

CSCI 5622 Machine Learning

ML Computational Learning Theory

DATE READ	DUE	
Today, Oct 7	7	Notes Papers 3&4
Mon, Oct 12	Bagging & Boosting	Exper. 1 plan (1 pg)
Wed, Oct 14	Random Forests	Literature Review

www.RodneyNielsen.com/teaching/CSCI5622-F09/

Instructor: Rodney Nielsen

Assistant Professor Adjunct, CU Dept. of Computer Science

Research Assistant Professor, DU, Dept. of Electrical & Computer Engr.

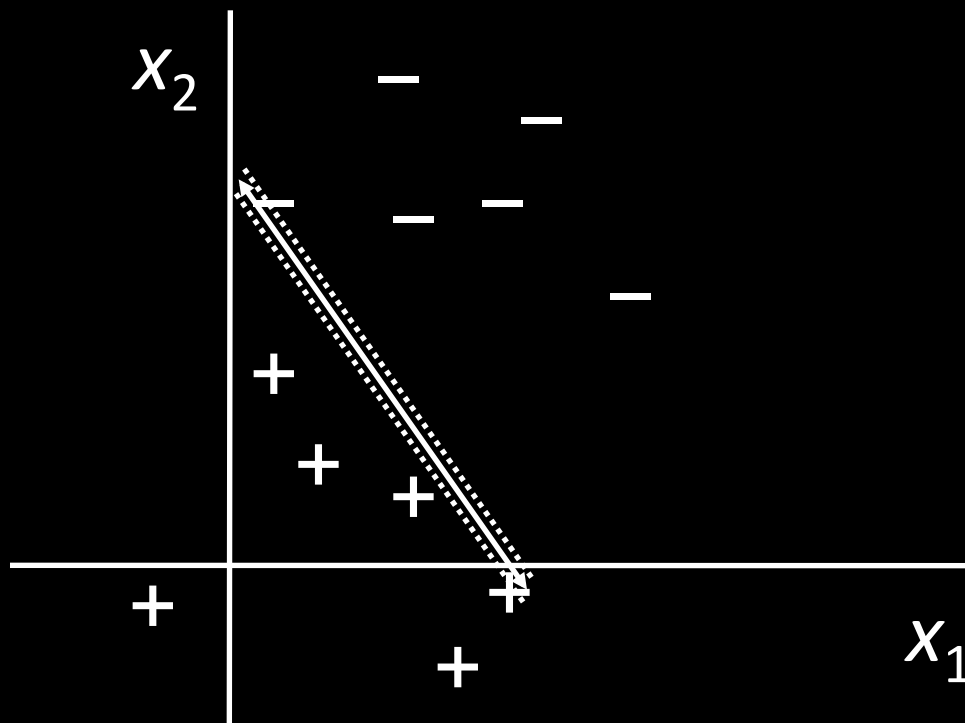
Research Scientist, Boulder Language Technologies

ML Support Vector Machines (SVMs)

- **Generalization usually as good and often significantly better than other methods**

