Does Unlabeled Data Help?

Worst-case Analysis of the
Sample Complexity of Semi-supervised Learning.

Ben-David, Lu and Pal; COLT, 2008.

Presentation by Ashish Rastogi
Courant Machine Learning Seminar.

Outline

- Introduction & Motivation
- SSL in Practice (SSL assumptions)
- Preliminaries
- Conjecture about Sample Complexity
- Proof of Conjecture for a Simple Concept Class in Realizable Setting
- Proof of Conjecture for another Simple Concept Class in Agnostic Setting
- Conclusion

Introduction

Setting	Input to the algorithm (Distribution *D*)	Algorithm output	Performance of the algorithm	Examples
Supervised Learning	labeled examples	a function that maps points to labels	expected error on an unseen point	Support Vector Machines, AdaBoost
Unsupervised Learning	unlabeled examples	a function that maps points to labels	expected error on an unseen point	Clustering
Semi-supervised Learning (SSL)	labeled & unlabeled examples	a function that maps points to labels	expected error on an unseen point	
Transductive Learning	labeled & unlabeled examples	labels of the unlabeled examples	average error on unlabeled examples	Transductive SVMs, Graph regularization

Motivation

- In real world problems,
  - much more unlabeled data than labeled data.
  - labeling costly, both in terms of time and money.
- Examples: web document classifiers, speech recognition, protein sequences, ...

- Can unlabeled data help prediction accuracy?
  - intuitively, if labels continuous, expect better performance.
  - but, a precise theoretical analysis of the merits of SSL missing.

Introduction

- Question: Can SSL help *without* any prior assumptions on the distribution of labels?
- Measure “help” in terms of reduced labeled sample complexity.
- Sample complexity: how much training data needed to learn effectively.
- Answer: For some simple hypotheses space, not significantly,
  - at most a factor of 2 reduction in the sample complexity.

SSL in Practice

- Binary classification problem.
- Access to labeled training data.

SSL in Practice

- Binary classification problem.
- Access to labeled training data.

- More confidence with unlabeled data if labels "continuous".
- But, need an assumption on the distribution of labels of .

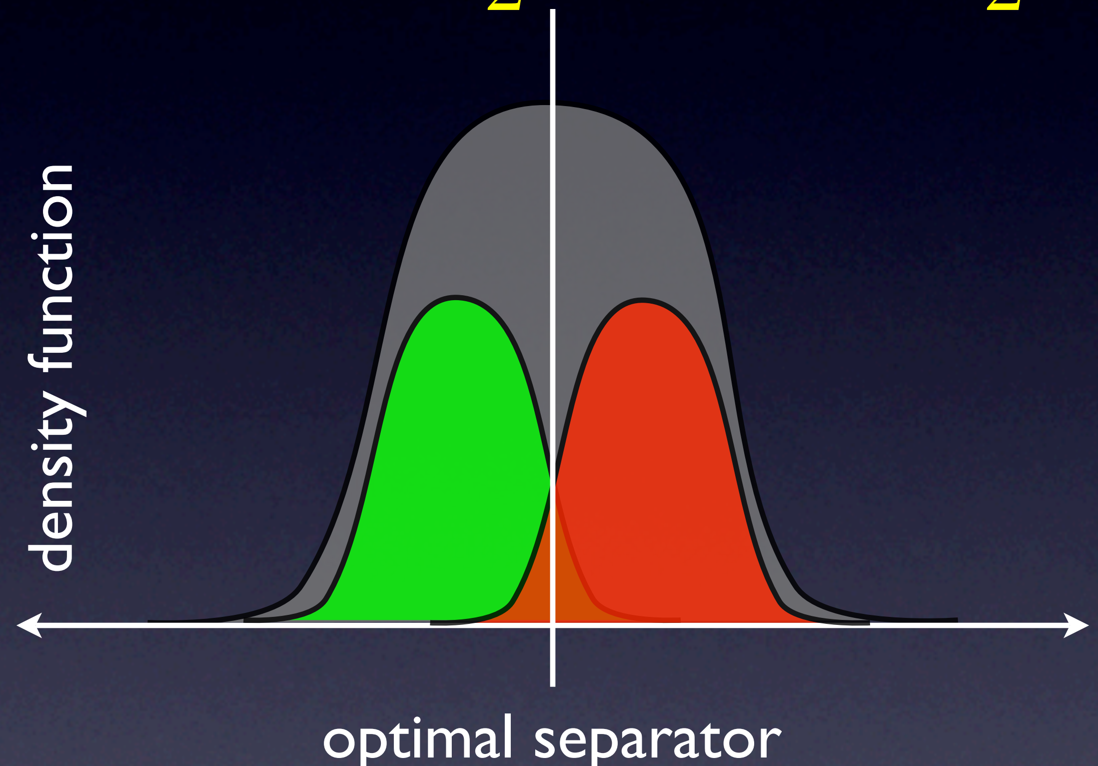
SSL in Practice

- Common assumptions when using semi-supervised learning:
  - decision boundary should lie in a low-density region.
  - points in the same cluster likely to be of the same class.
  - if two points in a high-density region are close, then so should be the corresponding outputs.
- Difficulty: assumptions hard to verify and not formalizable.
- No analysis of precise conditions under which SSL is better.
- There are bounds for classification and regression under SSL,
  - but none shown to be provably better than the supervised setting.

