



Machine learning: lecture 13

Tommi S. Jaakkola
MIT CSAIL
tommi@csail.mit.edu



Topics

- Sequential prediction and description length
 - minimum description length principle (MDL), asymptotic expansion
- Probability models and structure
 - mixing, mixtures, and the EM-algorithm



Fair sequential prediction

- We wish to predict (generate) labels y_1, \dots, y_n associated with input examples x_1, \dots, x_n .
- In a fair sequential prediction we predict each label based on the corresponding input and preceding labels and examples

y_1 is predicted based on x_1 alone

y_2 is predicted based on x_2 and $D_1 = \{(x_1, y_1)\}$

y_3 is predicted based on x_3 and $D_2 = \{(x_1, y_1), (x_2, y_2)\}$

...



Fair sequential prediction

- We wish to predict (generate) labels y_1, \dots, y_n associated with input examples x_1, \dots, x_n .
- In a fair sequential prediction we predict each label based on the corresponding input and preceding labels and examples

y_1 is predicted based on x_1 alone

$$P(y_1|x_1) = \int P(y_1|x_1, \theta)P(\theta)d\theta$$

y_2 is predicted based on x_2 and $D_1 = \{(x_1, y_1)\}$

$$P(y_2|x_2, D_1) = \int P(y_2|x_2, \theta)P(\theta|D_1)d\theta$$

y_3 is predicted based on x_3 and $D_2 = \{(x_1, y_1), (x_2, y_2)\}$

$$P(y_3|x_3, D_2) = \int P(y_3|x_3, \theta)P(\theta|D_2)d\theta$$



Fair sequential prediction

- Our fair sequential prediction method defines a valid probability distribution over the training labels given the examples:

$$P(y_1|x_1)P(y_2|x_2, D_1)P(y_3|x_3, D_2) \cdots P(y_n|x_n, D_{n-1})$$



Fair sequential prediction

- Our fair sequential prediction method defines a valid probability distribution over the training labels given the examples:

$$P(y_1|x_1)P(y_2|x_2, D_1)P(y_3|x_3, D_2) \cdots P(y_n|x_n, D_{n-1})$$

This distribution does not depend on the order in which we processed the examples and, in fact, is equal to *Bayesian marginal likelihood*:

$$\int P(y_1|x_1, \theta) \cdots P(y_n|x_n, \theta)P(\theta)d\theta$$



Fair sequential prediction

- Our fair sequential prediction method defines a valid probability distribution over the training labels given the examples:

$$P(y_1|x_1)P(y_2|x_2, D_1)P(y_3|x_3, D_2) \cdots P(y_n|x_n, D_{n-1})$$

How well this distribution predicts the training labels depends on the "complexity" of the model $P(y|x, \theta)$, $\theta \in \Theta$ and how appropriate the prior $P(\theta)$ is.



Fair sequential prediction

- Our fair sequential prediction method defines a valid probability distribution over the training labels given the examples:

$$P(y_1|x_1)P(y_2|x_2, D_1)P(y_3|x_3, D_2) \cdots P(y_n|x_n, D_{n-1})$$

How well this distribution predicts the training labels depends on the "complexity" of the model $P(y|x, \theta)$, $\theta \in \Theta$ and how appropriate the prior $P(\theta)$ is.

- if the model is *too flexible*: the posterior $P(\theta|D_{i-1})$ requires many training examples before it focuses on useful parameter values



Fair sequential prediction

- Our fair sequential prediction method defines a valid probability distribution over the training labels given the examples:

$$P(y_1|x_1)P(y_2|x_2, D_1)P(y_3|x_3, D_2) \cdots P(y_n|x_n, D_{n-1})$$

How well this distribution predicts the training labels depends on the "complexity" of the model $P(y|x, \theta)$, $\theta \in \Theta$ and how appropriate the prior $P(\theta)$ is.

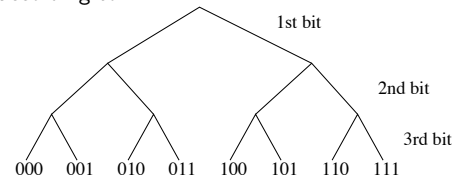
- if the model is *too flexible*: the posterior $P(\theta|D_{i-1})$ requires many training examples before it focuses on useful parameter values
- if the model is *too simple*: the posterior concentrates quickly but the predictions remain poor



Description length and probabilities

- If we can predict the training labels with high probability, then we can communicate them effectively (with few bits)
- It takes $-\log_2 P(y_1, \dots, y_n)$ bits to communicate y_1, \dots, y_n according to distribution P .

