Maximum Entropy for Everyone

John Langford

"Classify like a Bayesian and learn like a Support Vector Machine."

Why another machine learning algorithm?

- Strong overfitting robustness

- Competitive performance

- Implications for learning

Overfitting Robustness

Theoretically:

John Langford, Matthias Seeger, Nimrod Megiddo, "An Improved Predictive Accuracy Bound for Averaging Classifiers" submitted to ICML 2001

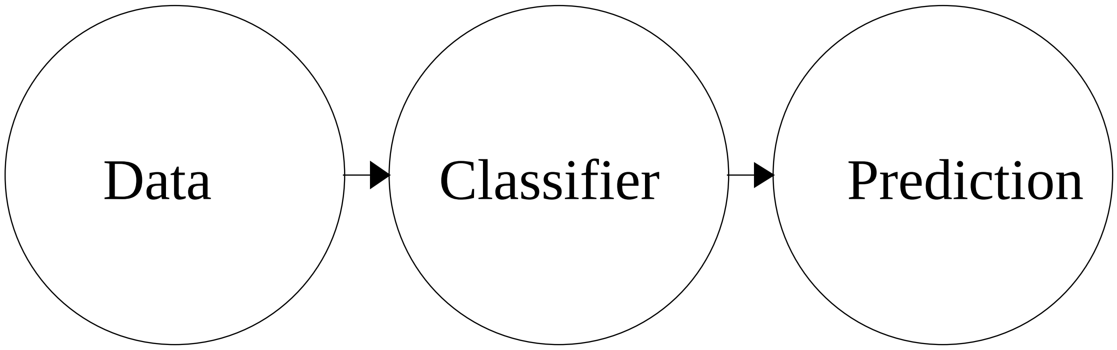
Experimentally: Overfitting appears rare

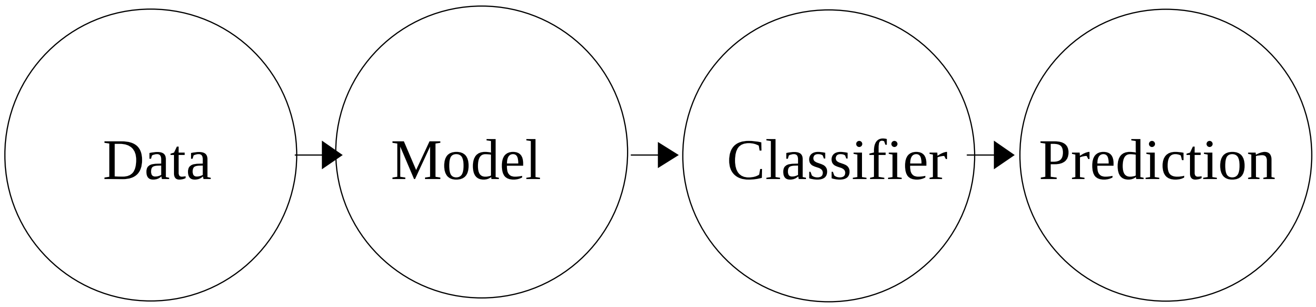
Competitive performance

Training error/testing error

classifier	crabs	breast cancer
SVM, linear	5/3	8/10
SVM, RBF $\sigma = 0.3$	1/18	0/11
SVM, 3rd order	3/6	1/13
Maximum likelihood	4/7	10/16
Maximum Entropy	2/3	3/8

Implications for learning





dd

The plan

1. *How do you classify?*

2. How do you learn?

3. How does this relate to other learning algorithms?

“Classify like a Bayesian and learn like a Support Vector Machine.”

How do you classify?