[Vapnik, 98; El-Yaniv & Pechyony, 06; Cortes et. al, 08]

SSL in Practice

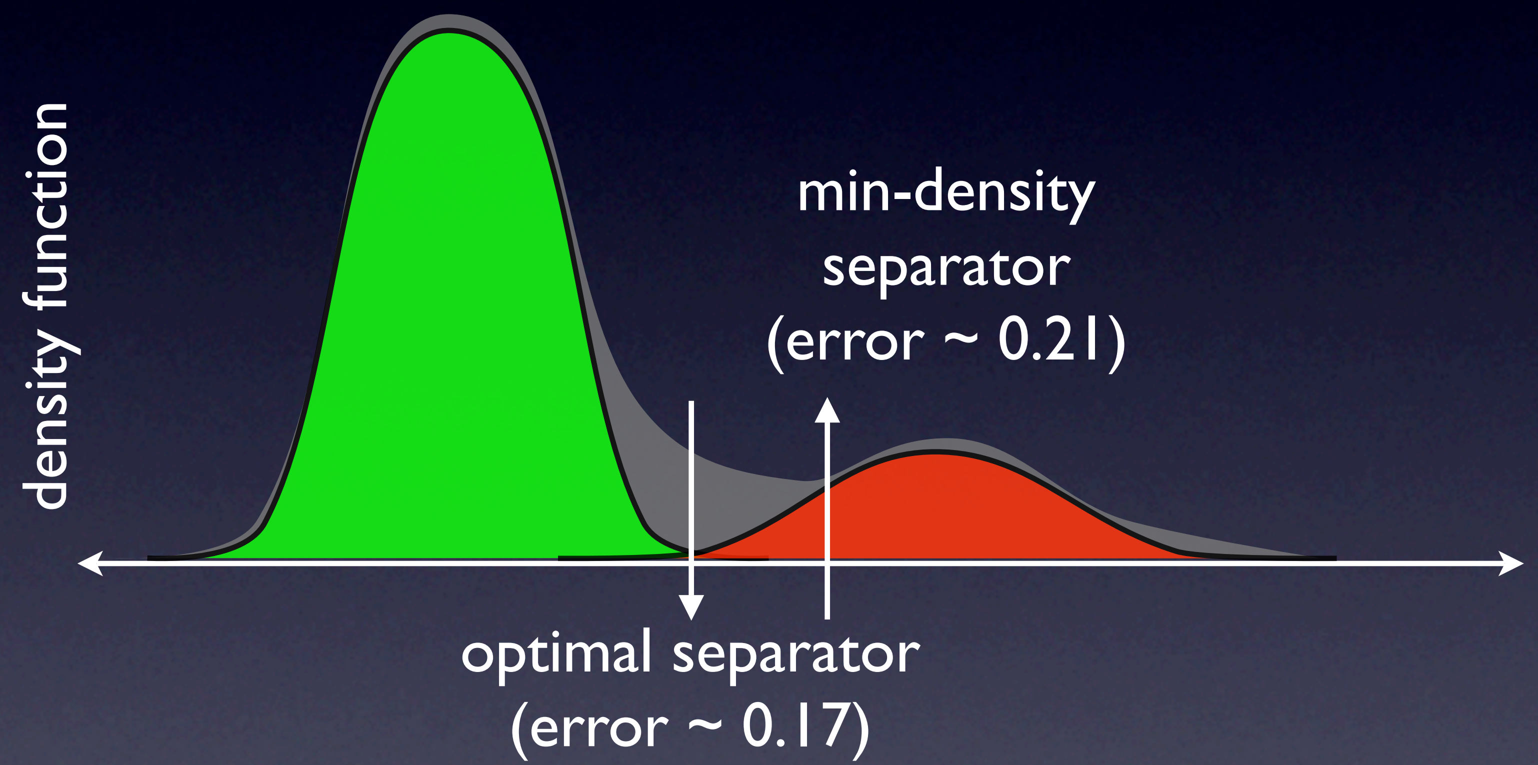
- Low-density separators not always good,
  - two overlapping gaussians; $\frac{1}{2}N(-1,1) + \frac{1}{2}N(1,1)$



  - optimal separator in a very high-density region.

SSL in Practice

- Low-density separators can be quite bad!
  - two gaussians with different variances: $\frac{1}{2}N(-2,1) + \frac{1}{2}N(2,2)$



