

# **CS195-5 : Introduction to Machine Learning**

## **Lecture 18**

Greg Shakhnarovich

October 25, 2006

---

# Announcements

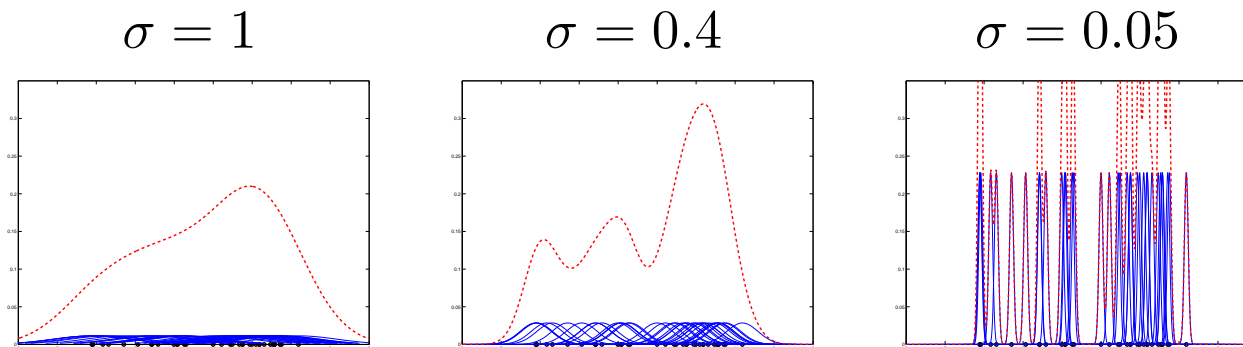
- Reminder: midterm on Friday in CIT 367, 11am.
- PS4 out on Friday.

# Review

- Kernel density estimation:

$$\hat{p}(\mathbf{x}_0) = \frac{1}{N} \sum_{i=1}^N K(\mathbf{x}_0, \mathbf{x}_i).$$

- $K$  must be a valid pdf itself, e.g.  $K(\mathbf{x}, \mathbf{z}) = \mathcal{N}(\mathbf{x} - \mathbf{z}; 0, \sigma^2 \mathbf{I})$
- The effect of setting the kernel width:



---

# Nearest neighbor classification

- Predict  $y_0$  using the label of the nearest neighbor of  $\mathbf{x}_0$  among training examples.
- For  $C = 2$ ,  $R^* \leq R_\infty \leq 2R^*(1 - R^*)$ .
  - In general,  $R^* \leq R_\infty \leq 2R^*(1 - \frac{C}{C-1}R^*)$
- Note:  $R^* \leq \frac{C-1}{C}$  so this means  $R_\infty \leq 2R^*$ .
- Main disadvantages:
  - Rate of convergence  $R_N \rightarrow R_\infty$  can be slow.
  - Expensive to compute, especially with lots of data in high dimensions!

---

## Midterm review: concepts

- Loss:  $L : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$
- Risk of a hypothesis  $h : \mathcal{X} \rightarrow \mathcal{Y}$ :

$$R(h) = E_{(\mathbf{x}_0, y_0)} [L(h(\mathbf{x}_0), y_0)]$$

- Empirical loss on training data  $X_N = \{\mathbf{x}_i, y_i\}, i = 1, \dots, N$ :

$$L_N(h) = \frac{1}{N} \sum_{i=1}^N L(h(\mathbf{x}_i), y_i)$$

- Learning via empirical risk minimization:  $h^* = \operatorname{argmin}_h L_N(h)$ .

---

# Regression

- Statistical model:  $y = f(\mathbf{x}; \mathbf{w}) + \nu$ .
- Linear fit:  $\hat{y}(\mathbf{x}) = \sum_{j=0} w_j x_j$ ,  $x_0 \equiv 1$ .
- Minimum squared loss  $\equiv$  ML with Gaussian noise assumption

$$\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- Prediction errors have zero mean and zero correlation with any *linear* function of  $\mathbf{x}$ .
- Generalized linear model:  $f(\mathbf{x}; \mathbf{w}) = \sum_{j=0}^m w_j \phi_j(\mathbf{x})$ , with  $\phi_0(\mathbf{x}) \equiv 1$ .
  - Solution identical to linear case, with suitable  $\mathbf{X}$ .