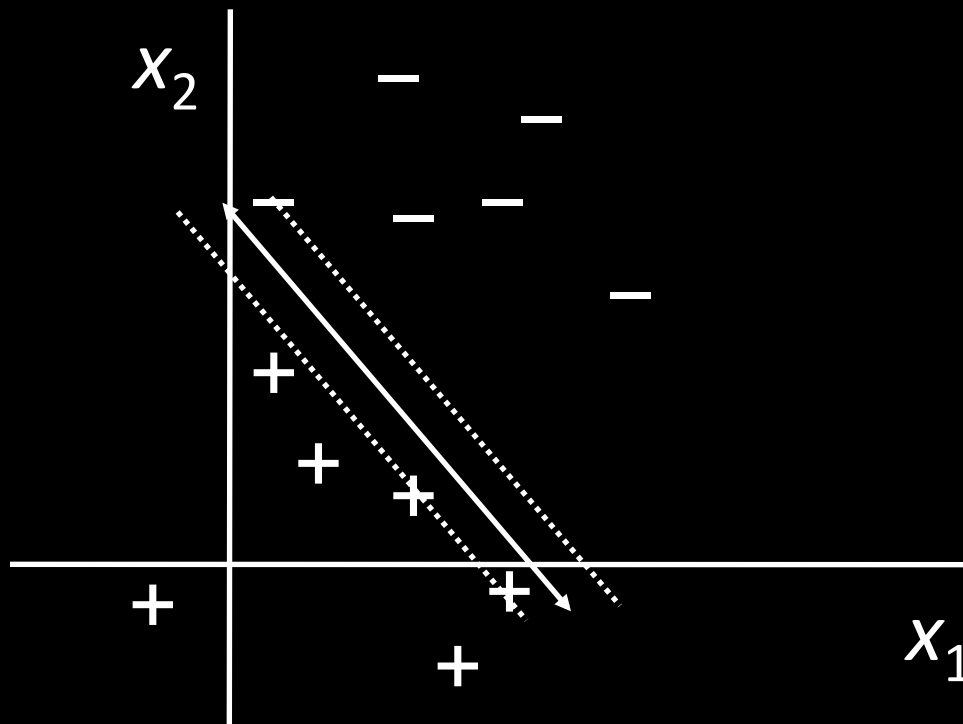
ML Linear Support Vector Machines

- Search for hyperplane that maximizes the margin ($d_+ + d_-$)



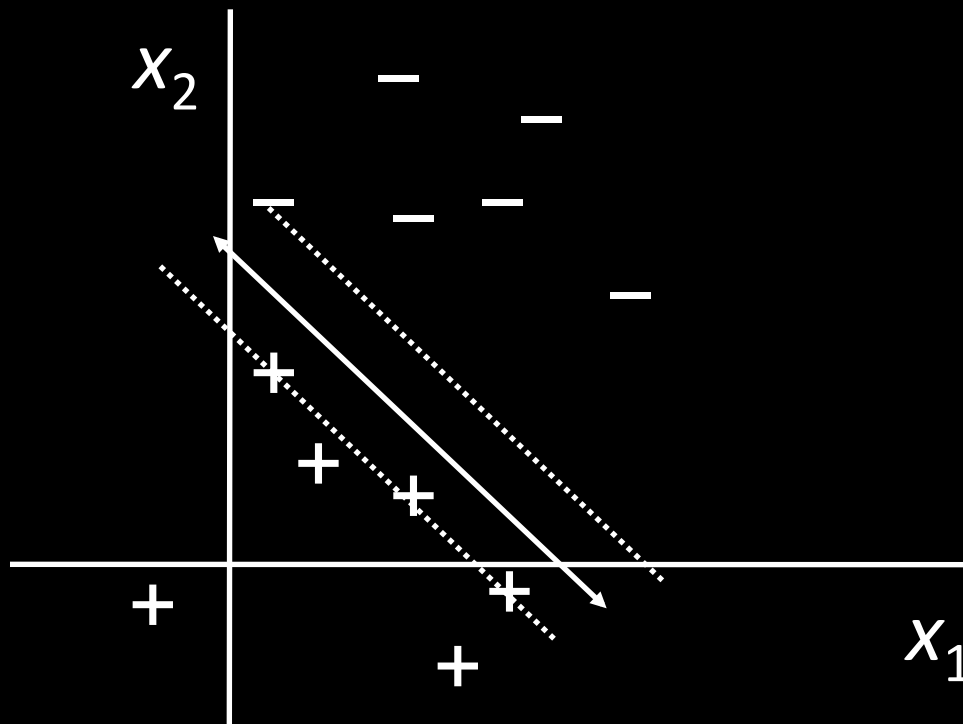
ML Linear Support Vector Machines

- Search for hyperplane that maximizes the margin ($d_+ + d_-$)



