Modern Topics in Learning Theory

Maria-Florina Balcan
04/26/2006

Outline

• Kernels & Large Margin Classifiers
  - Hot topic in recent years
• Kernels as Features [BBV04]
• General Similarity Functions [BB06]
  - Hot topic in the near future

Standard Supervised Learning

• X - instance space
• S={(x_i, l_i)} - set of labeled examples
  - x_i, x_{i+1}, ... drawn i.i.d. from some distr. D over X and labeled by target concept c
  - l_i \in \{-1,1\} - binary classification
• Do some optimization over S to find h with small error over D.
  - err(h)=P_{x \sim D}[h(x) \neq c(x)] \rightarrow error of h w.r.t. to c (and D)

Linear Separators

• Well studied and understood.
• Instance space: X=\mathbb{R}^n
• Hypothesis class - class of linear decision surfaces in \mathbb{R}^n
  - h(x)=w \cdot x + b, if h(x) \geq 0, then label x as positive (+1), otherwise label it as negative (-1)

Nonlinear Classification

• IDEA: Map each point to a higher dimensional feature space and construct linear separator in that higher dimensional space.
Nonlinear Classification

\[ \phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \quad (x_1, x_2) \rightarrow (x_1^2, x_2^2, \sqrt{2}x_1x_2) \]

\[ \psi(x) \psi(x') = (x_1^2, x_2^2, \sqrt{2}x_1x_2)^T \]

Kernels - Main Idea

- **Kernels - Main Idea**
  - K(., .) - kernel if it can be viewed as a *legal* definition of inner product:
    - \( \exists \phi : \mathbb{X} \rightarrow \mathbb{R}^n \) such that \( K(x,y) = \phi(x) \cdot \phi(y) \)
    - range of \( \phi \) - "o-space"
    - \( N \) can be very large
    - But think of \( \phi \) as implicit, not explicit!
  - Examples
    - Polynomial Kernel: \( X = \mathbb{R}^n \), \( K(x,y) = (1 + x \cdot y)^d \)
      - \( n = 3, d = 2, \phi : \mathbb{R}^3 \rightarrow \mathbb{R}^{10} \)
      - \( \phi(x) = (1, x_1^2, x_1^3, x_1^2x_2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_2, \sqrt{2}x_2, x_3, \sqrt{2}x_2x_3) \)

Kernels

- **Examples**:
  - Linear: \( K(x,y) = x \cdot y \)
  - Polynomial: \( K(x,y) = (1 + x \cdot y)^d \)
  - Gaussian: \( K(x,y) = \exp \left[ \frac{-||x-y||^2}{2\sigma^2} \right] \)
- **Closure Properties**
  - \( K(x,y) = K_i(x,y) + K_j(x,y) \)
  - \( K(x,y) = c \cdot K_i(x,y) \)
  - \( K(x,y) = K_i(x,y) \cdot K_j(x,y) \)
  - Easily create new kernels using basic ones!

Kernels

- **Kernels**
  - \( K \) is a kernel iff
    - \( K \) is symmetric
    - for any set of training points \( x_1, x_2, ..., x_n \) and for any \( a_1, a_2, ..., a_n \in \mathbb{R} \) we have:
      - \( \sum_{i,j} a_i a_j K(x_i, x_j) \geq 0 \)

Lin. Separators: Perceptron algorithm

- **Algorithm**:
  - Start with all-zeroes weight vector \( w \).
  - Given example \( x \), predict positive if \( w \cdot x \geq 0 \).
  - On a mistake, update as follows:
    - Mistake on positive, then update \( w \leftarrow w + x \)
    - Mistake on negative, then update \( w \leftarrow w - x \)
  - Easy to kernelize \( \rightarrow w \) is a weighted sum of examples:
    - \( w = a_1 x_1 + \cdots + a_i x_i \)
    - So, replace \( w \cdot x = a_1 x_1 \cdot x_1 + \cdots + a_i x_i \cdot x \) with
      - \( a_i K(x_i, x) \)
      - \( \sum_{i} a_i K(x_i, x) \)
Do we have good generalization?

- Standard SC - the amount of data we need depends on VC-dim of the hypothesis class.
  - VC-dim for the class of linear sep. in $\mathbb{R}^m$ is $m+1$.

- Then, do we pay a lot from sample size point of view for going up?