"Like a Bayesian"

Bayes optimal: $\mathrm{sign}(\int_H P(h)h(x)dh)$

Maximum Entropy: $\mathrm{sign}(\int_H P(h)h(x)dh)$

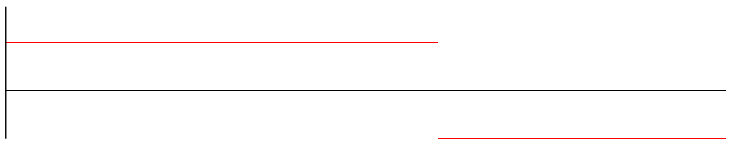
- $P(h)$ is a distribution but not a posterior

- $h(x)$ is a *discriminant* and not a classifier

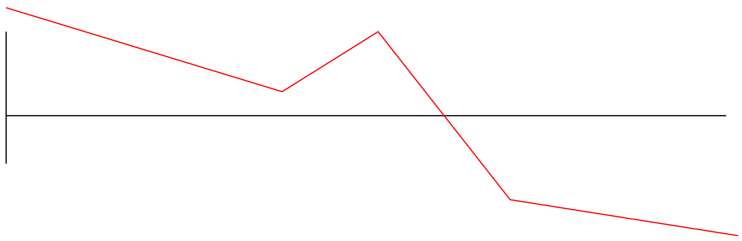
Discriminants

Discriminants are classifiers + confidences

Classifier: $h(x) = \{-1, 1\}$



Discriminant: $h(x) = (-\infty, \infty)$

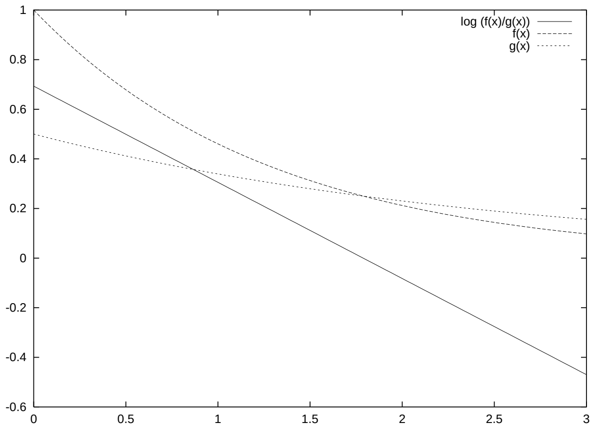


How do you make a discriminant?

How do you make a discriminant?

Pick two distributions, $f_1(x)$ and $f_2(x)$.

$$h(x) = \ln \frac{f_1(x)}{f_2(x)} + b$$



b is a bias.

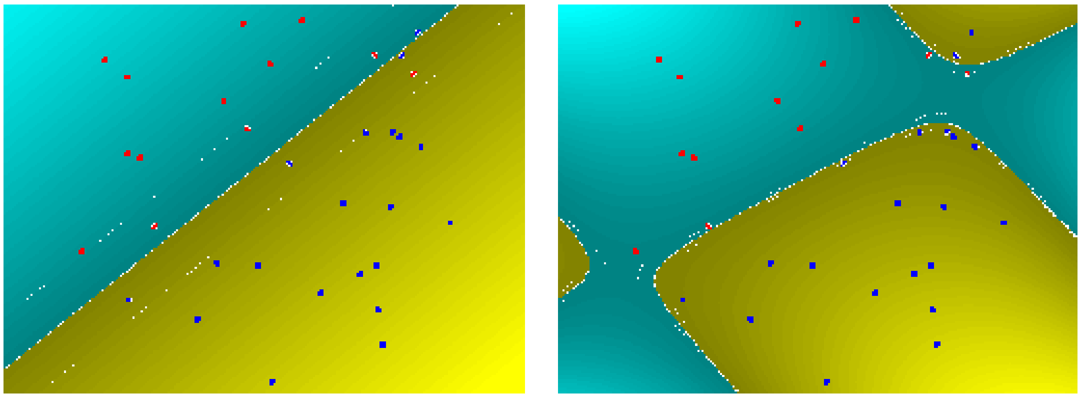
Example

Suppose we have a two dimensional data set.