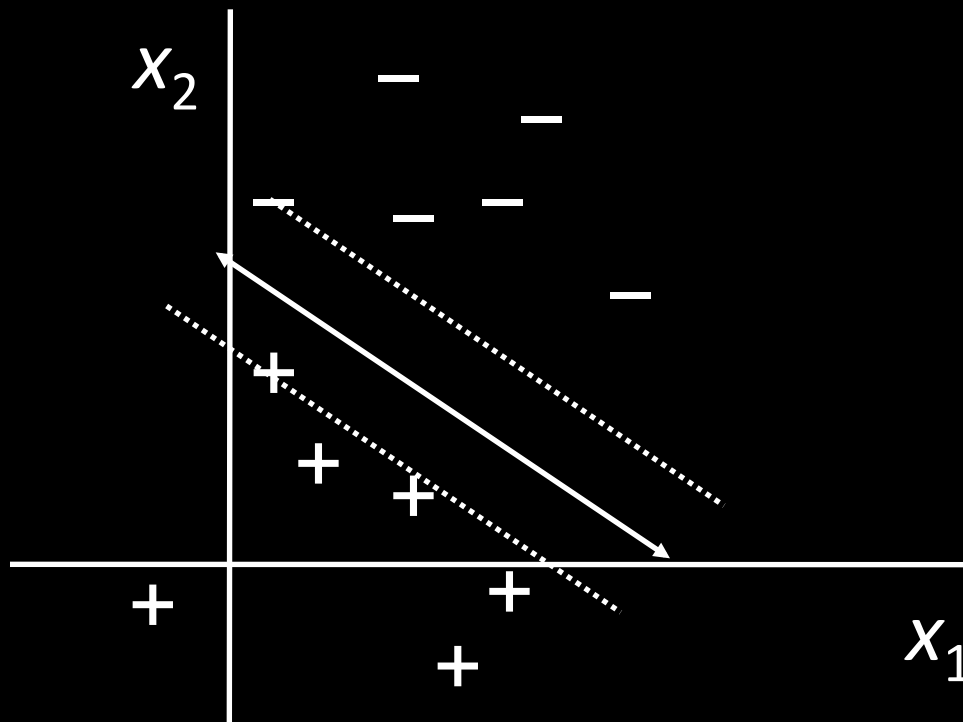
ML Linear Support Vector Machines

- Search for hyperplane that maximizes the margin ($d_+ + d_-$)