---

# Estimation

- ML estimator based on data  $X_N$ :

$$\hat{\theta}_{ML} = \underset{\theta}{\operatorname{argmax}} p(X_N; \theta).$$

- Properties of an estimator  $\hat{\theta}$  with respect to the true  $\theta^*$ :

- $\operatorname{bias}(\hat{\theta}) = E_{X_N} [\hat{\theta} - \theta^*]$

- Variance:  $\operatorname{var}(\hat{\theta}) = E_{X_N} \left[ \left( \hat{\theta} - E_{X_N} [\hat{\theta}] \right)^2 \right]$

- Consistency:  $\lim_{N \rightarrow \infty} \hat{\theta}(X_N) = \theta^*$

- $\text{error} = \text{bias}^2 + \text{variance}$ .

- Cramer-Rao inequality: min variance for an unbiased estimator.

---

# Bias-variance tradeoff and overfitting

- In regression/classification, error = bias<sup>2</sup>+variance+irreducible error.
- Model flexibility (complexity) is directly related to variance, inversely related to bias.
- Cross-validation is a tool to control overfitting.

---

# Regularization

- Main idea: constrain the model by penalizing the objective (likelihood, empirical error etc.) by complexity, e.g. magnitude of coefficients.
- Ridge regression: penalty is  $\lambda \sum_j w_j^2$ .
- Lasso regression: penalty is  $\lambda \sum_j |w_j|$ .
- Naive Bayes: forces diagonal feature covariance.
- SVM:  $\min \|\mathbf{w}\|$  subject to margin constraints.

---

# Bayesian estimation

- Prior: captures our belief about parameters.
- Ridge regression and lasso can be expressed as MAP estimation with Gaussian/Laplacian priors.
- Conjugate prior for a given likelihood function: posterior has the same form.
  - Example: Beta prior for binomial likelihood

$$p(\theta | X_N) \propto \theta^{N_1} (1 - \theta)^{N_0} \cdot \text{Beta}(a, b) \propto \text{Beta}(\theta; N_1 + a, N_0 + b).$$

---

# Classification

- Bayes (optimal) classifier:  $h(\mathbf{x}) = \operatorname{argmax}_c p(c | \mathbf{x})$
- Generative methods: model  $p(\mathbf{x} | c), p(c) \Rightarrow$  discriminants  $\delta_c(\mathbf{x}) = \log p(\mathbf{x} | c) + \log p(c)$ 
  - Gaussian: linear/quadratic discriminants depending on the covariances
- Fisher's LDA: find projection

$$\operatorname{argmax}_{\mathbf{w}} \frac{\text{separation between projected means}^2}{\text{sum of projected within-class variances}}$$

- Optimal if the classes are equi-variant Gaussians.

---

# Classification

- Discriminative methods: model  $p(c | \mathbf{x})$  directly.
- Logistic regression: fit  $p(c | \mathbf{x})$  with a sigmoid function (via gradient ascent).
- Generalized linear LR:

$$\hat{p}(1 | \mathbf{x}) = \sigma \left( \sum_{j=0}^m w_j \phi_j(\mathbf{x}) \right)$$

- Multiclass: softmax

$$\hat{p}(c | \mathbf{x}) = \frac{\exp \left( \sum_{j=0}^m w_j \phi_{jc}(\mathbf{x}) \right)}{\sum_{k=1}^C \exp \left( \sum_{j=0}^m w_j \phi_{jk}(\mathbf{x}) \right)}$$

---

# SVM

- Two key ideas:
  - Max-margin classification,
  - The kernel trick.
- SVM classifier:

$$\hat{y} = \text{sign} \left( w_0 + \sum_{\alpha_i > 0} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) \right).$$

with  $\alpha_i > 0$  corresponding to support vectors.