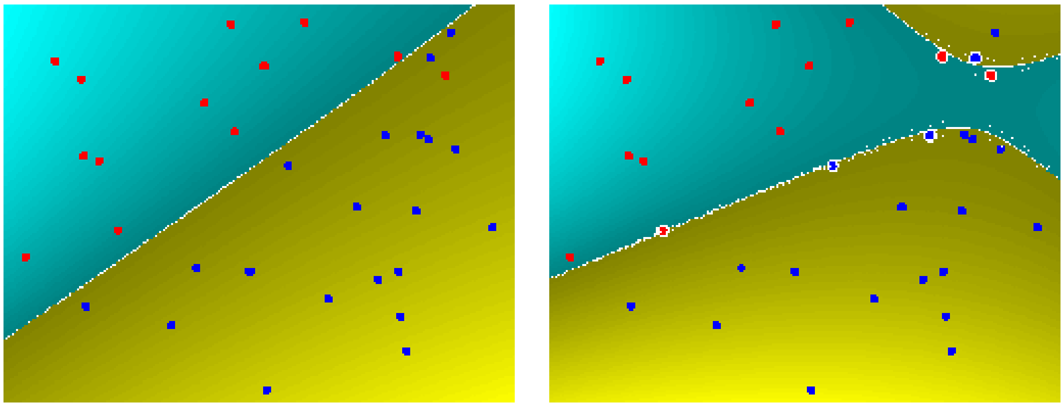
$f_1(x)$ = a gaussian

$f_2(x)$ = a different gaussian

Example



(a) Linear SVM (b) Polynomial Kernel SVM



(a) Max Likelihood (b) Max Ent Discrimination

The plan

1. How do you classify?

2. *How do you learn?*

3. How does this relate to other learning algorithms?

How do you learn? (efficiently)

"Learn like a support vector machine"

We want: the distribution with the largest entropy subject to constraints.

We also want: tractable computation

Why do we want large entropy?

What should the constraints be?

Entropy

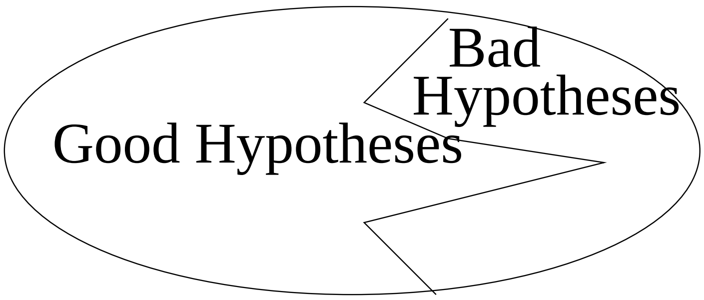
Discriminant distribution: $P(h)$

Entropy: $\int_H P(h) \log \frac{1}{P(h)} dh$

Large entropy $\Rightarrow$ Few choices $\Rightarrow$ No overfitting

Constraints

$y \int_H P(h)h(x)dh > \gamma$ = large margin for every example



Example

Input space: $X = \{0, 1\}$

Discriminant space with two elements.

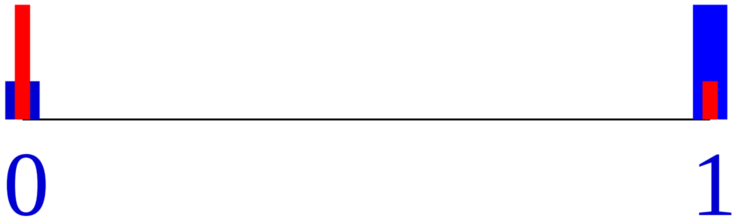
One example: $x = 0$ and $y = 1$

Suppose $\gamma = 0.1$.

hypotheses

$$h_1 = \{f_1(x) = \frac{1}{4} + \frac{x}{2},\ f_{-1}(x) = \frac{3}{4} - \frac{x}{2},\ b = 0\}$$

$$h_2 = \{f_1(x) = \frac{3}{4} - \frac{x}{2},\ f_{-1}(x) = \frac{1}{4} + \frac{x}{2},\ b = 0\}$$



Example, cont.