Large Margin Classifiers

- If $S$ is a set of labeled examples, then a vector $w$ has margin $\gamma$ w.r.t. $S$ if
  \[ \min_{(x,d)\in S} \frac{w^T x}{||w||^2} > \gamma \]

- A vector $w$ has margin $\gamma$ with respect to $P$ (the combined distribution over labeled examples) if
  \[ \Pr_{(x,d)\in P} \left[ \frac{w^T x}{||w||^2} < \gamma \right] = 0. \]

- If large margin, then the amount of data we need depends only on $\gamma$ and it’s independent on the dimension of the (instance) space!

Kernels & Large Margins

- If $S$ is a set of labeled examples, then a vector $w$ in the $\phi$-space has margin $\gamma$ if:
  \[ \min_{(x,d)\in S} \frac{w^T \phi(x)}{||w||^2} > \gamma \]

- A vector $w$ in the $\phi$-space has margin $\gamma$ with respect to $P$ if:
  \[ \Pr_{(x,d)\in P} \left[ \frac{w^T \phi(x)}{||w||^2} < \gamma \right] = 0. \]

- A vector $w$ in the $\phi$-space has error $\alpha$ at margin $\gamma$ if:
  \[ \Pr_{(x,d)\in P} \left[ \frac{w^T \phi(x)}{||w||^2} < \gamma \right] \leq \alpha \quad \text{($\alpha, \gamma$)-good kernel} \]
Outline

• Kernels & Large Margin Classifiers
  Hot topic in recent years
• Kernels as Features [BBV04]
• General Similarity Functions [BB06]
  Hot topic in the near future

Kernels as Features [BBV04]

• Main Idea:
  • Designing a kernel function is much like designing a feature space.
  • Given a good kernel K, we can reinterpret K as defining a new set of $O(1/\gamma^2)$ features.

Kernels as Features [BBV04]

• If indeed large margin under K, then a random linear projection of the $\phi$-space down to a low dimensional space approximately preserves linear separability.
  • by Johnson-Lindenstrauss lemma!

Main Idea - Johnson-Lindenstrauss lemma

- For any vectors $u,v$ with prob. $(1-\delta)$, $\angle(u,v)$ is preserved up to $\pm \gamma/2$, if we use $d = O\left(\frac{1}{\gamma^2} \log \frac{1}{\delta}\right)$.
- Usual use in algorithms: $m$ points, set $\delta=O(1/m^2)$.
- In our case, if we want w.h.p. $\exists$ separator of error $\epsilon$, use $d = O\left(\frac{1}{\epsilon^2} \log \frac{1}{\delta}\right)$.
- $\exists$ if $c$ has margin $\gamma$ in the $\phi$-space, then $F^*(D,c)$, will w.h.p. have a linear separator of error at most $\epsilon$.
Problem Statement

- For a given kernel K, the dimensionality and description of φ(x) might be large, or even unknown.
  - Do not want to explicitly compute φ(x).
- Given kernel K - produce such a mapping F efficiently:
  - running time that depends polynomially only on 1/γ and the time to compute K.
  - no dependence on the dimension of the "φ-space".

Main Result [BBV04]

- Positive answer - if our procedure for computing the mapping F is also given black-box access to the distribution D (unlabeled data).

Formally.....

- Given black-box access to K(·,·), given access to D and γ, ε, δ, construct, in poly time, F:X→R^d, where |d| = O(1/ε log(1/δ)) s.t. if c has margin γ in the φ-space, then with prob. 1 - δ, the induced distribution in R^d is separable with error ≤ ε.

3 methods (from simplest to best)

1. Draw d examples x_1, ..., x_d from D. Use:
   \[ F_d(x) = (K(x,x_1), ..., K(x,x_d)) \]
   For d = (8ε)(1/γ^2 + ln 1/δ), if P was separable with margin γ in φ-space, then w.h.p. this will be separable with error ε. (but this method doesn’t preserve margin).

2. Same d, but a little more complicated. Separable with error ε at margin γ/2.

3. Combine (2) with further projection as in JL lemma. Get d with log dependence on 1/ε, rather than linear. So, can set ε ≪ 1/d.

A Key Fact

Claim: If \exists unit-length w of margin γ in φ-space, then if draw x_1, ..., x_d \in D for d ≥ (8ε)(1/γ^2 + ln 1/δ), w.h.p. (1 - δ) exists w’ in span(φ(x_1), ..., φ(x_d)) of error ≤ ε at margin γ/2.

