \psi(x) \rangle = \langle \sum y_i \psi(x_i), \psi(x) \rangle$

$$= \sum y_i \langle \psi(x_i), \psi(x) \rangle = \sum y_i k(x_i, x)$$

computable

and

$$\langle X^T \tilde{K} X b, \psi(x) \rangle = \langle \psi(x), X^T \tilde{K} X^T b \rangle$$

$$= c^T \tilde{K} d \quad \text{where} \quad c_i = \langle \psi(x), \psi(x_i) \rangle$$

$$d_i = \langle \psi(x_i), \sum_{i'} y_{i'} \psi(x_{i'}) \rangle$$

So all computable via only knowing $K(x, \tilde{x})$ (ψ is completely implicit)!

That's the kernel trick

General theory

The kernel trick applied to ridge regression.

Does it apply to SVM? (yes)

What else?

Thm 16.1 "Representer Thm"

Suppose $\psi: \mathcal{X} \to \mathcal{H}$ ← Hilbert space now, not hypothesis class

and let $\phi \neq \operatorname{argmin}_w f(\langle w, \psi(x_1) \rangle, \ldots, \langle w, \psi(x_m) \rangle) + R(\|w\|)$

for arbitrary f and R is nondecreasing on $\mathbb{R}_+$

(ex: $f(a_1, \ldots, a_m) = \frac{1}{m} \sum_i l(a_i)$, $R(a) = a^2$

Note Hard + Soft SVM, and Ridge Regression, satisfy the assumptions)

then $\exists \alpha \in \mathbb{R}^m$ s.t.

$w = \sum_i \alpha_i \psi(x_i)$ is optimal.

(ie., $w \in \operatorname{span}(\{\psi(x_1), \ldots, \psi(x_m)\})$ and we can use the kernel trick, since now $\langle w, \psi(x) \rangle$ is computable)

Proof:

let w^* be an optimal soln, decompose it as

$$w^* = w + u, \quad w \in \operatorname{span}(\{\psi(x_1), \ldots, \psi(x_m)\}) = V \subseteq \mathcal{H}$$

$$u \in V^\perp \quad \text{since} \quad \mathcal{H} = V \oplus V^\perp \quad (\dim(V) < \infty)$$

So via orthogonality,

$$\|w^*\|^2 = \|w\|^2 + \|u\|^2 \Rightarrow \|w\| \leq \|w^*\|$$

and R nondecreasing $\Rightarrow R(\|w\|) \leq R(\|w^*\|)$

Since w^* and w differ only on $V^\perp$,
they both have the same value of
$f(\langle w, \psi(x_1)\rangle, \dots \langle w, \psi(x_m)\rangle)$

hence if w^* is optimal, w must be optimal
as well, and $w \in V$ as desired. □

(book has slight error: it didn't assume optimal solution exists)

Conclusion: Kernel-SVM is practical → either use dual problem, or primal problem using $\varphi(x_i)$ as the basis

Examples of Kernels

Given $K(\cdot,\cdot)$, sometimes we have an explicit expression for ψ, but not always

(and not just any $K(\cdot,\cdot)$ works ... more on that later)

① Polynomial Kernel

$K(x,x') = (c + \langle x, x'\rangle)^k$ (ψ not unique)

$= \langle \psi(x), \psi(x')\rangle$ for $\psi: \mathbb{R}^d \to \mathbb{R}^{\tilde{d}}$, $\tilde{d} = \binom{d+k}{k}$ or $\tilde{d} = (d+1)^k$

ex, $d=2$,

$$K(x,x') = (c + x_1x_1' + x_2x_2')^2 = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\,x_1x_2 \\ \sqrt{2c}\,x_1 \\ \sqrt{2c}\,x_2 \\ c \end{bmatrix}^T \begin{bmatrix} x_1'^2 \\ x_1'^2 \\ x_2'^2 \\ \sqrt{2}\,x_1'x_2' \\ \sqrt{2c}\,x_1' \\ \sqrt{2c}\,x_2' \\ c \end{bmatrix}$$

for larger d, you can form ψ, but you don't want to

② Gaussian Kernel (aka RBF, Radial Basis Fcn, or "SE" Squared Exponential)

most common in practice

$K(x,x') = e^{-\frac{\|x-x'\|^2}{2\sigma}}$ ← for $\sigma > 0$ ...

or often we write $e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$ to make it clear $\sigma^2 > 0$ and in analogy to normal distr.

what is ψ?

consider $d=1$ case, $K(x,x') = e^{-(x-x')^2/2}$ ($\sigma=1$ for now)

define $\psi: \mathbb{R} \to \ell^2$ by ℓ^2 on $\{0,1,2,\ldots\}$

$$(\psi(x))_n = \frac{1}{\sqrt{n!}} e^{-x^2/2} x^n$$

So $\|\psi(x)\|_{\ell^2}^2 = e^{-x^2} \cdot \left(\sum \frac{1}{n!} (x^2)^n\right)$

$= 1$ since $e^a = \sum \frac{1}{n!} a^n$ has ∞ radius of convergence

and note

$$\langle \psi(x), \psi(x') \rangle_{\ell^2} = \sum (\ldots)(\ldots)$$

$$= \exp\left(-\frac{x^2}{2} - \frac{x'^2}{2}\right) \sum_{n=0}^{\infty} \frac{(xx')^n}{n!}$$

$$= e^{xx'}$$

$$= \exp\left(-x^2/2 + xx' - x'^2/2\right)$$

$$= \exp\left(-(x-x')^2/2\right)$$

(3) Sigmoid Kernel

$a, b \geq 0$, $K(x,x') = \tanh\left(a\langle x, x' \rangle + b\right)$

(w/ SVM, this is similar to a simple neural network)

(4) discussion

The choice of kernel is one way to encode prior knowledge (just as choice of features was)

Similar to classic statistics: for regression, if x_i is a covariate, sometimes you choose to use $\log(x_i)$ instead

Often polynomial or Gaussian kernels are the only choices considered (and their parameters are hyperparameters that are fit)

... but kernels also used many places
(ex: Gaussian processes, Kriging in spatial stat.)
and there, often custom kernels are used.