Preliminaries

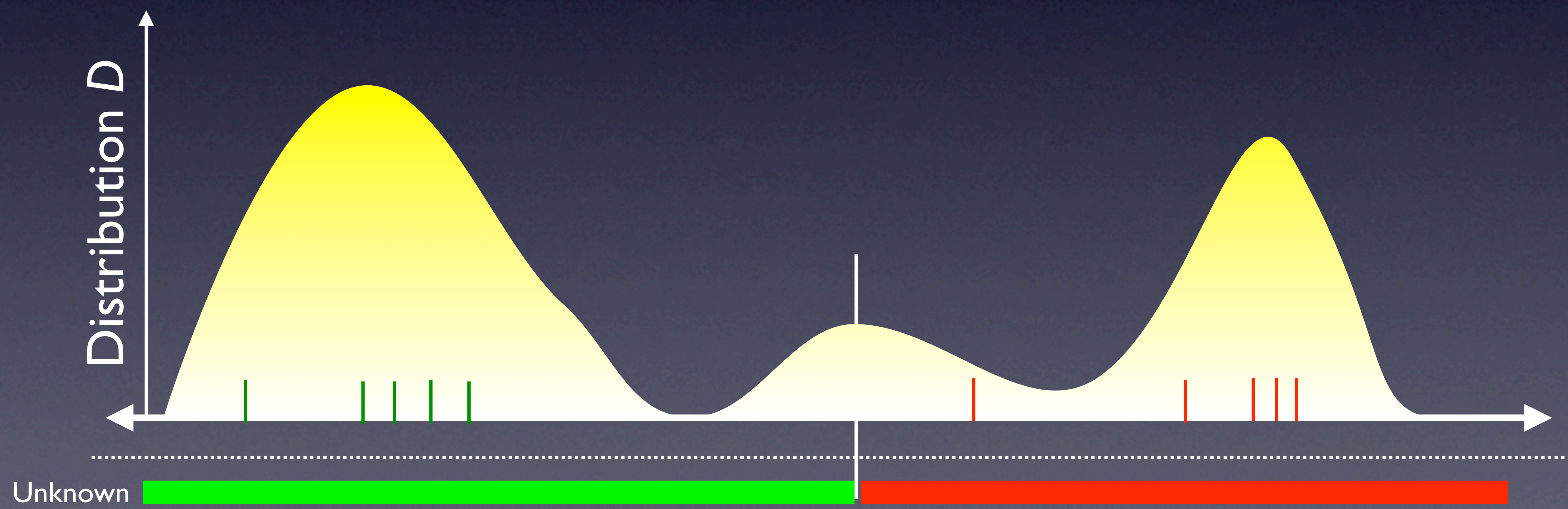
- Let X be a domain set.
- We focus on classification, so labels are $Y = \{0, 1\}$.
- Hypotheses are functions, $h : X \mapsto Y$. Hypothesis set H.
- Target function: a probability distribution over $X \times Y$.
- For this talk, assume Y a deterministic function of X, i.e:

$$\Pr[y = 1|x], \Pr[y = 0|x] \in \{0, 1\}.$$

- Realizable setting: target function $f \in H$, minimum error 0.
  [also called the consistent case]
- Agnostic setting: target function $f \notin H$, minimum error not 0.
  [also called the inconsistent case]

Preliminaries

- In the SSL setting considered, the learning algorithm receives:
  - a labeled training sample $S = \{(x_i, y_i)_{i=1}^{m}\}$.
  - the entire unlabeled distribution D.

