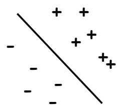
15-859(B) Machine Learning Theory

Lecture 9: Margins, kernels, and similarity functions

Avrim Blum
02/13/08

Basic Supervised learning setting

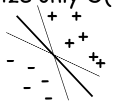
- Examples are points x in instance space, like R^n.
- Labeled + or -.
- Assume drawn from some probability distribution:
  - Distribution D over x, labeled by target function c.
  - Or distribution P over (x, l)
  - Will call P (or (c,D)) our "learning problem".
- Given labeled training data, want algorithm to do well on new data.



Margins

If data is separable by large margin γ, then that's a good thing. Need sample size only $\tilde{O}(1/\gamma^2)$.

$|w\cdot x|/|x| \geq \gamma,\ |w|=1$



Some ways to see it:

1. The perceptron algorithm does well: makes only $1/\gamma^2$ mistakes.
2. Margin bounds: whp all consistent large-margin separators have low true error.
3. Really-Simple-Learning + boosting…
4. Random projection…

A really simple learning algorithm

Suppose our problem has the property that whp a sufficiently large sample S would be separable by margin γ. Here is another way to see why this is good for learning.

Consider the following simple algorithm…

1. Pick a random hyperplane.
2. See if it is any good.
3. If it is a weak-learner (error rate $\leq \frac{1}{2} - \gamma/4$), plug into boosting. Else don't. Repeat.

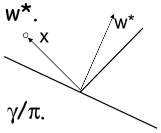
Claim: if data has a large margin separator, there's a reasonable chance a random hyperplane will be a weak-learner.

A really simple learning algorithm

Claim: if data has a separator of margin γ, there's a reasonable chance a random hyperplane will have error $\leq \frac{1}{2} - \gamma/4$. [all hyperplanes through origin]

Proof:

- Pick a (positive) example x. Consider the 2-d plane defined by x and target w*.
- $\Pr_h(h\cdot x \leq 0 \mid h\cdot w^* \geq 0)$
  $\leq (\pi/2 - \gamma)/\pi = \frac{1}{2} - \gamma/\pi.$
- So, $E_h[\mathrm{err}(h) \mid h\cdot w^* \geq 0] \leq \frac{1}{2} - \gamma/\pi.$
- Since err(h) is bounded between 0 and 1, there must be a reasonable chance of success.



QED

Another way to see why large margin is good

Johnson-Lindenstrauss Lemma:

Given n points in R^n, if project randomly to R^k, for $k = O(\varepsilon^{-2} \log n)$, then whp all pairwise distances preserved up to $1 \pm \varepsilon$ (after scaling by $(n/k)^{1/2}$).

Cleanest proofs: IM98, DG99

JL Lemma

Given n points in R^n, if project randomly to R^k, for $k = O(\varepsilon^{-2} \log n)$, then whp all pairwise distances preserved up to $1\pm\varepsilon$ (after scaling).

Cleanest proofs: IM98, DG99

Proof intuition:

- Consider a random unit-length vector $(x_1,x_2,\ldots,x_n)\in R^n$. What does x_1 coordinate look like?
- $E[x_1^2]=1/n$. Usually $\leq c/n$.
- If indep, $\Pr[|(x_1^2 + \ldots + x_k^2) - k/n| \geq \varepsilon k/n] \leq e^{-O(k\varepsilon^2)}$.
- So, at $k = O(\varepsilon^{-2} \log n)$, with prob $1 - 1/\text{poly}(n)$, projection to 1st k coordinates has length $(k/n)^{1/2} (1 \pm \varepsilon)$.
- Now, apply this to vector $v_{ij} = p_i - p_j$, projecting onto random k-diml space.

Whp **all** v_{ij} project to length $(k/n)^{1/2}(1\pm\varepsilon)|v_{ij}|$

JL Lemma, cont

Proof easiest for slightly different projection:

- Pick k vectors $u_1, \ldots, u_k$ iid from n-diml gaussian.
- Map $p \to (p \cdot u_1, \ldots, p \cdot u_k)$.
- What happens to $v_{ij} = p_i - p_j$?
  - Becomes $(v_{ij} \cdot u_1, \ldots , v_{ij} \cdot u_k)$
  - Each component is iid from 1-diml gaussian, scaled by $|v_{ij}|$.
  - For concentration on sum of squares, plug in version of Hoeffding for RVs that are squares of gaussians.
- So, whp all lengths apx preserved, and in fact not hard to see that whp all angles are apx preserved too.