For more ex., see exercises in ch 6 in Mohri et al.

What are the restrictions on K?

Lemma 16.2 (Mercer's Thm, simplified)

A symmetric function $K: X \times X \to \mathbb{R}$ implements an inner product in some Hilbert space iff K is psd (possemidef.)

A kernel is psd if $\forall m$, $\forall \{x_1, \dots, x_m\}$, the Gram matrix $G_{ij} = K(x_i, x_j)$ is a psd matrix

$G = G^T$, $G \succeq 0$ iff $\lambda_i(G) \geq 0$ $\forall i$

Turns out not all Hilbert spaces can be described w/ a kernel
Those that can are Reproducing Kernel Hilbert Spaces (RKHS)

In particular, a RKHS has the property that it is over an underlying domain Ω (think of $L^2(\Omega)$ for $\Omega = [0,1]$ or $\Omega = \mathbb{R}$, or $l^2(\Omega)$ for $\Omega = \mathbb{N}$ or $\mathbb{Z}$)

and if $x \in \Omega$,

$f \in \mathcal{H}$, then $\delta_x \in \mathcal{H}^*$ $(\equiv \mathcal{H})$

ex: $\mathcal{H} = l^2(\mathbb{N})$, $\Omega = \mathbb{N}$,

$f \in l^2$ is $f = (f_1, f_2, f_3, \dots)$

then $\delta_n(f) = f_n$ is a bounded linear functional

$l^2(\mathbb{N})$ is a RKHS (so is $l^2(\mathbb{Z})$)

ex: $\mathcal{H} = L^2(\Omega = [0,1])$

$f \in L^2$, $x \in [0,1]$,

$\delta_x(f) = f(x)$ $\left(= \int f(x')\delta(x'-x)dx' \right)$ in physics notation

this is not bounded

(not even well-defined, since L^2 is not defined pointwise)

So $L^2([0,1])$ is not a RKHS

(nor is $L^2(\mathbb{R})$)

History

Kernels in general, Mercer (1909)

Schoenberg (1938)

★ Aronszajn (1950)

In ML, Boser, Guyon, Vapnik '92 (SVM)

Schölkopf et al '98 (general)

Supplement

Forming the Gram matrix is slow!

$O(m^2 d)$ computation

many attempts to improve

One broad strategy is the Nyström method (also used in integral eq'n) which picks out representative pts.

A well-known 2007 paper (Rahimi, Recht) "Random Fourier Features"

makes approx. maps Ψ st.

$$K(x,x') \approx \langle \Psi(x), \Psi(x') \rangle$$

for Shift-invariant kernels

$$K(x,x') = G(x-x')$$ Ex

Gaussian $\exp(-\frac{1}{2}\|x-x'\|^2)$

Laplacian $\exp(-\|x-x'\|)$

Cauchy $\prod_{i=1}^{d} \frac{2}{1+(x_i-x_i')^2}$

relying on

Bochner's Thm A cts kernel of the form $K(x,x')=G(x-x')$ defined over a locally compact set X is pos. def. iff G is the Fourier transform of a non-neg. measure, i.e., $G(x)=\int_X e^{i\omega x} p(\omega)d\omega$

For standard shift-inv. kernels, $p(\omega)$ is known

Idea: approximate $G(x)$ via Monte Carlo integration: draw a set

$\{\omega_i\}_{i=1}^{N}$, iid, $\omega_i \sim p(\omega)$, equiv. to an approx. feature map into $\mathbb{R}^N$. Of course MC converges very slowly but not too bad if X has small covering number

(and in practice, low-accuracy is ok, since we don't need to approximate the kernel very well, just enough to give us "features" to learn from)

Other Kernel methods

probably 3 most common Kernel methods

1. Kernel-SVM
2. Kernel-ridge regression

] we discussed

3. Kernel-PCA → see §14.4.4 in Kevin Murphy's text (trick for centering data)

or more generally, inside a Generalize Linear Model (GLM)

cf. §14.3 Murphy

4. Kernelized nearest-neighbor and K-medoids

See §14.6 Murphy for comparison of kernel methods

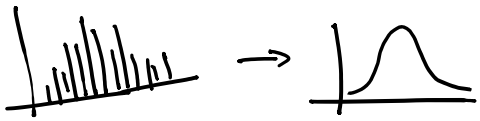
"Kernel" is used in other contexts too, eg,

1D kernel $K(x)$ s.t. $\int K(x)dx = 1$, $\int x K(x) dx = 0$,

like Gaussian kernel $\int x^2 K(x)dx > 0$

used for Kernel density estimation or Parzen window dens. est.

Idea is to turn an empirical histogram into an estimate of the pdf



Other examples of kernels

Matern kernel, common in GP, Kriging ← Gaussian Processes, see next lecture

based on modified Bessel function

(shift invariant)

String kernels for natural language processing (NLP)

See Murphy for more