

Machine Learning

Naïve Bayes Model, Error Decomposition

DSC 240

Feb 18, 2025

Instructor: Prof. Yu-Xiang Wang

Last lecture

- Probabilistic models and max-likelihood estimation (MLE)
- MLE for Bernoulli estimation, Gaussian estimation
- Derive square loss and logistic loss
- The “plug-in” principle

Today

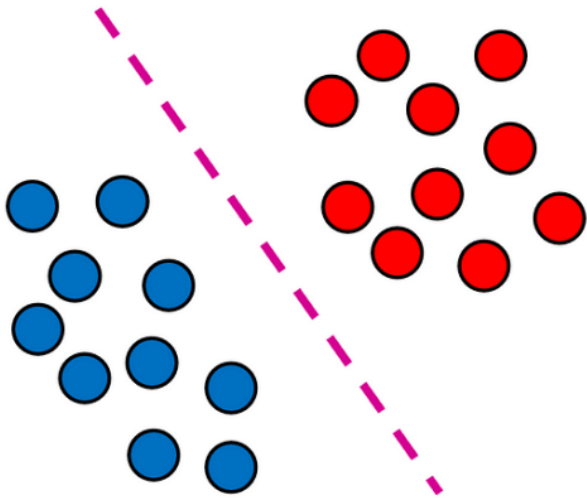
- Naïve Bayes models
 - Multinomial Naïve Bayes Model for “text classification”
 - Gaussian Naïve Bayes model
- Loss, Risk and Empirical Risk
 - Risk decomposition: Approximation error, Optimization error and generalization
- Motivate our next topic “decision trees and boosting”

Recap: We learned about directly modelling the predictive functions. There is another way... called “Probabilistic modelling”

- We can model how the data is generated in the first place.
 - Model the labeling process via a **conditional distribution $P(y | x)$** . This is known as a ***(probabilistic) discriminative model***.
 - Specifying decision-trees / linear classifiers / shapes of decision boundaries should be considered non-probabilistic discriminative models.
 - Model the **joint distribution $P(x,y)$** . Often one models the label distribution **$P(y)$** and a generative process **$P(x|y)$** . This is known as a ***generative model***.
- The natural prediction would be
 - $h(x) = \operatorname{argmax}_y P(y | x)$
 - If the data generative process is indeed $P(y|x)$, then this is “Bayes optimal”.

Discriminative models vs generative models

Discriminative



Generative

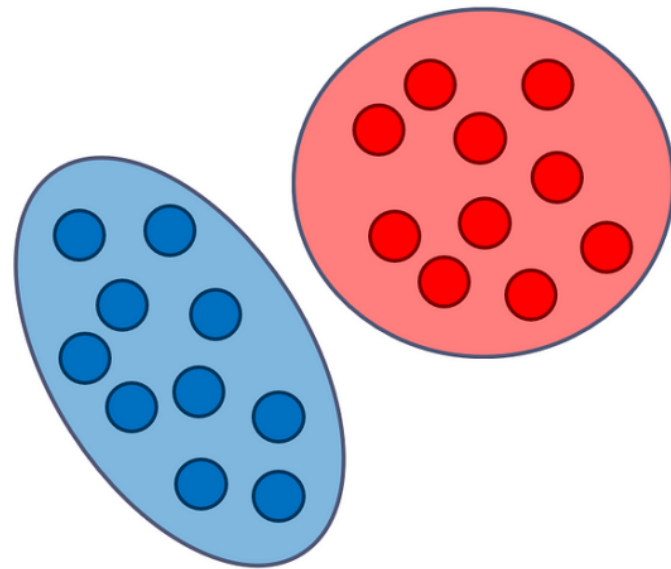


Image Credit: Dr. Roi Yehoshua

Generative model builds a world. How do we model the joint distribution $p(x,y)$?

- Prior: $p(y)$
- Per-class generative distribution $p(x|y)$
- Then how do we make inference about the label given feature?
 - Use $p(y|x)$. How to derive it? By Bayes rule:
- How to we determine $p(y)$ and $p(x|y)$?
 - Fit them using data by MLE!

Modeling $p(x|y)$ is challenging

Consider a dataset with 16 attributes (lets assume they are all binary).