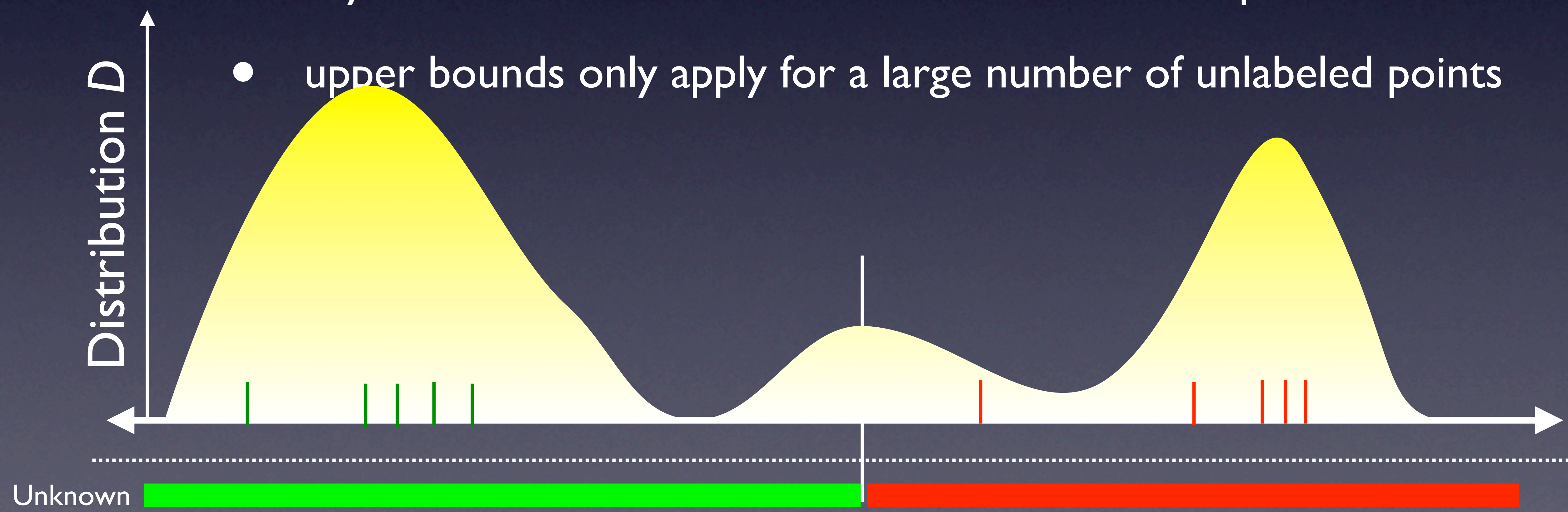


Preliminaries

- A supervised learning algorithm $A(S)$:
  - receives as input m labeled examples, $(x_i, y_i)_{i=1}^m$.
  - produces a hypothesis $h : X \mapsto Y$.
- An unsupervised learning algorithm $A(S, D)$:
  - receives as input m labeled examples, $(x_i, y_i)_{i=1}^m$ and a probability distribution D over X.
  - produces a hypothesis $h : X \mapsto Y$.
- Risk (test error): $\mathrm{Err}^D(h) = \Pr_{x \sim D}[h(x) \neq y_x]$
- Training error (empirical error): $\mathrm{Err}^S(h) = \dfrac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{h(x_i) \neq y_i}$

Preliminaries

- In the SSL setting considered, the learning algorithm receives:
  - a labeled training sample $S = \{(x_i, y_i)_{i=1}^m\}$
  - the entire unlabeled distribution D
    - really, this is the transductive setting (infinite unlabeled points)
    - any lower bound carries over to fewer unlabeled points
    - upper bounds only apply for a large number of unlabeled points



Sample Complexity

- For a class H of hypotheses, the sample complexity of an SSL algorithm $A(S, D)$, confidence $1-\delta$, accuracy ϵ is

$$m(A(\cdot, D), H, \epsilon, \delta) \quad = \quad \min \{ m \in \mathbb{N} \ : \ \Pr_{S \sim D^m} \left[\mathrm{Err}^D(A(S, D)) - \inf_{h \in H} \mathrm{Err}^D(h) > \epsilon \right] < \delta \}$$

- In words, the number of samples needed for the error of the learning algorithm to be within ϵ of the optimal hypothesis with a probability at least $1-\delta$.
- In the realizable case: $\inf_{h \in H} \mathrm{Err}^D(h) = 0$
- Symmetric difference of $h_1, h_2 \in H$:

$$h_1 \Delta h_2 = \mathbf{1}_{\{x \in X \ : \ h_1(x) \neq h_2(x)\}}$$

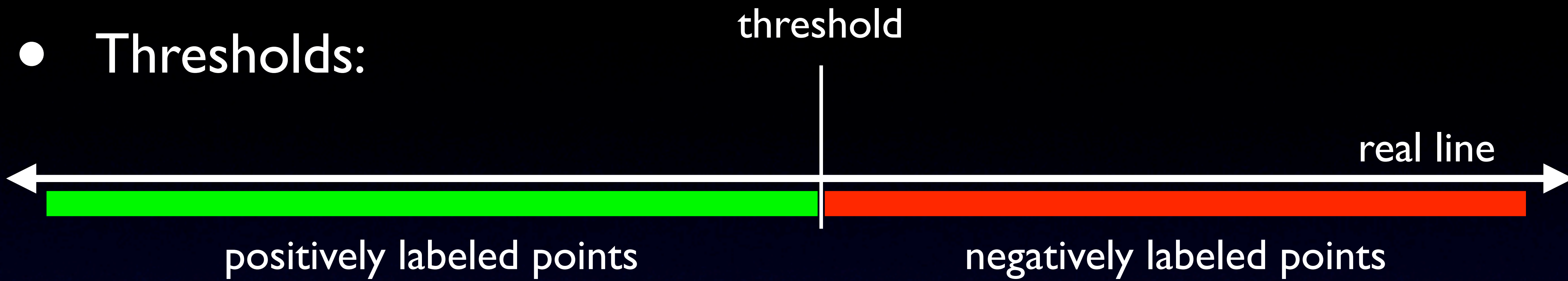
Conjectures

- For any hypothesis class H, there exists a constant $c \geq 1$ there exists a single supervised algorithm A, such that for any dist. D and any SSL algorithm B, the sample complexity of the supervised algorithm is larger by at most a factor of c.

$$\sup_{f \in H} m(A(\cdot), H, \epsilon, \delta) \leq c \cdot \sup_{f \in H} m(B(\cdot, D), H, \epsilon, \delta)$$

Two Hypotheses Classes

- Thresholds:



$$H = \{\mathbf{1}_{(-\infty,t]} : t \in \mathbb{R}\}$$

- Union of d intervals:

$$\mathrm{UI}_d = \{\mathbf{1}_{[a_1,a_2)\cup\ldots\cup[a_{2l-1},a_{2l}):l\leq d}\}$$