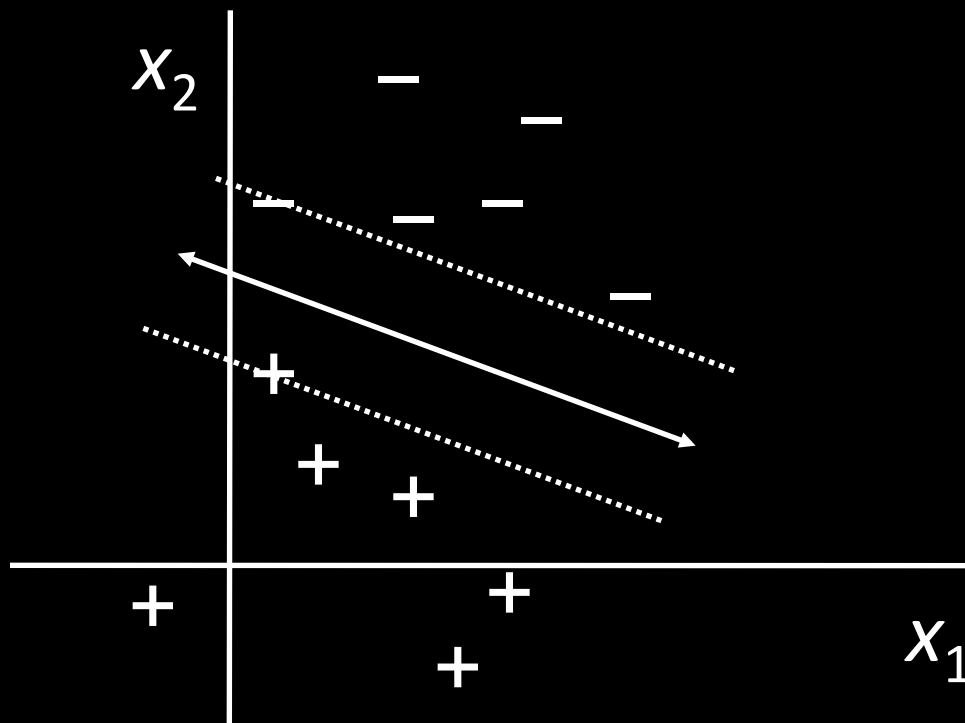
ML Linear Support Vector Machines

- Search for hyperplane that maximizes the margin ($d_+ + d_-$)



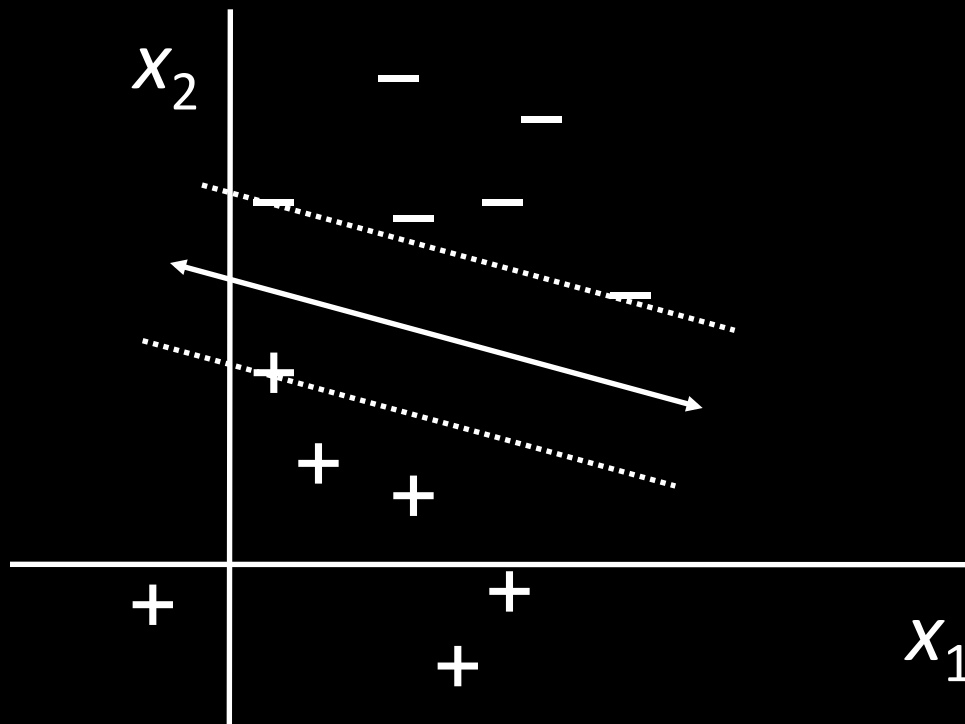
ML Linear Support Vector Machines

- Search for hyperplane that maximizes the margin ($d_+ + d_-$)