How many parameters to we need to estimate to fully determine $p(\mathbf{X}|Y)$?

age	employe	education	edun	marital	...	job	relation	race	gender	hour	country	wealth
39	State_gov	Bachelors	13	Never_mar	...	Adm_cleric	Not_in_fam	White	Male	40	United_States	poor
51	Self_emp	Bachelors	13	Married	...	Exec_man	Husband	White	Male	13	United_States	poor
39	Private	HS_grad	9	Divorced	...	Handlers_c	Not_in_fam	White	Male	40	United_States	poor
54	Private	11th	7	Married	...	Handlers_c	Husband	Black	Male	40	United_States	poor
28	Private	Bachelors	13	Married	...	Prof_speci	Wife	Black	Female	40	Cuba	poor
38	Private	Masters	14	Married	...	Exec_man	Wife	White	Female	40	United_States	poor
50	Private	9th	5	Married_sp	...	Other_serv	Not_in_fam	Black	Female	16	Jamaica	poor
52	Self_emp	HS_grad	9	Married	...	Exec_man	Husband	White	Male	45	United_States	rich
31	Private	Masters	14	Never_mar	...	Prof_speci	Not_in_fam	White	Female	50	United_States	rich
42	Private	Bachelors	13	Married	...	Exec_man	Husband	White	Male	40	United_States	rich
37	Private	Some_coll	10	Married	...	Exec_man	Husband	Black	Male	80	United_States	rich
30	State_gov	Bachelors	13	Married	...	Prof_speci	Husband	Asian	Male	40	India	rich
24	Private	Bachelors	13	Never_mar	...	Adm_cleric	Own_child	White	Female	30	United_States	poor
33	Private	Assoc_acc	12	Never_mar	...	Sales	Not_in_fam	Black	Male	50	United_States	poor
41	Private	Assoc_voc	11	Married	...	Craft_repa	Husband	Asian	Male	40	*MissingV	rich
34	Private	7th_8th	4	Married	...	Transport	Husband	Amer_Indi	Male	45	Mexico	poor
26	Self_emp	HS_grad	9	Never_mar	...	Farming_fi	Own_child	White	Male	35	United_States	poor
33	Private	HS_grad	9	Never_mar	...	Machine_c	Unmarried	White	Male	40	United_States	poor
38	Private	11th	7	Married	...	Sales	Husband	White	Male	50	United_States	poor
44	Self_emp	Masters	14	Divorced	...	Exec_man	Unmarried	White	Female	45	United_States	rich
41	Private	Doctorate	16	Married	...	Prof_speci	Husband	White	Male	60	United_States	rich

Learning the values for the full conditional probability table would require enormous amounts of data.

Simple example

Binary vectors, 2^3 rows +
binary output $Y \in \{0, 1\}$

x_1	x_2	x_3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

1. What is the number of parameters required to determine $p(X|y)$?

2. What happens if there are d Boolean features?

Naïve Bayes assumption

- Naïve Bayes classifiers assume that given the class label Y the features are **conditional independent** of each other

$$p(\mathbf{X}|y) = \prod_j p_j(x^j|Y)$$

- p_j : specific model for attribute j .

Simple example with naïve Bayes assumption

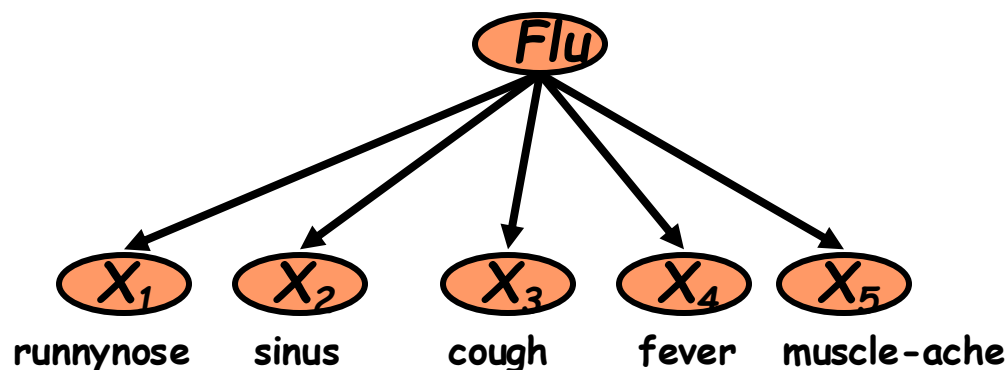
Binary vectors, 2^3 rows +
binary output $Y \in \{0, 1\}$

x_1	x_2	x_3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

1. What is the number of parameters required to determine $p(X|y)$?

.2 What happens if there are d Boolean features?