- FYI, recall that:
  - VC(thresholds) = 1, VC(d intervals) = $2d$

Learning Thresholds

- Focus on the realizable case.
- Will show that the sample complexity of SSL for learning thresholds is half the sample complexity of supervised learning.
- Easy part: sample complexity of supervised learning.
- Theorem: let H be the hypothesis space and L the learning algorithm that returns the left-most hypothesis that is consistent with the training set. Then for any distribution D, for any $\epsilon, \delta > 0$ and any target $h \in H$,

$$m(A(\cdot), H, D, \epsilon, \delta) \leq \frac{\ln(1/\delta)}{\epsilon}$$

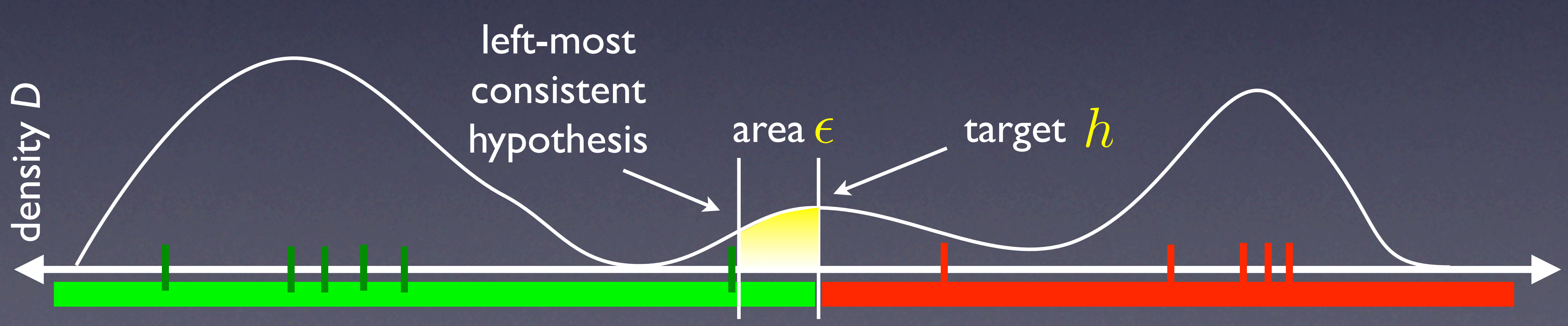
Learning Thresholds

- Theorem: let H be the hypotheses space and L the learning algorithm that returns the left-most hypothesis that is consistent with the training set. Then for any distribution D, for any $\epsilon, \delta > 0$ and any target $h \in H$,

$$m(A(\cdot), H, D, \epsilon, \delta) \leq \frac{\ln(1/\delta)}{\epsilon}$$

- Failure probability the same as no point lies in the shaded area.

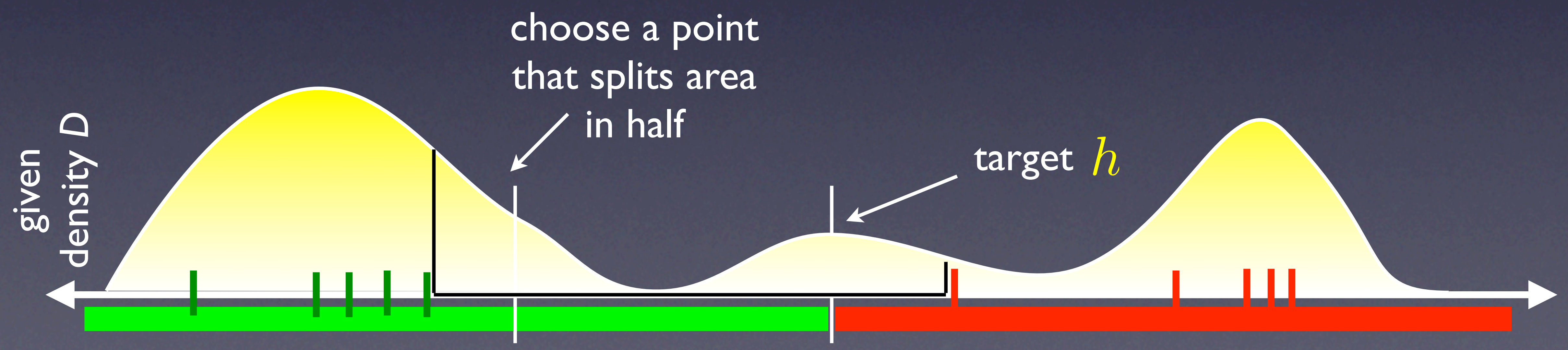
$$(1-\epsilon)^m \leq \exp(-\epsilon m) \leq \delta$$



Learning Thresholds

- An SSL algorithm for learning thresholds: [explain via picture] [Kääriäinen, 05]
- Theorem: Let H be a hypothesis space and B the above SSL learning algorithm. For any continuous distribution D, any ϵ, δ and any target $h \in H$,

$$m(B(\cdot, D), H, \epsilon, \delta) \leq \frac{\ln(1/\delta)}{2\epsilon} + \frac{ln2}{2\epsilon}$$

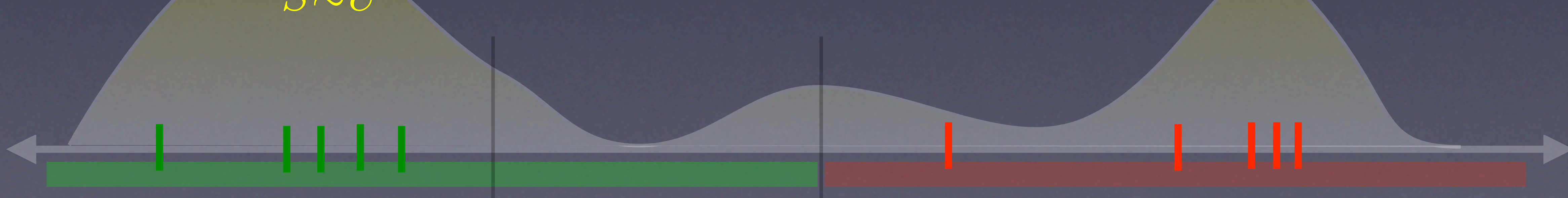


Learning Thresholds

- **Theorem**: Let H be a hypothesis space and B the above SSL learning algorithm. For any continuous distribution D, any ϵ, δ and any target $h \in H$,