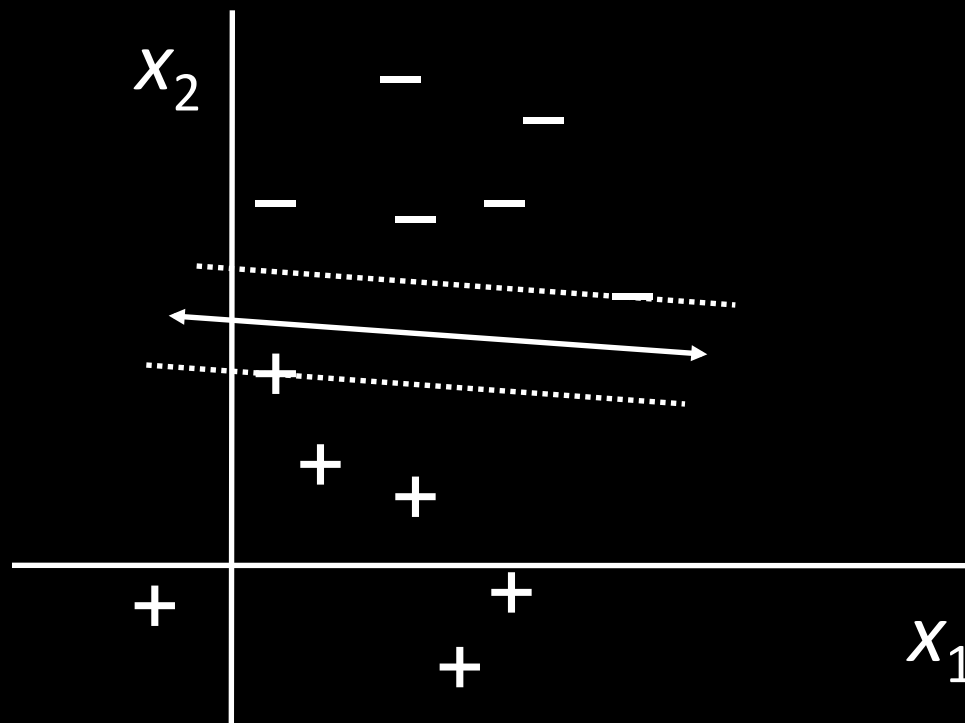
ML Linear Support Vector Machines

- Search for hyperplane that maximizes the margin ($d_+ + d_-$)