Example: The Naïve Bayes Classifier



- **Conditional Independence Assumption:** features are independent of each other given the class:

$$P(X_1, \dots, X_5 | C) = P(X_1 | C) \cdot P(X_2 | C) \cdot \dots \cdot P(X_5 | C)$$

- **Interpretation:** given that you have Flu, the event that you experience each of the five symptoms are independent (is this a valid assumption?)
- **The task of classification:** Predict disease using symptoms

Example: Text classification

Machine learning is a subset of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. In other words, it's a process of data analysis that automates analytical model building. Machine learning involves the creation and use of algorithms that can learn from and make decisions or predictions based on data...



Human ?



Machine ?

X is a sequence of words

y is "Human" or "Machine"

Naïve Bayes Assumption:

$P(X | \text{"Machine"}) = P(\text{Word 1} | \text{"Machine"}) \dots P(\text{Word N} | \text{"machine"})$

MLE for fitting the Naïve Bayes Model for text classification

All machine generated texts.
(m documents
 i th document with N_i
words.)

$$\hat{p}(y = \text{"machine"}) = \frac{m}{m + n}$$

$$\hat{p}(\text{Word} = w | y = \text{"machine"}) = \frac{\# \text{ of } w}{\sum_{i=1}^m N_i}$$

All human written text.
(n documents. j th
document with N_j words.)

$$\hat{p}(y = \text{"human"}) = \frac{n}{m + n}$$

$$\hat{p}(\text{Word} = w | y = \text{"human"}) = \frac{\# \text{ of } w}{\sum_{j=1}^n N_j}$$

2 min exercise: Naïve Bayes Model for text classification

Sentence 1: "Why do you cry?"
Sentence 2: "Yes."
Sentence 3: "Hasta La Vista,
Baby!"
Sentence 4: "I will be back."

Sentence 1: "You mean
people?"
Sentence 2 "I don't know. We
just cry."
Sentence 3: "No, no, no, no.
You gotta listen to the way
people talk. "

- What is our estimate of $P(y)$?
- What is our estimate for $P(\text{word} = \text{"No"} | y)$ and $P(\text{word} = \text{"You"} | y)$?

Prediction with Naïve Bayes Model: “Just plug in”

Once we computed all parameters for attributes in both classes we can easily decide on the label of a **new** sample \mathbf{x} :

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y_k} p(y = y_k | \mathbf{x}) \\ &= \operatorname{argmax}_{y_k} \frac{p(\mathbf{x} | y = y_k) p(y = y_k)}{p(\mathbf{x})} \\ &= \operatorname{argmax}_{y_k} \prod_j p(\mathbf{x}^j | y = y_k) p(y = y_k)\end{aligned}$$

Try it on this example:
New sentence: “I know now why you cry”

- Perform this computation for both class 1 and class 2 and select the class that leads to a higher probability as your decision

Exercise: Applying Naïve Bayes model

S1: "Why do you cry?"

S2: "Yes."

S3: "Hasta La Vista,
Baby!"

S4: "I will be back."

S1: "You mean people?"

S2: "I don't know. We just
cry."

S3: "No, no, no, no. You
gotta listen to the way
people talk. "

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y_k} p(y = y_k | \mathbf{x}) \\ &= \operatorname{argmax}_{y_k} \frac{p(\mathbf{x} | y = y_k) p(y = y_k)}{p(\mathbf{x})} \\ &= \operatorname{argmax}_{y_k} \prod_j p(\mathbf{x}^j | y = y_k) p(y = y_k)\end{aligned}$$

- New sentence: "*I know now why you cry*"
- How does the algorithm classify this sentence?
 - Need $p(y)$ and $p(x|y)$.

Naïve Bayes model with continuous variables

- So far we assumed a binomial or discrete distribution for the data given the model ($p(\mathbf{x}^i|y)$)
- However, in many cases the data contains continuous features:
 - Height, weight, Levels of genes in cells, Brain activity
- **Gaussian Naïve Bayes model:**

$$x_i|y \sim \mathcal{N}(\mu_y, \sigma_y^2)$$

You will learn more about Gaussian Naïve Bayes models in HW3!

What you will learn in HW3

- Decision boundaries of Naïve bayes classifiers
- Naïve Bayes vs linear classifier

One more thing: regularization from probabilistic modeling point of view

- Recall:
 - Linear regression / OLS \Leftrightarrow MLE under Gaussian Linear model
 - Logistic regression \Leftrightarrow MLE under Bernoulli-Logistic model
- How about L1 / L2 regularized linear / logistic regression?
 - Do they have a probabilistic interpretation?
 - Turns out they correspond to Maximum A Posteriori estimate under certain Bayesian models, where the parameters are also (assumed to be) random
 - After observing data, the distribution of the parameters are updated from prior to posterior.