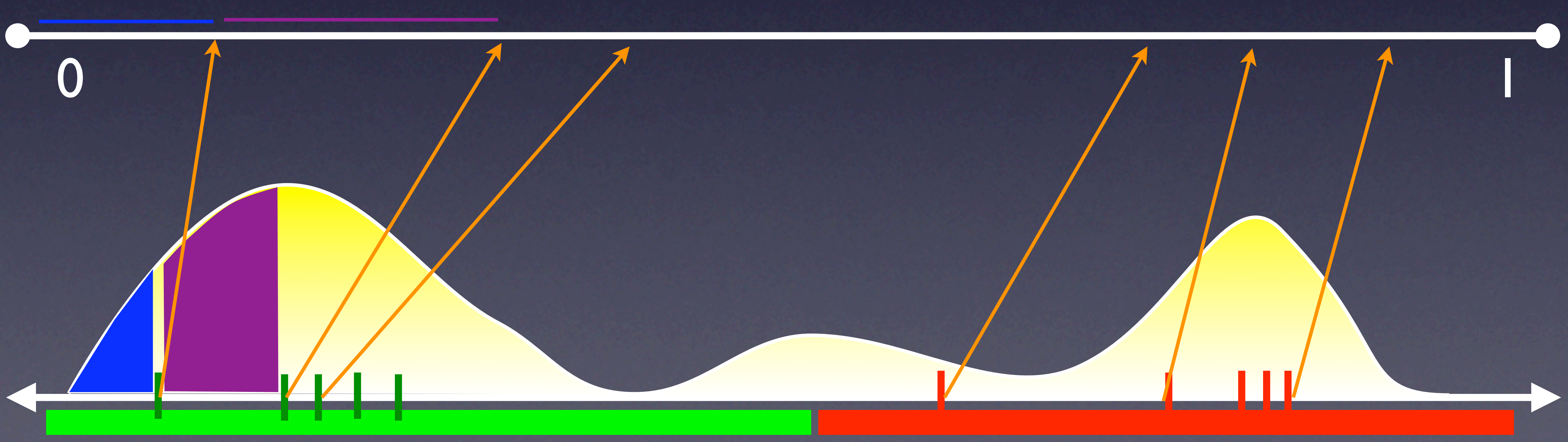
$$m(B(\cdot, D), H, \epsilon, \delta) \leq \frac{\ln(1/\delta)}{2\epsilon} + \frac{ln2}{2\epsilon}$$

- **Proof**: Let I be the open interval containing D
  - Step 1: Map I to $(0,1)$ via a function $F(\cdot)$. Transform training sample $S \mapsto S'$ via F
  - Step 2: $m(B(\cdot, D), H, \epsilon, \delta) \leq m(A(\cdot, U_{(0,1)}), H, \epsilon, \delta)$
  - Step 3: $\Pr_{S \sim U^m}\left[\mathrm{Err}^U(A(S', U)) \geq \epsilon\right] \leq 2(1-2\epsilon)^m$



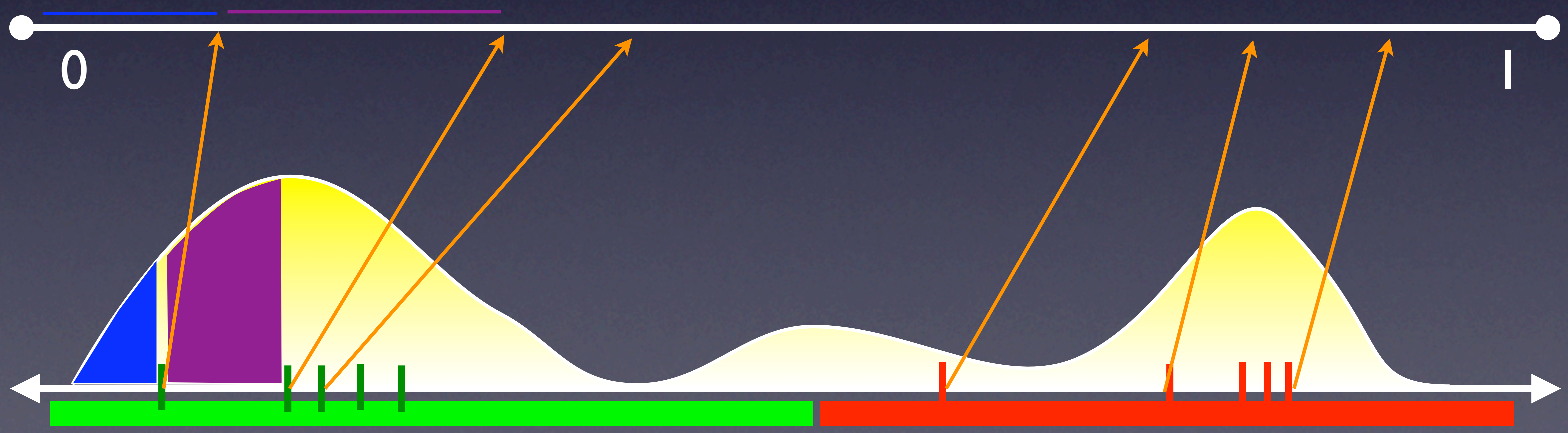
Learning Thresholds

- Proof: Let I be the open interval containing D
  - Step 1: Map I to $(0,1)$ via a function $F(\cdot)$. Transform training sample $S \mapsto S'$ via F
  - The mapping [in pictures]:



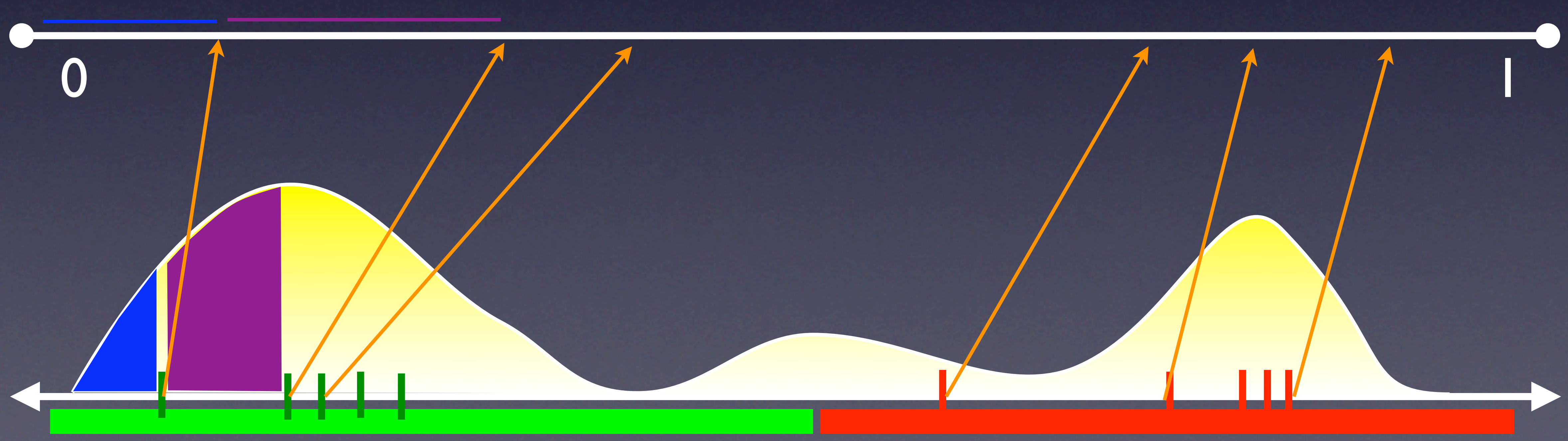
Learning Thresholds

- Proof: Let I be the open interval containing D
  - Step 1: Map I to $(0,1)$ via a function $F(\cdot)$. Transform training sample $S \mapsto S'$ via F
  - The mapping [in pictures]: naturally stretches out D to a uniform distribution in $(0,1)$.
  - Define the corresponding SSL algorithm on $(0,1)$.



Learning Thresholds

- **Proof**: Let I be the open interval containing D
  - Step 3: $\Pr_{S \sim U^m}\left[\mathrm{Err}^U(A(S', U)) \geq \epsilon\right] \leq 2(1-2\epsilon)^m$
    - **Proof**: technical and (unnecessarily?) complicated.
      - Based on bad events. Can possibly be made simpler.



Learning Thresholds

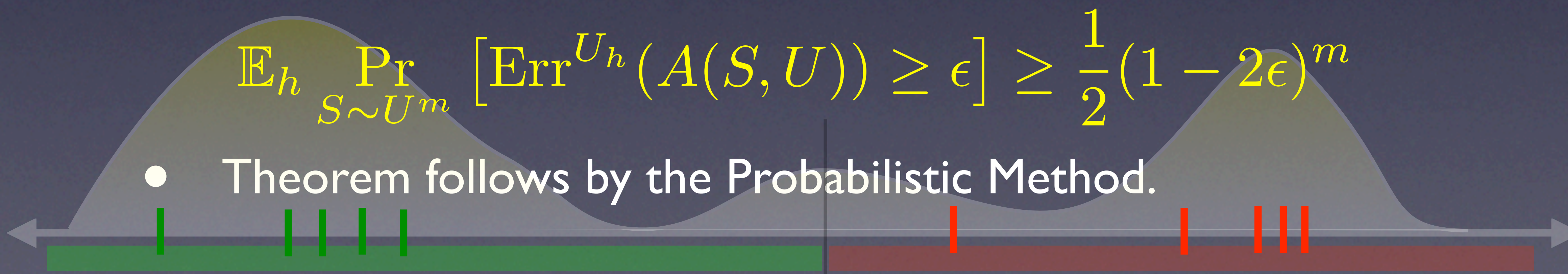
- **Theorem**: For any (randomized) SSL algorithm A, any small ϵ, any δ, any continuous distribution D over an open interval, there exists a target $h \in H$ such that

$$m(A(\cdot, D), \epsilon, \delta) \geq \frac{\ln(1/\delta)}{2.01\epsilon} - \frac{\ln 2}{2.01\epsilon}$$