Our constraint: $y \int p(h)h(x)dh \geq \gamma$

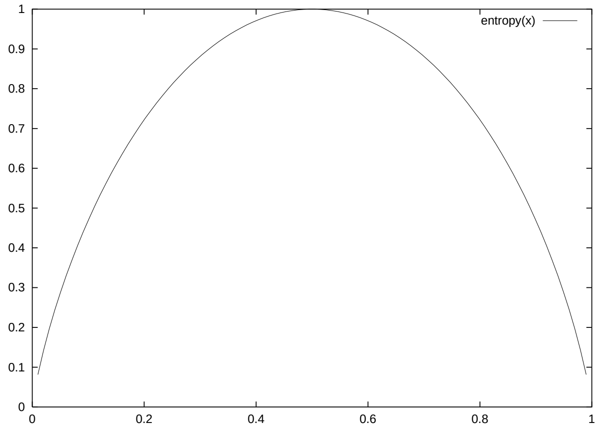
$\Rightarrow p_1 h_1(0) + (1 - p_1)h_2(0) > \gamma$

$h_1(0) = \ln \frac{1}{3}$, $h_2(0) = \ln 3$

Entropy: $p_1 \ln \frac{1}{p_1} + (1 - p_1) \ln \frac{1}{(1-p_1)}$

Example, cont.

Entropy:



Constraint: $p_1 \ln \frac{1}{3} + (1 - p_1) \ln 3 \leq 0.1$

so we have: $P(h_1) \simeq 0.45$ and $P(h_2) \simeq 0.55$

Details to make it all work

- What if the margin γ is impossible to achieve?

- What if we feel Bayesian today and have a prior?

Minimum Relative Entropy Discrimination (MRE)

Minimum Relative Entropy *is* Maximum Entropy except with a prior over the margin γ and discriminant space H.

Relative Entropy = KL Divergence

Relative entropy = entropic distance from your prior.

Prior: $P_0(h, \gamma)$

Relative entropy: $\int P(h, \gamma) \log \frac{P(h,\gamma)}{P_0(h,\gamma)} dh$

Relative constraint: $y \int P(h, \gamma) h(x) dh d\gamma \geq 0$

How can we make this tractable?

Tractable Solution

We need a magic bullet to make all of this feasible.

Magic: $P(h,\gamma) = \frac{P_0(h,\gamma)e^{\sum_i \lambda_i[y_i h(x_i)-\gamma_i]}}{Z(\lambda)}$ where the λ are chosen to minimize $Z(\lambda)$.

Normalization Constant:

$$Z(\lambda) = \int P_0(h,\gamma)e^{\sum_i \lambda_i[y_i h(x_i)-\gamma_i]} dh d\gamma$$

Normalization Constant (= Partition Function)

$Z(\lambda)$ is *critical*. It must be quickly evaluatable and minimizable. No quick evaluation $\Rightarrow$ no quick solution.

When can we evaluate $Z(\lambda)$ quickly?

When can we evaluate $Z(\lambda)$ quickly?

Assume h is parameterized by $\theta_1, ..., \theta_n$.

Helpful hint: if $\sum_i y_i = 0$ independence is preserved

$$P_0(\theta_1, ..., \theta_n, \gamma) = Q_0(\theta_1, ..\theta_i) R_0(\theta_{i+1}, ..., \gamma)$$

$$\Rightarrow P(\theta_1, ..., \theta_n, \gamma) = Q(\theta_1, ..\theta_i) R(\theta_{i+1}, ..., \gamma)$$

The laundry list

- Gaussian based discriminants
- General graphical models with some fixed structure
- Tree models with unfixed structure

Other uses

Anomaly detection: How do you learn with examples from only one class?

Uncertain labels: What if you don't know what the output was with complete certainty?

The plan

1. How do you classify?

2. How do you learn?

3. *How does this relate to other learning algorithms?*

How does this relate to other learning algorithms?

- Support Vector Machines
- Bayes-Optimal

Support Vector Machines

Training: The optimization problem is often reduced to solving for Lagrange multipliers.

Formally: SVMs are asymptotically equivalent for a certain sequence of priors and discriminants.

Bayes-Optimal

Very similar classification

Maximum Entropy might(?) be more robust when the underlying model is wrong.

The algorithm

1. Train by finding the maximum entropy distribution over the discriminant space subject to the constraint that examples are usually correct under the distribution.

2. Test by averaging with respect to the maximum entropy distribution.