An example of a Bayesian model and prior

$$\theta \sim \mathcal{N}(0, s^2)$$

$$y_i \sim \mathcal{N}(x_i^T \theta, \sigma^2)$$

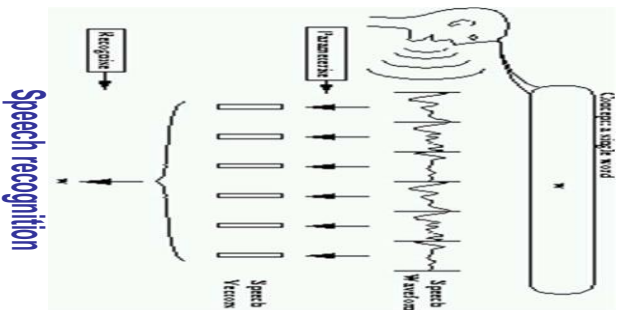
- Maximum A Posteriori (MAP) Estimation

$$\hat{\theta} = \arg \max_{\theta} p(\text{Data}|\theta)\pi(\theta)$$

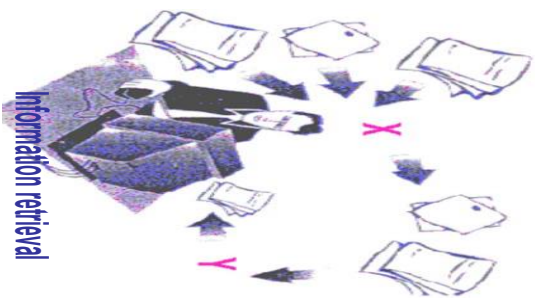
Checkpoint: Probabilistic models

- Probabilistic / Generative / Bayesian models are very powerful and interpretable method for modeling the world.
 - Customize ML models for your applications.
 - Explicitly model the dependence and causality
 - Encode domain knowledge using “priors”.
 - Naïve Bayes model is the simplest of them all!
- Resources: Constructing Probabilistic models.
 - Learn more about Bayesian Networks / Causality / Graphical models in a dedicated course.
 - Open source tools for constructing such models. Stan: <https://mc-stan.org/> , JAGS: <https://mcmc-jags.sourceforge.io/>

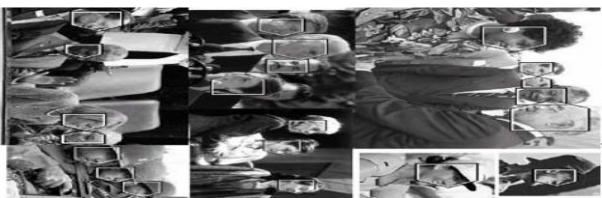
Checkpoint: many applications of probabilistic modeling



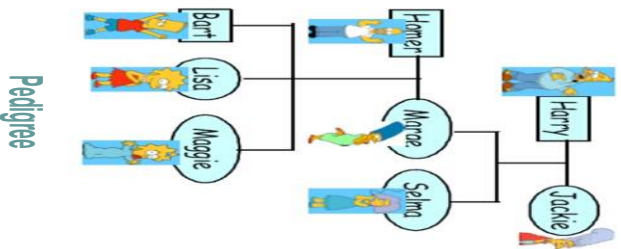
Speech recognition



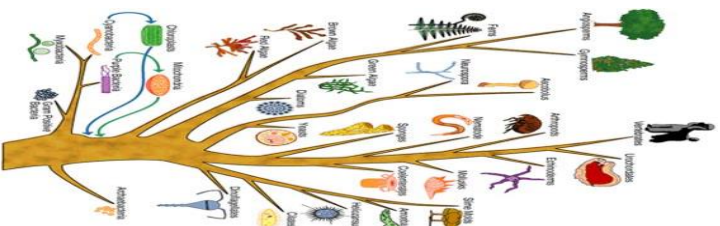
Information retrieval



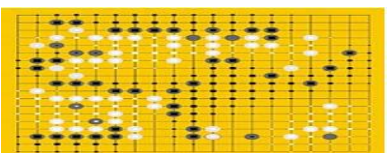
Computer vision



Pedigree



Evolution



Games



Robotic control



Planning

© Eric Xing @ CMU, 2005-2014

Revisit the problem of machine learning... now that we learned probabilities / statistics