Proof: Let S = {φ(x_i)} for examples x drawn so far.
- w_m = proj(w; span(S)), w_m = w - w_m.
- Soc w_m is large if Pr_x(∥w_m · φ(x)∥ ≥ γ/2) ≥ ε; else small.
- If small, then done w’ = w_m.
- Else, next x has at least ε prob of improving S.
- \[ |w_m|^2 - |w_m'|^2 - (γ/2)^2 \]
- Can happen at most 4/γ^2 times.

A First Attempt

- If draw x_1, ..., x_d \in D for d = (8ε)(1/γ^2 + ln 1/δ), then w.h.p.
  exists w’ in span(φ(x_1), ..., φ(x_d)) of error ≤ ε at margin γ/2.

- So, for some w’ = α_1φ(x_1) + ... + α_dφ(x_d),
  \[ Pr_{x \in D}(||w’ · φ(x)|| ≥ |1|) ≤ ε. \]

- But notice that w’ · φ(x) = α_1K(x, x_1) + ... + α_dK(x, x_d).
  ⇒ vector (α_1, ..., α_d) is a separator in the feature space (K(x, x_1), ..., K(x, x_d)) with error ≤ ε.

- But margin not preserved because of length of target, examples.

A First Mapping

- Draw a set S = {x_1, ..., x_d} of d = O(1/ε) unlabeled examples from D.
- Run K(x, y) for all x, y \in S, get M(S) = (K(x, x_j))_{x \in S, y \in S}.
- Place S into d-dim. space based on K (or M(S)).
A First Mapping, cont

- What to do with new points?
- Extend the embedding \( F_t \) to all of \( X \):
  - consider \( F_t : X \rightarrow \mathbb{R}^d \) defined as follows: for \( x \in X \), let \( F_t(x) \in \mathbb{R}^d \) be the point such that \( F_t(x) \cdot F_t(x_i) = K(x_i,x) \), for all \( i \in \{1, ..., d\} \).
- The mapping is equivalent to orthogonally projecting \( \phi(x) \) down to \( \text{span}(\phi(x_1), ..., \phi(x_d)) \).

An improved mapping

- A two-stage process, compose the first mapping, \( F_t \), with a random linear projection.
- Combine two types of random projection:
  - a projection based on points chosen at random from \( D \).
  - a projection based on choosing points uniformly at random in the intermediate space.

Improved Mapping - Properties

- Given \( \epsilon, \delta, \gamma < 1 \), \( d = \Omega\left(\frac{1}{\gamma^2} \log \frac{1}{\delta}\right) \), if \( P \) has margin \( \gamma \) in the \( \phi \)-space, then with probability \( \geq 1-\delta \), our mapping into \( \mathbb{R}^d \), has the property that \( F(D,c) \) is linearly separable with error at most \( \epsilon \), at margin at most \( \gamma/4 \), given that we use \( n = \Omega\left(\frac{1}{\gamma^2} + \ln \frac{1}{\delta}\right) \) unlabeled examples.

Improved Mapping, Consequences

- If \( P \) has margin \( \gamma \) in the \( \phi \)-space then we can use \( n = \tilde{O}(1/\epsilon^2) \) unlabeled examples to produce a mapping into \( \mathbb{R}^d \) for \( d = O\left(\frac{1}{\gamma^2} \log \frac{1}{\epsilon^2} \right) \) such that w.h.p. data is linearly separable with error \( \leq \epsilon d / d \).
- The error rate of the induced target function in \( \mathbb{R}^d \) is so small that a set \( S \) of \( \tilde{O}(d/\epsilon^2) \) labeled examples will, w.h.p., be perfectly separable in \( \mathbb{R}^d \).
  - Can use any generic, zero-noise linear separator algorithm in \( \mathbb{R}^d \).

Implications, Open Problems

- Designing a kernel function -- designing a feature space.
- Alternative to "kernelizing" a learning algorithm:
  - rather than modifying the alg. to use kernels, construct instead a mapping into a low-diml space using the kernel and \( D \); then run any un-kernelized alg over examples drawn from the mapped distribution.
- Open problem: Produce the desired mappings \( F : X \rightarrow \mathbb{R}^d \) in an oblivious manner (without access to \( D \)) for natural/standard kernels.
Outline
• Kernels & Large Margin Classifiers
  Hot topic in recent years
• Kernels as Features [BBV04]
• General Similarity Functions [BB06]
  Hot topic in the near future

General Similarity Functions
Goal: definition of “good similarity function” for a learning problem that:
1. Talks in terms of more natural direct properties:
   • no implicit high-diml spaces
   • no requirement of positive-semidefiniteness
2. If \( K \) satisfies these properties for our given problem, then has implications to learning:
   • can’t just say any function is a good one
3. Is broad: includes usual notion of “good kernel” (one that induces a large margin separator in \( \phi \)-space).