Example: suppose each configuration (y_1, y_2, y_3) is equally likely according to P



We need $-\log_2 P(y_1, y_2, y_3) = -\log_2(1/8) = 3$ bits to describe each y .



Description length and model selection

- We need

$$-\log_2 \int P(y_1|x_1, \theta) \cdots P(y_n|x_n, \theta) P(\theta) d\theta$$

bits to communicate labels y_1, \dots, y_n given examples x_1, \dots, x_n with a model $P(y|x, \theta)$, $\theta \in \Theta$ and prior $P(\theta)$.

- **Minimum description length (MDL) principle:** We select the model+prior combination that requires the fewest number of bits (maximizes the Bayesian marginal likelihood)



Asymptotic approximation

- For large n we can use the following asymptotic expansion:

$$\underbrace{\sum_{i=1}^n (-\log_2 P(y_i|x_i, \hat{\theta}))}_{\approx \text{DL of data}} + \underbrace{\frac{d}{2} \log_2(n)}_{\approx \text{DL of model}}$$

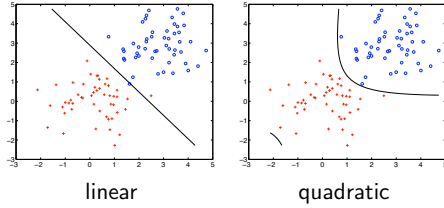
where $\hat{\theta}$ is the maximum likelihood setting of the parameters and d is the effective number of parameters in the model.

- The negative of this is also known as the *Bayesian information criterion* or BIC for short.



Description length: example

- Example: polynomial logistic regression, $n = 100$

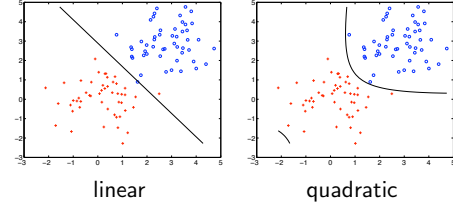


$$\sum_{i=1}^n \left(-\log_2 P(y_i | \mathbf{x}_i, \hat{\theta}) \right) + \frac{d}{2} \log_2(n)$$



Description length: example

- Example: polynomial logistic regression, $n = 100$



$$\sum_{i=1}^n \left(-\log_2 P(y_i | \mathbf{x}_i, \hat{\theta}) \right) + \frac{d}{2} \log_2(n)$$

degree	# param	DL(data)	DL(model)	MDL score
1	3	5.6 bits	9.9 bits	15.5 bits
2	6	2.4 bits	19.9 bits	22.3 bits



Topics

- Sequential prediction and description length
 - minimum description length principle (MDL), asymptotic expansion
- Probability models and structure
 - mixing, mixtures, and the EM-algorithm



What are we missing?

- So far we have solved simple binary classification problems, predicting y given \mathbf{x} , by estimating
 - discriminant functions (e.g., SVMs and boosting)
 - conditional probabilities (e.g., logistic regression)
- What about problems where
 - we have to predict multiple inter-connected labels for each input example (e.g., a set of topics for a document)
 - we have to switch between classifiers in the course of making predictions (e.g., changes in market conditions)
 - the inputs are incomplete in the sense that some of the components are missing (e.g., patient records)
 - the input examples come in different potentially unobserved types (e.g., mixed populations)



Structure and mixtures

- If we wish to take into account the fact that there are different underlying types of examples, we have to first identify them
- We can hypothesize that
 1. there are m underlying types $y = 1, \dots, m$
 2. each type y occurs with frequency $P(y)$
 3. examples of type y are governed by distribution $p(\mathbf{x}|y)$
- According to this model each observed example \mathbf{x} can be assumed to have come from a "mixture distribution":

$$p(\mathbf{x}) = \sum_{j=1}^m P(y = j) p(\mathbf{x}|y = j)$$

- We need to parameterize and estimate such models from samples $\mathbf{x}_1, \dots, \mathbf{x}_n$