- What can we assume about the data?
 - Our model of $P(y|x)$ or $P(x|y)$ is correct?
 - Is what we derive based on the assumptions that the probabilistic model is correct valid?
- **Statistics:** Let's assume these models are correct so we can make inference about the true parameter.
 - Need more assumptions.
- **Machine learning:** The model is always wrong. Let's not worry about the true parameters and focus on prediction!
 - Need less assumptions

Loss, Empirical Risk, and Risk

- Loss function

$$\ell(h, (x, y))$$

- Empirical Risk function

$$\hat{R}(h, \text{Data}) = \frac{1}{n} \sum_{i=1}^n \ell(h, (x_i, y_i))$$

- (Population) Risk function

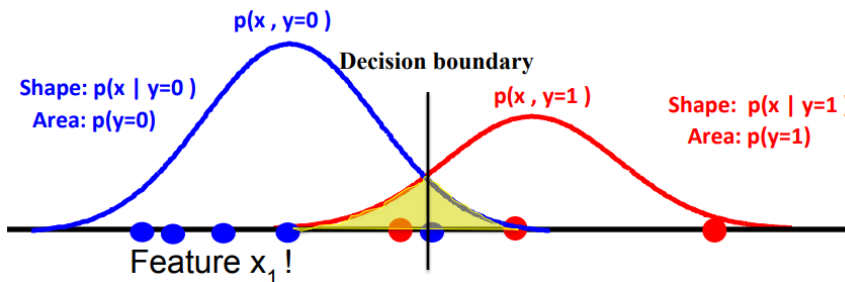
$$R(h, \mathcal{D}) = \mathbb{E}_{\mathcal{D}}[\ell(h, (x_i, y_i))]$$

Bayes optimal classifier, optimal classifier within the hypothesis class, Empirical Risk Minimizer

- Bayes Optimal classifier: $h_{\text{Bayes}} = \arg \min_h R(h)$

- For 0-1 loss, the Bayes optimal classifier is

$$h_{\text{Bayes}} = \arg \max_y p(y|x) = \arg \max_y p(x|y)p(y)$$



- Optimal (within hypothesis class) classifier $h^* = \arg \min_{h \in \mathcal{H}} R(h)$
- ERM Classifier $h_{\text{ERM}} = \arg \min_{h \in \mathcal{H}} \hat{R}(h)$
- My classifier: $\hat{h} = \text{My_Learning_Algorithm}(\text{Data})$

Risk Decomposition

$$\begin{aligned} & R(\hat{h}) - R(h_{\text{Bayes}}) \\ = & R(\hat{h}) - \hat{R}(\hat{h}) + \hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}}) + \hat{R}(h_{\text{ERM}}) \\ & - \hat{R}(h^*) + \hat{R}(h^*) - R(h^*) + R(h^*) - R(h_{\text{Bayes}}) \\ \leq & \hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}}) + R(h^*) - R(h_{\text{Bayes}}) + R(\hat{h}) - \hat{R}(\hat{h}) + \hat{R}(h^*) - R(h^*) \end{aligned}$$

$$\begin{aligned} & \mathbb{E}[R(\hat{h})] - R(h_{\text{Bayes}}) \\ \leq & \mathbb{E}[\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})] + R(h^*) - R(h_{\text{Bayes}}) + \mathbb{E}[R(\hat{h}) - \hat{R}(\hat{h})] \end{aligned}$$

Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

	Optimization error	Generalization Error	Approximation Error
Definition	$\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})$	$R(\hat{h}) - \hat{R}(\hat{h})$	$R(h^*) - R(h_{\text{Bayes}})$
Challenges	<ul style="list-style-type: none"> Finding ERM for some loss functions is NP-Hard. Efficiency isn't enough. Need to be scalable. 	<ul style="list-style-type: none"> We do not observe Risk! Don't have infinite data. Large generalization error \Leftrightarrow Overfitting 	<ul style="list-style-type: none"> Don't know data distribution. No knowledge of Bayes optimal classifier. Large approx. error \Leftrightarrow Underfitting!
What we have learned to address these challenges?	"Just-relax" Surrogate loss, Gradient Descent, SGD	Holdout, Cross-Validation Regularization Statistical learning theory	Better features More flexible decision boundaries Better probabilistic models But how to minimize approx. error automatically?