A first Attempt: Definition satisfying properties (1) and (2)
• \( K : (x, y) \mapsto [-1, 1] \) is an \((\varepsilon, \gamma)\)-good similarity for \( P \) if at least a \( 1-\varepsilon \) prob mass of \( x \) satisfy:

\[
E_{y \sim P} [K(x, y) | \ell(y) = \ell(x)] \geq E_{y \sim P} [K(x, y) | \ell(y) \neq \ell(x)] + \gamma
\]

How can we use it?
At least a \( 1-\varepsilon \) prob mass of \( x \) satisfy:

\[
E_{y \sim P} [K(x, y) | \ell(y) = \ell(x)] \geq E_{y \sim P} [K(x, y) | \ell(y) \neq \ell(x)] + \gamma
\]

• Draw \( S^+ \) of \( O(\gamma^2 \ln(1/\delta^2)) \) positive examples.
• Draw \( S^- \) of \( O(\gamma^2 \ln(1/\delta^2)) \) negative examples.
• Classify \( x \) based on which gives better score.
• Hoeffding: for any given “good \( x \)”, prob of error over draw of \( S^+, S^- \) at most \( \delta \).
• So, at most \( \delta \) chance our draw is bad on more than \( \delta \) fraction of “good \( x \)”. So overall error rate \( \leq \varepsilon + \delta \).

But not broad enough
• \( K(x, y) = x \cdot y \) has good (large margin) separator but doesn’t satisfy the previous definition:
  • half of positives are more similar to negatives that to typical positives

But not broad enough
• Idea: would work if we didn’t pick \( y \)’s from top-left.
• Broaden to say: OK if \( \exists \) large region \( R \) s.t. most \( x \) are on average more similar to \( y \in R \) of same label than to \( y \in R \) of other label.
Broader Definition

- $K(x,y)\rightarrow[-1,1]$ is an $(\varepsilon, \gamma)$-good similarity for $P$ if exists a weighting function $w(y)\in[0,1]$ s.t. at least $1-\varepsilon$ mass of $x$ satisfy:
  \[ E_{y\sim P}[w(y)K(x,y)|\ell(y)=\ell(x)] \geq E_{y\sim P}[w(y)K(x,y)|\ell(y)\neq\ell(x)] + \gamma \]

- How to use it:
  - Draw $S^+ = \{y_1, \ldots, y_n\}, S^- = \{z_1, \ldots, z_n\}$. $n=\tilde{O}(1/\gamma^2)$
  - Use to "triangulate" data:
    - $F(x) = [K(x,y_1), \ldots, K(x,y_n), K(x,z_1), \ldots, K(x,z_n)]$.
  - Whp, exists good separator in this feature space
    - $w = [w(y_1), \ldots, w(y_n), -w(z_1), \ldots, -w(z_n)]$

- And furthermore
  - An $(\varepsilon, \gamma)$-good kernel is an $(\varepsilon', \gamma')$-good similarity function under this definition.
    - $\varepsilon' = \varepsilon + \varepsilon_{\text{extra}}, \gamma' = \gamma^2\varepsilon_{\text{extra}}$

Proof (very rough sketch):

- Set $w(y)=0$ for the $\varepsilon$ fraction of "bad" $y$'s.
- Imagine repeatedly running margin-Perceptron on multiple samples $S$ from remainder.
- Set $w(y) \propto \ell(y)E[\text{weight}(y) | y \in S]$

And furthermore

Good Kernels are Good Similarity Functions

- An $(\varepsilon, \gamma)$-good kernel [margin $\geq \gamma$ on at least $1-\varepsilon$ fraction of $P$] is an $(\varepsilon', \gamma')$-good sim fn under this definition.
  - $\varepsilon' = \varepsilon + \varepsilon_{\text{extra}}, \gamma' = \gamma^2\varepsilon_{\text{extra}}$

Implications

- Provide the first rigorous explanation showing why a kernel is a good similarity function.
- Our algorithms do not require positive semidefinite functions!