Random projection and margins

Natural connection [AV99]:

- Suppose we have a set S of points in R^n, separable by margin γ.
- JL lemma says if project to random k-dimensional space for $k=O(\gamma^{-2} \log |S|)$, whp still separable (by margin $\gamma/2$).
  - Think of projecting points and target vector w.
  - Angles between p_i and w change by at most $\pm\gamma/2$.
- Could have picked projection before sampling data.
- So, it's really just a k-dimensional problem after all. Do all your learning in this k-diml space.

So, random projections can help us think about why margins are good for learning. [note: this argument does NOT imply uniform convergence in original space]

OK, now on to kernels…

Generic problem

- Given a set of images: , want to train a classifier to distinguish men from women.
- Problem: pixel representation not good for LTFs

Classic advice:

- Use a complicated neural net.
- But these are hard to train.

Modern advice:

- Use a Kernel! K(,) = Φ( )·Φ( ). Φ is implicit, high-dimensional mapping.
- Many algorithms only interact with data through dot-products, so can be "kernelized". If data is separable in Φ-space by large margin, don't have to pay for dim.

Generic problem

Modern advice:

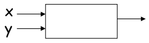
- Use a Kernel! K(,) = Φ( )· Φ( ). Φ is implicit, high-dimensional mapping.
- Many algorithms only interact with data through dot-products, so can be "kernelized". If data is separable in Φ-space by large margin, don't have to pay for dim.
- E.g., $K(x,y) = (1+x_1y_1)(1+x_2y_2)\ldots(1+x_ny_n)$.
  - Φ:(n-diml space) → (2^n-diml space).
- Conceptual warning: You're not really "getting all the power of the high dimensional space without paying for it". (Not enough to just be separable. Need large margin too.) As we saw from JL lemma, assumption of large margin means it's really an $\tilde{O}(1/\gamma^2)$-dimensional problem after all.

Question: do we need the notion of an implicit space to understand what makes a kernel helpful for learning?

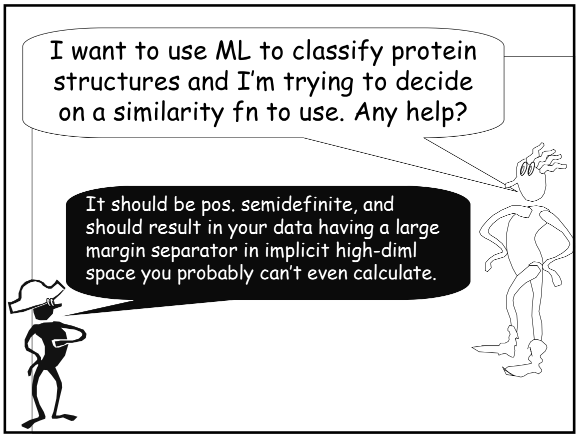
Kernel fns have become very popular

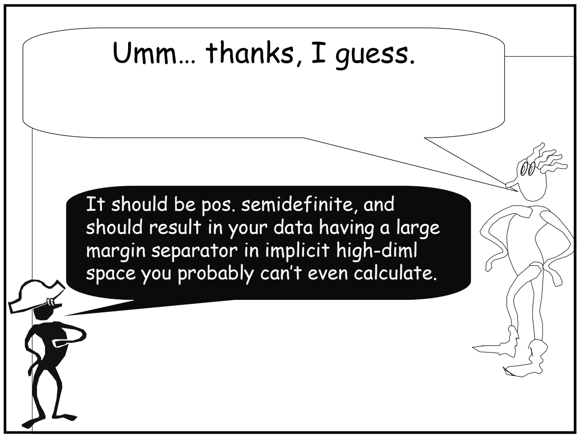
…but there's something a little funny:

- On the one hand, operationally a kernel is just a similarity function: $K(x,y) \in [-1,1]$, with some extra requirements. [here I'm scaling to $|\Phi(x)| = 1$]



- And in practice, people think of a good kernel as a good measure of similarity between data points.
- But Theory talks about margins in implicit high-dimensional Φ-space. $K(x,y) = \Phi(x)\cdot\Phi(y)$.

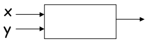

I want to use ML to classify protein structures and I'm trying to decide on a similarity fn to use. Any help?
It should be pos. semidefinite, and should result in your data having a large margin separator in implicit high-diml space you probably can't even calculate.


Umm… thanks, I guess.
It should be pos. semidefinite, and should result in your data having a large margin separator in implicit high-diml space you probably can't even calculate.

Kernel fns have become very popular

…but there's something a little funny:

- On the one hand, operationally a kernel is just a similarity function: $K(x,y) \in [-1,1]$, with some extra requirements. [here I'm scaling to $|\Phi(x)| = 1$]



- And in practice, people think of a good kernel as a good measure of similarity between data points.
- But Theory talks about margins in implicit high-dimensional Φ-space. $K(x,y) = \Phi(x)\cdot\Phi(y)$.

