

Finally: the "kernel trick"!

maximize_{$$\alpha$$} $\sum_{i} \alpha_{i} - \frac{1}{2} \sum_{i,j} \alpha_{i} \alpha_{j} y_{i} y_{j} K(\mathbf{x}_{i}, \mathbf{x}_{j})$

$$K(\mathbf{x}_{i}, \mathbf{x}_{j}) = \Phi(\mathbf{x}_{i}) \cdot \Phi(\mathbf{x}_{j})$$

$$\sum_{i} \alpha_{i} y_{i} = 0$$

$$C \ge \alpha_{i} \ge 0$$

- Never represent features explicitly
 Compute dot products in closed form
- Constant-time high-dimensional dotproducts for many classes of features
- Very interesting theory Reproducing Kernel Hilbert Spaces
 - □ Not covered in detail in 10701/15781, more in 10702

 $\mathbf{w} = \sum_{i} \alpha_{i} y_{i} \Phi(\mathbf{x}_{i})$

 $b = y_k - \mathbf{w}.\Phi(\mathbf{x}_k)$ for any k where $C > lpha_k > 0$

©Carlos Guestrin 2005-2009

Common kernels



Polynomials of degree d

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{v})^d$$

Polynomials of degree up to d

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{v} + 1)^d$$

Gaussian kernels

$$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{||\mathbf{u} - \mathbf{v}||}{2\sigma^2}\right)$$

Sigmoid

$$K(\mathbf{u}, \mathbf{v}) = \tanh(\eta \mathbf{u} \cdot \mathbf{v} + \nu)$$

©Carlos Guestrin 2005-2009

.

Overfitting?



- Huge feature space with kernels, what about overfitting???
 - Maximizing margin leads to sparse set of support vectors
 - □ Some interesting theory says that SVMs search for simple hypothesis with large margin
 - ☐ Often robust to overfitting

©Carlos Guestrin 2005-200

What about at classification time



For a new input \mathbf{x} , if we need to represent $\Phi(\mathbf{x})$, we are in trouble!

- Recall classifier: sign(w.Φ(x)+b)
- Using kernels we are cool!

$$K(\mathbf{u}, \mathbf{v}) = \Phi(\mathbf{u}) \cdot \Phi(\mathbf{v})$$

ecall classifier: sign(
$$\mathbf{w}.\Phi(\mathbf{x})$$
+b) sing kernels we are cool!
$$K(\mathbf{u},\mathbf{v}) = \Phi(\mathbf{u})\cdot\Phi(\mathbf{v})$$
 be $b = y_k - \mathbf{w}.\Phi(\mathbf{x}_k)$ for any k where $C > \alpha_k > 0$

SVMs with kernels



- Choose a set of features and kernel function
- Solve dual problem to obtain support vectors α_i
- At classification time, compute:

$$\mathbf{w} \cdot \Phi(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i)$$

$$b = y_k - \sum_i \alpha_i y_i K(\mathbf{x}_k, \mathbf{x}_i)$$
 for any k where $C > \alpha_k > 0$

Remember kernel regression



Remember kernel regression???

- $w_i = \exp(-D(x_i, query)^2 / K_w^2)$
- How to fit with the local points?