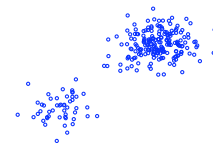


Mixture densities

- A mixture of Gaussians model

$$p(\mathbf{x}|\theta) = \sum_{i=1}^m p_j p(\mathbf{x}|\mu_j, \Sigma_j)$$

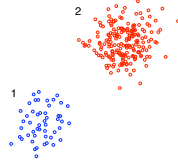
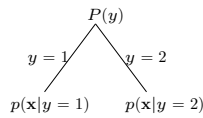
where $\theta = \{p_1, \dots, p_m, \mu_1, \dots, \mu_m, \Sigma_1, \dots, \Sigma_m\}$ contains all the parameters of the mixture model. $\{p_j\}$ are known as *mixing proportions or coefficients*.





Mixture densities

- Data generation process:



$$p(\mathbf{x}|\theta) = \sum_{j=1,2} p_j \cdot p(\mathbf{x}|\mu_j, \Sigma_j) \quad (\text{mixture of Gaussians})$$

- Any data point \mathbf{x} could have been generated in two ways; the component responsible for generating \mathbf{x} needs to be *inferred*.

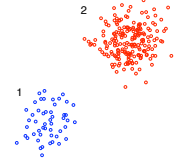


Mixture density estimation

- Suppose we want to estimate a two component mixture of Gaussians model.

$$p(\mathbf{x}|\theta) = p_1 p(\mathbf{x}|\mu_1, \Sigma_1) + p_2 p(\mathbf{x}|\mu_2, \Sigma_2)$$

- If each example \mathbf{x}_i in the training set were labeled $y_i = 1, 2$ according to which mixture component (1 or 2) had generated it, then the estimation would be easy.

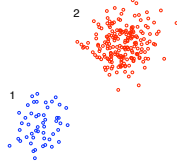


Labeled examples \Rightarrow no credit assignment problem



Mixture density estimation

- When examples are labeled, we can estimate each Gaussian independently
- Let $\delta(j|i)$ be an indicator function of whether example i is labeled j . Then for each $j = 1, 2$



$$\hat{p}_j \leftarrow \frac{\hat{n}_j}{n}, \quad \text{where } \hat{n}_j = \sum_{i=1}^n \delta(j|i)$$

$$\hat{\mu}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^n \delta(j|i) \mathbf{x}_i$$

$$\hat{\Sigma}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^n \delta(j|i) (\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T$$



Mixture density estimation: credit assignment

- Of course we don't have such labels ... but we can guess what the labels might be based on our current mixture distribution
- We can, for example, evaluate the posterior probability that an observed \mathbf{x} was generated from the first mixture component

$$\begin{aligned} P(y=1|\mathbf{x}, \theta) &= \frac{P(y=1) \cdot p(\mathbf{x}|y=1)}{\sum_{j=1,2} P(y=j) \cdot p(\mathbf{x}|y=j)} \\ &= \frac{p_1 p(\mathbf{x}|\mu_1, \Sigma_1)}{\sum_{j=1,2} p_j p(\mathbf{x}|\mu_j, \Sigma_j)} \end{aligned}$$

This solves the *credit assignment* problem

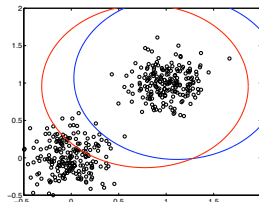


Mixture density estimation: credit assignment

- We get soft labels or posterior probabilities of which Gaussian generated which example:

$$\hat{p}(j|i) \leftarrow P(y_i = j|\mathbf{x}_i, \theta)$$

where $\sum_{j=1,2} \hat{p}(j|i) = 1$ for all $i = 1, \dots, n$.

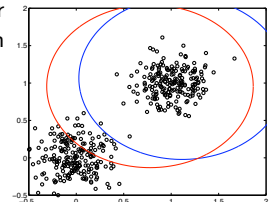


Mixture density estimation: credit assignment

- We get soft labels or posterior probabilities of which Gaussian generated which example:

$$\hat{p}(j|i) \leftarrow P(y_i = j|\mathbf{x}_i, \theta)$$

where $\sum_{j=1,2} \hat{p}(j|i) = 1$ for all $i = 1, \dots, n$.