Can we bring these views together?

Goal: notion of "good similarity function" that…

1. Talks in terms of more intuitive properties (no implicit high-diml spaces…)
2. If K satisfies these properties for our given problem, then has implications to learning
3. Is broad: includes usual notion of "good kernel" (one that induces a large margin separator in Φ-space).

[Recent work with Nina Balcan, with extensions by Nati Srebro]

Defn satisfying (1) and (2):

- Say have a learning problem P (distribution D over examples labeled by unknown target f).
- Sim fn $K:(x,y)\rightarrow[-1,1]$ is (ε,γ)-good for P if at least a $1-\varepsilon$ fraction of examples x satisfy:

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

- Note: you can have this property without being a legal kernel.
- Q: how could you use this to learn?

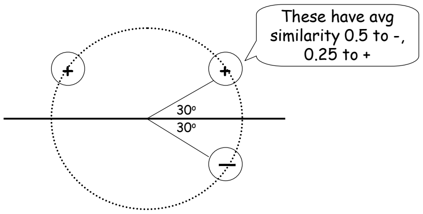
How to use it

At least a $1-\varepsilon$ prob mass of x satisfy:

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

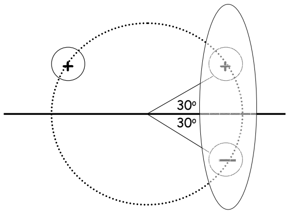
- Draw S^+ of $O((1/\gamma^2)\ln 1/\delta^2)$ positive examples.
- Draw S^- of $O((1/\gamma^2)\ln 1/\delta^2)$ negative examples.
- Classify x based on which gives better score.
  - Hoeffding: for any given "good x", prob of error over draw of S^+,S^- at most δ^2.
  - So, at most δ chance our draw is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

But not broad enough



- $K(x,y)=x\cdot y$ has good separator but doesn't satisfy defn. (half of positives are more similar to negs that to typical pos)

But not broad enough



- Idea: would work if we didn't pick y's from top-left.
- Broaden to say: OK if ∃ 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. (even if don't know R in advance)

Broader defn…

- Ask that exists a set R of "reasonable" y (allow probabilistic) s.t. almost all x satisfy

$$E_y[K(x,y)|\ell(y)=\ell(x),R(y)] \geq E_y[K(x,y)|\ell(y)\neq\ell(x), R(y)]+\gamma$$

- And at least ε probability mass of reasonable positives/negatives.
- But now, how can we use for learning??

Broader defn…

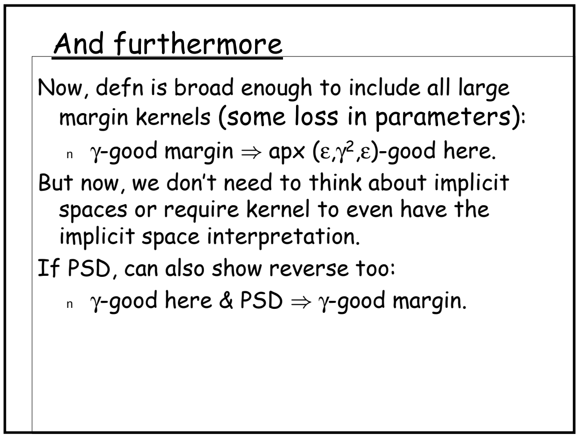
- Ask that exists a set R of "reasonable" y (allow probabilistic) s.t. almost all x satisfy

$$E_y[K(x,y)|\ell(y)=\ell(x),R(y)] \geq E_y[K(x,y)|\ell(y)\neq\ell(x), R(y)]+\gamma$$

  - Draw $S = \{y_1,\dots,y_n\}$, $n\approx1/(\gamma^2\varepsilon)$.

    could be unlabeled
  - View as "landmarks", use to map new data:

    $F(x) = [K(x,y_1), \dots,K(x,y_n)]$.
  - Whp, exists separator of good L_1 margin in this space: $w=[0,0,1/n_+,1/n_+,0,0,0,-1/n_-,0,0]$
  - n_+ $[n_-]$ = # reasonable pos [neg] in S.
  - So, take new set of examples, project to this space, and run good L_1 alg (Winnow).



And furthermore

Now, defn is broad enough to include all large margin kernels (some loss in parameters):

- γ-good margin ⇒ apx $(\varepsilon,\gamma^2,\varepsilon)$-good here.

But now, we don't need to think about implicit spaces or require kernel to even have the implicit space interpretation.

If PSD, can also show reverse too:

- γ-good here & PSD ⇒ γ-good margin.