Predict the weighted average of the outputs: predict = $\sum w_i y_i / \sum w_i$

SVMs v. Kernel Regression



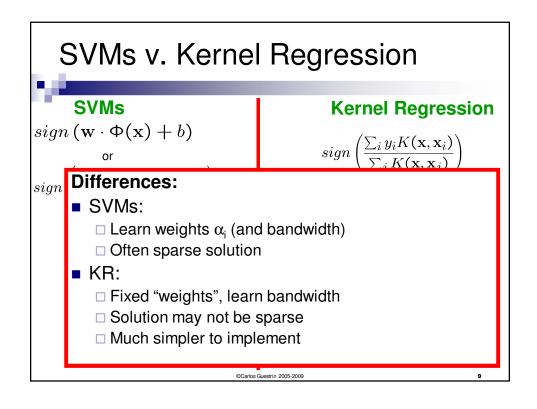
SVMs

$$sign\left(\mathbf{w}\cdot\Phi(\mathbf{x})+b\right)$$

or
$$sign\left(\sum_{i}\alpha_{i}y_{i}K(\mathbf{x},\mathbf{x}_{i})+b\right)$$

Kernel Regression

$$sign\left(\frac{\sum_{i} y_{i} K(\mathbf{x}, \mathbf{x}_{i})}{\sum_{j} K(\mathbf{x}, \mathbf{x}_{j})}\right)$$



	SVMs	Logistic Regression
Loss function		
High dimensional features with kernels		

Kernels in logistic regression

$$P(Y = 1 \mid x, \mathbf{w}) = \frac{1}{1 + e^{-(\mathbf{w} \cdot \Phi(\mathbf{x}) + b)}}$$

Define weights in terms of support vectors:

$$\mathbf{w} = \sum_{i} \alpha_{i} \Phi(\mathbf{x}_{i})$$

$$P(Y = 1 \mid x, \mathbf{w}) = \frac{1}{1 + e^{-(\sum_{i} \alpha_{i} \Phi(\mathbf{x}_{i}) \cdot \Phi(\mathbf{x}) + b)}}$$

$$= \frac{1}{1 + e^{-(\sum_{i} \alpha_{i} K(\mathbf{x}, \mathbf{x}_{i}) + b)}}$$

lacksquare Derive simple gradient descent rule on α_{i}

©Carlos Guestrin 2005-2009

4.

What's the difference between SVMs and Logistic Regression? (Revisited)



	SVMs	Logistic Regression
Loss function	Hinge loss	Log-loss
High dimensional features with kernels	Yes!	Yes!

estrin 2005-2009

What you need to know

- Dual SVM formulation
 - ☐ How it's derived
- The kernel trick
- Derive polynomial kernel
- Common kernels
- Kernelized logistic regression
- Differences between SVMs and logistic regression

©Carlos Guestrin 2005-200

13

14

PAC-learning, VC Dimension Machine Learning – 10701/15781 Carlos Guestrin Carnegie Mellon University October 26th, 2009

©Carlos Guestrin 2005-2009

What now...



- We have explored many ways of learning from data
- But...
 - ☐ How good is our classifier, really?
 - □ How much data do I need to make it "good enough"?

©Carlos Guestrin 2005-2009

15

A simple setting...



- Classification
 - □ m data points
 - ☐ **Finite** number of possible hypothesis (e.g., dec. trees of depth d)
- A learner finds a hypothesis h that is consistent with training data
 - \Box Gets zero error in training error_{train}(h) = 0
- What is the probability that h has more than ε true error?
 - \square error_{true}(h) $\geq \varepsilon$

©Carlos Guestrin 2005-2009

How likely is a bad hypothesis to get *m* data points right?

- Hypothesis h that is consistent with training data → got m i.i.d. points right
 - □ h "bad" if it gets all this data right, but has high true error
- Prob. h with error_{true}(h) $\geq \varepsilon$ gets one data point right
- Prob. *h* with error_{true}(h) ≥ ε gets *m* data points right

©Carlos Guestrin 2005-2009

17

But there are many possible hypothesis that are consistent with training data

©Carlos Guestrin 2005-2009

How likely is learner to pick a bad hypothesis

- Prob. h with error_{true}(h) $\geq \varepsilon$ gets m data points right
- There are *k* hypothesis consistent with data
 ☐ How likely is learner to pick a bad one?

©Carlos Guestrin 2005-2009

19

Union bound

■ P(A or B or C or D or ...)

©Carlos Guestrin 2005-2009

How likely is learner to pick a bad hypothesis

- Prob. h with error_{true}(h) $\geq \varepsilon$ gets m data points right
- There are *k* hypothesis consistent with data

 ☐ How likely is learner to pick a bad one?