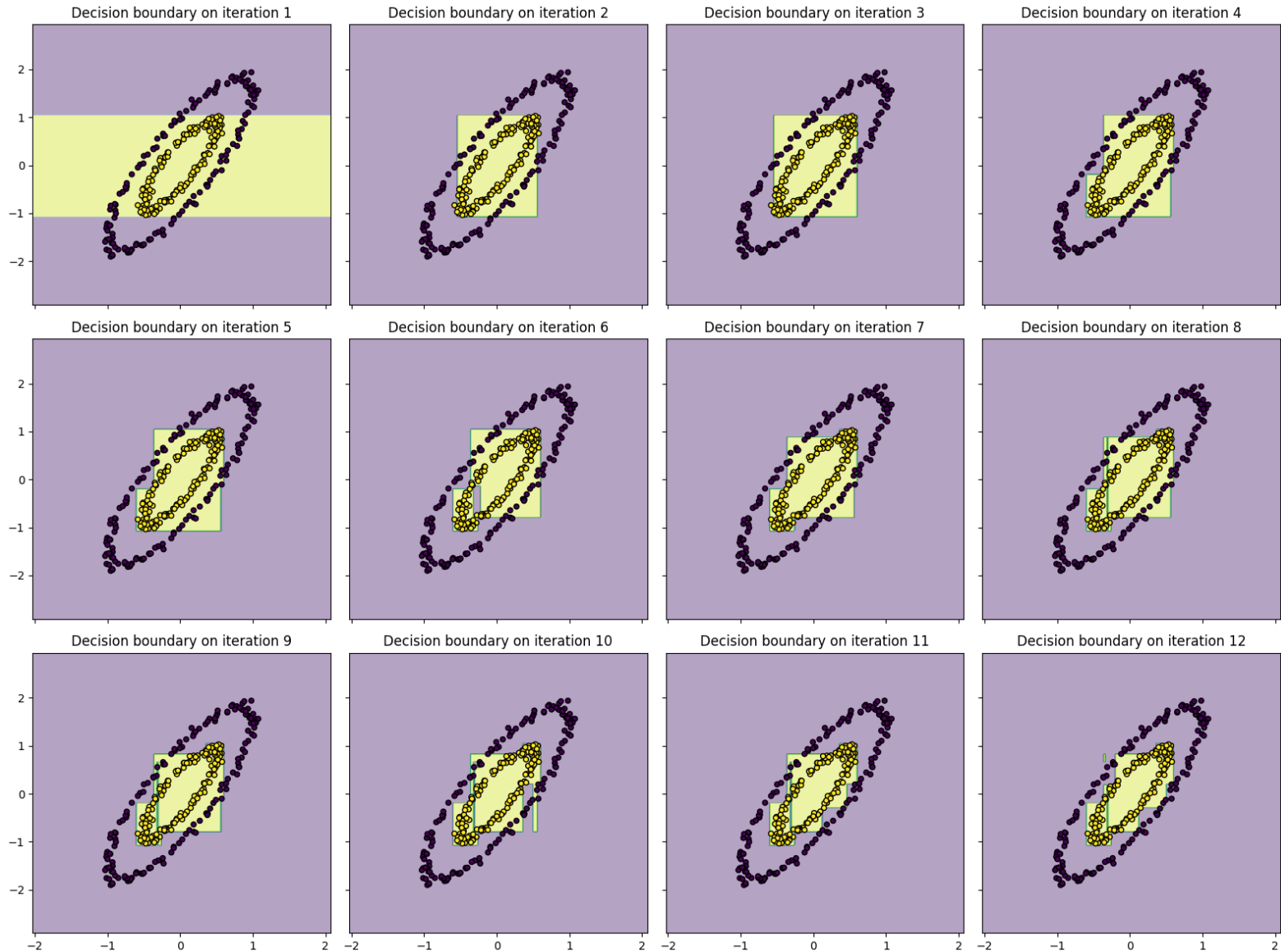
Often there is a tradeoff.

More **flexible** hypothesis class => **smaller approximation error**
 but **larger generalization error** (more prone to overfitting)
 and sometimes **harder optimization**

Three main approaches for expanding the hypothesis class (systematically minimizing the approx. error)

- Kernel methods (lift features to higher-dimensional space)
 - e.g., adding polynomial expansion, add interaction terms
 - Other nonlinear transformation of the original features
- Boosting and Bagging (Ensemble learning)
 - Combine many weak learners (e.g., decision trees with depth 3) into a strong learner (e.g., by majority voting...)
- Deep Learning
 - Train large neural networks using SGD
 - Learn feature representation and classification jointly.

Illustration of a boosting algorithm



Next Lecture

- Decision trees and Boosting
- Feature expansion and kernels