- When the Gaussians are almost identical (as in the figure), $\hat{p}(1|i) \approx \hat{p}(2|i)$ for almost any available point \mathbf{x}_i . Even slight differences can help us determine how we should modify the Gaussians.



The EM algorithm: iteration k

E-step: softly assign examples to mixture components

$$\hat{p}(j|i) \leftarrow P(y_i = j | \mathbf{x}_i, \theta^{(k)}), \text{ for all } j = 1, 2 \text{ and } i = 1, \dots, n$$

M-step: estimate new mixture parameters $\theta^{(k+1)}$ based on the soft assignments (can be done separately for the two Gaussians)

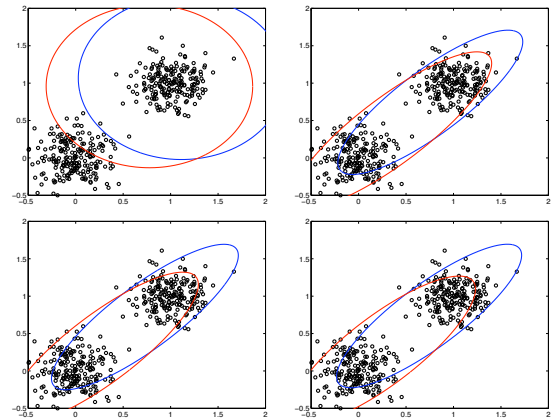
$$\hat{p}_j \leftarrow \frac{\hat{n}_j}{n}, \text{ where } \hat{n}_j = \sum_{i=1}^n \hat{p}(j|i)$$

$$\hat{\mu}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^n \hat{p}(j|i) \mathbf{x}_i$$

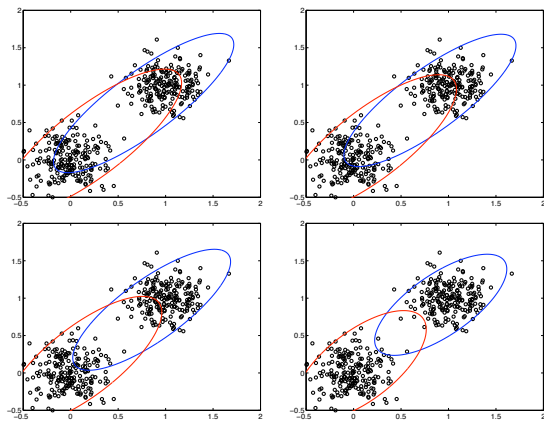
$$\hat{\Sigma}_j \leftarrow \frac{1}{\hat{n}_j} \sum_{i=1}^n \hat{p}(j|i) (\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T$$



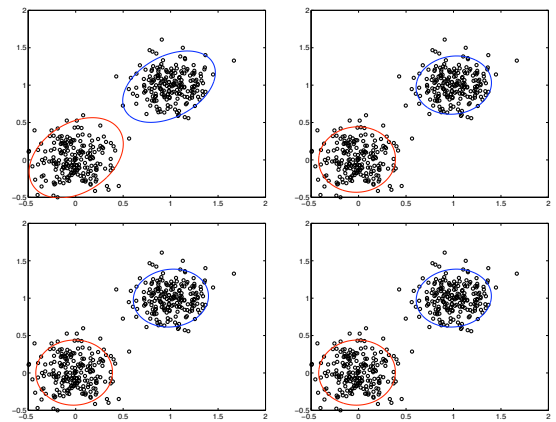
Mixture density estimation: example



Mixture density estimation



Mixture density estimation



The EM-algorithm

- Each iteration of the EM-algorithm *monotonically* increases the (log-)likelihood of the n training examples $\mathbf{x}_1, \dots, \mathbf{x}_n$:

$$\log p(\text{data} | \theta^{(k)}) = \sum_{i=1}^n \log \left(\overbrace{p_1 p(\mathbf{x}_i | \mu_1, \Sigma_1) + p_2 p(\mathbf{x}_i | \mu_2, \Sigma_2)}^{p(\mathbf{x}_i | \theta^{(k)})} \right)$$

where $\theta^{(k)} = \{p_1, p_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$ specifies the parameters of the mixture model at the k^{th} iteration.