©Carlos Guestrin 2005-200

21

Review: Generalization error in finite hypothesis spaces [Haussler '88]

■ **Theorem**: Hypothesis space H finite, dataset D with m i.i.d. samples, $0 < \varepsilon < 1$: for any learned hypothesis h that is consistent on the training data:

$$P(\mathsf{error}_{true}(h) > \epsilon) \le |H|e^{-m\epsilon}$$

©Carlos Guestrin 2005-2009

Using a PAC bound

- Typically, 2 use cases: $P(\text{error}_{true}(h) > \epsilon) \leq |H|e^{-m\epsilon}$
 - \square 1: Pick ε and δ , give you m
 - \square 2: Pick m and $\delta,$ give you ϵ

©Carlos Guestrin 2005-2009

23

Review: Generalization error in finite hypothesis spaces [Haussler '88]

■ **Theorem**: Hypothesis space H finite, dataset D with m i.i.d. samples, $0 < \varepsilon < 1$: for any learned hypothesis h that is consistent on the training data:

$$P(\mathsf{error}_{true}(h) > \epsilon) \le |H|e^{-m\epsilon}$$

Even if h makes zero errors in training data, may make errors in test

©Carlos Guestrin 2005-2009

Limitations of Haussler '88 bound

- $P(\operatorname{error}_{true}(h) > \epsilon) \le |H|e^{-m\epsilon}$
 - Consistent classifier

■ Size of hypothesis space

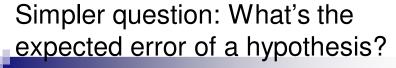
©Carlos Guestrin 2005-200

25

What if our classifier does not have zero error on the training data?

- - A learner with zero training errors may make mistakes in test set
 - What about a learner with *error*_{train}(h) in training set?

©Carlos Guestrin 2005-2009



- The error of a hypothesis is like estimating the parameter of a coin!
- Chernoff bound: for m i.i.d. coin flips, $x_1,...,x_m$, where $x_i \in \{0,1\}$. For $0 < \varepsilon < 1$:

$$P\left(\theta - \frac{1}{m}\sum_{i} x_{i} > \epsilon\right) \le e^{-2m\epsilon^{2}}$$

©Carlos Guestrin 2005-2009

27

Using Chernoff bound to estimate error of a single hypothesis

$$P\left(\theta - \frac{1}{m}\sum_{i} x_{i} > \epsilon\right) \le e^{-2m\epsilon^{2}}$$

©Carlos Guestrin 2005-2009

But we are comparing many hypothesis: **Union bound**

For each hypothesis h_i:

$$P\left(\operatorname{error}_{true}(h_i) - \operatorname{error}_{train}(h_i) > \epsilon\right) \le e^{-2m\epsilon^2}$$

What if I am comparing two hypothesis, h₁ and h₂?

©Carlos Guestrin 2005-200

29

Generalization bound for |H| hypothesis



■ **Theorem**: Hypothesis space H finite, dataset D with m i.i.d. samples, $0 < \varepsilon < 1$: for any learned hypothesis h:

$$P\left(\operatorname{error}_{true}(h) - \operatorname{error}_{train}(h) > \epsilon\right) \le |H|e^{-2m\epsilon^2}$$

©Carlos Guestrin 2005-2009

PAC bound and Bias-Variance tradeoff

$$P\left(\operatorname{error}_{true}(h) - \operatorname{error}_{train}(h) > \epsilon\right) \le |H|e^{-2m\epsilon^2}$$

or, after moving some terms around, with probability at least 1-δ:

error
$$_{true}(h) \leq \operatorname{error}_{train}(h) + \sqrt{rac{\ln |H| + \ln rac{1}{\delta}}{2m}}$$

■ Important: PAC bound holds for all *h*, but doesn't guarantee that algorithm finds best *h*!!!

©Carlos Guestrin 2005-2009