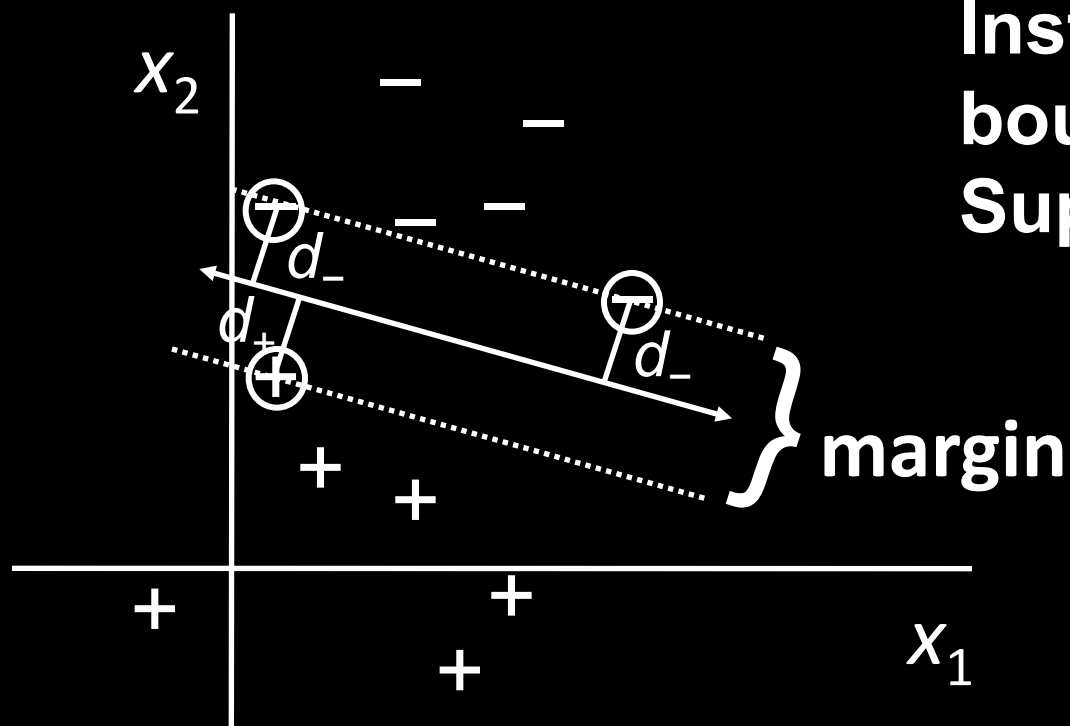
ML Linear Support Vector Machines

- Search for hyperplane that maximizes the margin ($d_+ + d_-$)



ML Linear Support Vector Machines

- Search for hyperplane that maximizes the margin ($d_+ + d_-$)



Instances on the boundaries are the Support Vectors

- Kernels are an effective method of projection

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1..N} \alpha_i y^{(i)} \mathbf{x}^{(i)T} \mathbf{x} + b\right)$$

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1..N} \alpha_i y^{(i)} \phi(\mathbf{x}^{(i)})^T \phi(\mathbf{x}) + b\right)$$

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1..N} \alpha_i y^{(i)} K(\mathbf{x}^{(i)}, \mathbf{x}) + b\right)$$

ML Most common Kernels

- Polynomial Kernels

$$K(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{xz})^d$$

- Radial Basis Functions

$$K(\mathbf{x}, \mathbf{z}) = \exp \frac{-(\mathbf{x} - \mathbf{z})^2}{2\sigma^2}$$

- **Provide the benefits of working in higher dimensional space**
- **Avoid the computational problems of working in higher dimensional space**
- **Avoid the theoretical curse of dimensionality problems of working in higher dimensional space**

ML **SVM Key Properties**

- **Duality**
- **Kernels**
- **Margin**
- **Convexity**
- **Sparseness**

ML Computational Learning Theory

- When can and can't you successfully learn a function
- When can/can't a specific learner be successful
- PAC Learning (Probably Approximately Correct)
 - What classes of hypotheses can be learned from a polynomial number of examples
 - Define natural measure of complexity for H that allows bounding # of required training instances
- Mistake bound framework
 - # of training errors until correct h is found

ML PAC Learning Model

- **Probably Approximately Correct Learning Model**
- **The case of learning Boolean-valued functions from noise free data**
 - **But many of the results can be extended**
- **\mathcal{D} = distribution**
- **x = instance**
- **y = class**
- **$f: X \rightarrow \{0, 1\}$**
- **L = learner**
- **H = hypothesis space**

- Evaluate L based on performance of h over new instances x drawn randomly according to \mathcal{D}
- Characterize performance of different L using different hypothesis spaces H , when learning different target concepts $f(x) = y$
- Worst case analysis over all f and \mathcal{D}

Error of Hypothesis

- Interested in true error of approximation h relative to the target function f
- True error = $error_{\mathcal{D}}(h) = \Pr_{x \in \mathcal{D}} [f(x) \neq h(x)]$
- Training error = $error_{D_{Tr}}(h) = \Pr_{x \in D_{Tr}} [f(x) \neq h(x)]$
- How probable is it that the observed $error_{D_{Tr}}(h)$ is a misleading estimate of $error_{\mathcal{D}}(h)$?