- **Proof**:
  - Step 1: assume that D is uniform (similar trick as before)
  - Step 2: Fix A, ϵ and δ. Pick $t \sim U(0,1)$ and consider $h = \mathbf{1}_{[0,t]}$
    - h is a random variable. Show that:

$$\mathbb{E}_h \Pr_{S \sim U^m} \left[\mathrm{Err}^{U_h}(A(S,U)) \geq \epsilon\right] \geq \frac{1}{2}(1-2\epsilon)^m$$

    - Theorem follows by the Probabilistic Method.



Learning Thresholds

- Proof (sketch):
  - Step 1: assume that D is uniform (similar trick as before)
  - Step 2: Fix A, ϵ and δ. Pick $t \sim U(0,1)$ and consider $h = \mathbf{1}_{[0,t]}$
    - h is a random variable. Show that:

$$\mathbb{E}_h \Pr_{S \sim U^m} \left[\mathrm{Err}^{U_h}(A(S,U)) \geq \epsilon\right] \geq \frac{1}{2}(1-2\epsilon)^m$$

$$\begin{aligned}
\mathbb{E}_h \Pr_{S \sim U^m} \left[\mathrm{Err}^{U_h}(A(S,U)) \geq \epsilon\right] &= \mathbb{E}_h \Pr_{S \sim U^m} [\text{Bad Event}] \\
&= \mathbb{E}_h \mathbb{E}_{S \sim U^m} \mathbf{1}_{\text{Bad Event}} \\
&= \mathbb{E}_{S \sim U^m} \mathbb{E}_h \mathbf{1}_{\text{Bad Event}} \\
&= \mathbb{E}_{S \sim U^m} \Pr[\text{Bad Event}] \\
&\geq \mathbb{E}_{S \sim U^m} \sum_i \max(x_{i+1} - x_i - 2\epsilon, 0).
\end{aligned}$$

Agnostic Case

- Notion of $b-\text{shatterable}$ distributions: b clusters can be shattered by the concept class. Probabilistic targets.
- Learning thresholds on the real line: $\Theta\left(\ln(1/\delta)/\epsilon^2\right)$
- Union of d intervals: $\Theta\left(\frac{2d+\ln(1/\delta)}{\epsilon^2}\right)$
- Lemma [Anthony, Bartlett]: Suppose P uniformly distributed over two Bernoulli distributions, $\{P_1, P_2\}$ such that probability of heads: $P_1(\mathrm{H}) = 1/2-\gamma, P_2(\mathrm{H}) = 1/2+\gamma$
Further, suppose $\xi_1, \ldots, \xi_m$ are $\{H, T\}$ -valued random variables, with $\Pr[\xi_i = H] = P(H)$ for all i. Then for any decision function $f : \{H,T\}^m \mapsto \{P_1, P_2\}$

$$\mathbb{E}_P \Pr_{\xi\sim P^m} [f(\xi) \neq P] > \frac{1}{4}\left(1-\sqrt{1-\exp\left(\frac{-4m\gamma^2}{1-4\gamma^2}\right)}\right)$$

Agnostic Case

- Lemma: Fix any X, H, D and an $m > 0.$ Suppose there are $h, g \in H$ such that $D(h \Delta g) > 0$. Define probability dist. P_h, P_g over $X \times Y$ such that:

  $$P_h((x, h(x)) \mid x) = P_g((x, g(x)) \mid x) = 1/2 + \gamma$$

  - note that P_h, P_g have the same marginal distribution when h, g agree and have "close" distributions when h, g disagree.

Agnostic Case

- **Lemma:** Fix any X, H, D and an $m > 0.$ Suppose there are $h, g \in H$ such that $D(h\Delta g) > 0$. Define probability dist. P_h, P_g over $X \times Y$ such that:

$$P_h((x, h(x)) \mid x) = P_g((x, g(x)) \mid x) = 1/2 + \gamma$$

- Let $A_D : (h\Delta g \times Y)^m \mapsto H$ be any function. Then for $x_1, \ldots, x_m \in h\Delta g,$ there exists $P \in \{P_h, P_g\}, y_i \sim P_{x_i}$

$$\Pr_{y_i}\left[\mathrm{Err}^P(A_D((x_i, y_i)_{i=1}^m) - \mathrm{OPT}_P > \gamma D(h\Delta g)\right] >$$

$$\frac{1}{4}\left(1 - \sqrt{1 - \exp\left(\frac{-4m\gamma^2}{1 - 4\gamma^2}\right)}\right)$$

Agnostic Case

- Use the previous lemma to construct a probabilistic target that achieves the desired sample complexity.

Conclusion

- Formal analysis of sample complexity of SSL.
- Comparison to sample complexity of supervised learning.
- No assumptions on the relationship between distribution of labels and distribution of unlabeled data.
- Limited advantage of SSL for basic concept classes over the real line.
- Open question: extend the result to any concept class of VC dimension d.

Thank You.