- Describe types of concepts you can reliably learn from reasonable amount of data and computation
- Can we find N such that the *true error* _{\mathcal{D}} (h) = 0
- Can we find N such that the *true error* _{\mathcal{D}} (h) < ϵ
- Can we find N such that there is only a δ chance of failing to learn with *error* _{\mathcal{D}} (h) < ϵ
- Learner *probably* learns h that is *approximately correct*

PAC Learnability

- The concept class F is said to be PAC learnable by L , which uses H , if:
 - For all f in F , \mathcal{D} over X , $0 < \epsilon < 0.5$, $0 < \delta < 0.5$,
 - There is a $(1-\delta)$ chance that L will output an h such that $error_{\mathcal{D}}(h) \leq \epsilon$
 - In time that is polynomial in $1/\epsilon$, $1/\delta$, $|X|$, and $encoding\ size(f)$

ML **Sample Complexity**

- **Sample Complexity: Growth in the number of required training examples with the problem size**
- **Consistent Learners: output hypotheses that perfectly fit the training data whenever possible**
- **Assuming no noise, it is reasonable to prefer a hypothesis that fits the training data**

- Can we bound the size of the training set required by any *consistent* learner?
- $N \geq (1/\varepsilon)(\ln |H| + \ln(1/\delta))$
- Given N training instances, a consistent learner will *probably*, with $p = (1-\delta)$, successfully learn to *approximate* the *correct* function f , with error less than ε

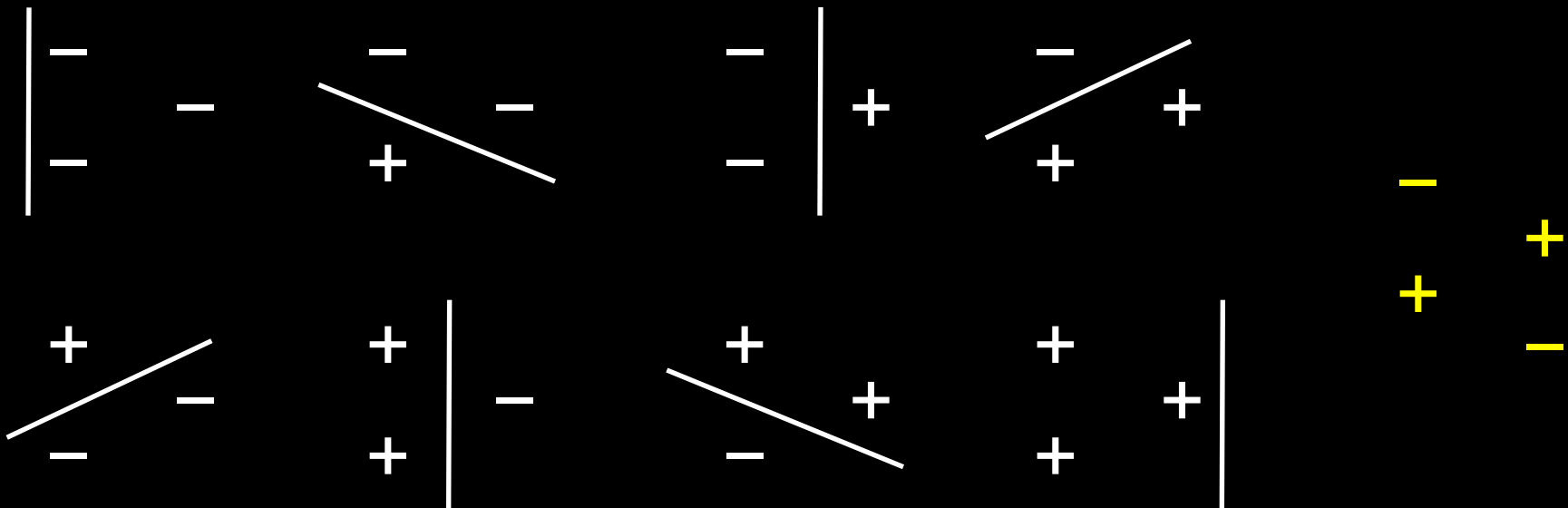
- Can we bound the size of the training set required by *any* learner (e.g., inconsistent and agnostic learners)?
- $N \geq (1/2\varepsilon^2)(\ln |H| + \ln(1/\delta))$
- Given N training instances, any learner will *probably*, with $p = (1-\delta)$, successfully learn to *approximate* the *correct* function f , with error less than ε

ML PAC Learnability

- Unbiased Learners
- $N \geq (1/\epsilon)(2^n \ln 2 + \ln(1/\delta))$

VC Dimension

- If for some set of N points & every possible labeling of that set, some $f(\alpha)$ in $\{f(\alpha)\}$ can correctly label the data, then the set of points is *shattered* by $\{f(\alpha)\}$
- Assume $\{f(\alpha)\}$ is the set of oriented lines and x in \mathbb{R}^2 , then $h=3$



ML Risk (or Generalization) Bounds

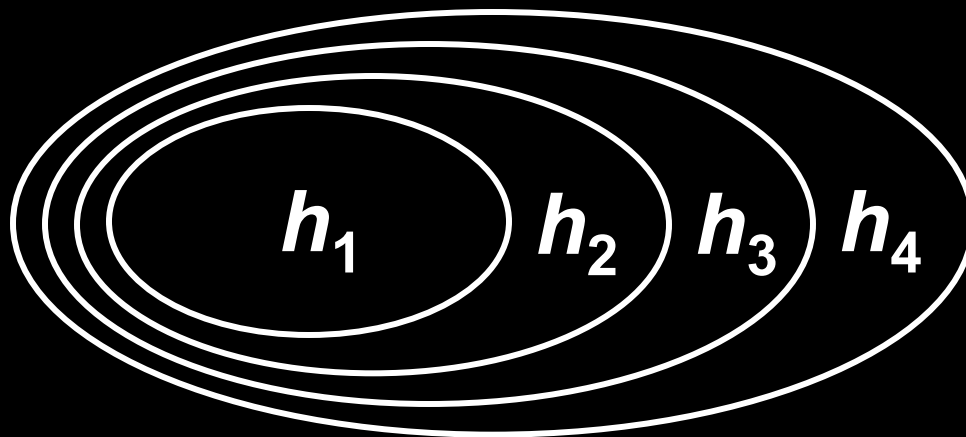
- Risk bound with probability $(1 - \eta)$:

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(1 + \log 2N/h) - \log \eta/4}{N}}$$

- $h \geq 0$, Vapnik Chervonenkis (VC) dimension
- h is a measure of capacity of learner
- Second term on right is the VC confidence
- Independent of $P(x,y)$, but must be i.i.d.
- Can't compute $R(\alpha)$
- If you know h , easy to compute risk bound

ML Structural Risk Minimization

- Find h that minimizes the actual risk
- Train a learner for each subset
- Choose min of empirical risk + VC confidence



ML Proj Discussion: Cross Validation

- When is CV okay?

ML Proj Discussion: Eval Comparisons

- **Baselines**
 - Random
 - Majority class (most frequent y in training data)
 - One or more obvious good features
 - Median from a challenge
- **Other**
 - Best prior results of similar algorithm if justified
 - Best prior results
 - Prior experiment results with your algorithm

ML **Statistical Testing**

- **Algorithm comparisons**
- **Annotation**
 - **Kappa**
 - **Double annotation**
 - **Adjudication**

ML **Final Term Paper**

- **Term paper is due to peer group by Wed Dec 2**
- **Feedback to peers is due by Mon Dec 7**
- **Term paper is due to me by Fri Dec 11**
- **Peer presentations by Tue Dec 15**
- **Class presentations on Wed Dec 16**

ML Presentations Next Week

- **Volunteers?**

ML

Fourth Review

- Yan
- David

ML FLAIRS-23: Intl AI Researchers Soc.

- <http://www.flairs-23.info/> deadline: Nov 23
- **Several special tracks**
 - Data Mining
 - Applied Natural Language Processing
 - Book chapter
 - AI, Cognitive Sem & Comp Ling: new perspectives
 - Games and Entertainment
 - AI Planning and Scheduling
 - Learning in Intelligent Systems
 - Spatio-Temporal Reasoning
 - Uncertain Reasoning