IML Tutorial
Generative Models

Jakob Jakob [1]

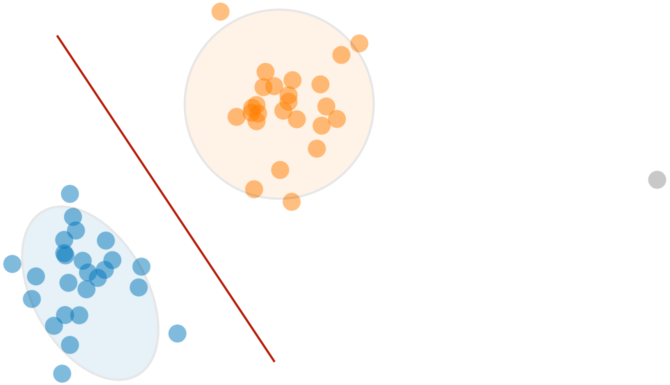
[1]ETH Zurich

Motivation

- Discriminative models
  - estimate directly $P(y|\mathbf{x})$ and do not consider $P(\mathbf{x})$
  - predict new $\mathbf{x}'$ based on seen/learned $\mathbf{x}_i$
  - predict an outlier $\mathbf{x}_o$ overconfidently

- Generative models
  - compute $P(y|\mathbf{x})$ after estimating $P(y, \mathbf{x})$ by considering $P(\mathbf{x})$
  - predict new $\mathbf{x}'$ based on $P(\mathbf{x})$ by seeing $\mathbf{x}_i$
  - are able to detect outliers

Motivation



Dicriminative models learn decision boundaries (red) and generative models learn class-conditional distributions (blue and orange blobs). There is also an outlier $\mathbf{x}_o$ (gray).

Generative Modeling

Estimating the joint distribution $P(y, \mathbf{x})$ directly is often not tractable (not enough data points).

Alternative approach

- Estimate prior on labels $P(y)$
- Based on data derive conditional distribution $P(\mathbf{x}|y)$
- Obtain posterior

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|y)P(y)}{\sum_{y'} P(\mathbf{x}|y')P(y')} = \frac{1}{Z}P(\mathbf{x}|y)P(y)$$

- Note: Computing Z is not necessary for predicting y from $P(y|\mathbf{x})$.
- If closed-form of $P(y|\mathbf{x})$ is not available choose a **conjugate prior**: $P(y|\mathbf{x})$ and $P(y)$ have the same algebraic form, e.g. $\mathcal{N}(\mu, \sigma)$.

When to use which approach?

- If the model is well-specified (you managed to build $P(\mathbf{x})$ correctly), generative modeling yields better results

- Else (much more often the case), it depends on how much data is available
  - small amount of data $\implies$ generative
  - more data $\implies$ discriminative

- More info [1]

[1] Ng, A.Y. and Jordan, M.I., 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Advances in neural information processing systems (pp. 841-848).

Exercise (former exam question)

You trained a generative model and want to predict a label $y \in \{0, 1\}$ for a new data point $\mathbf{x}$. Your model tells you:

- $P(Y = 1) = P(Y = 0) = 0.5$
- $P(\mathbf{X}|Y = 0) = 0.02$
- $P(\mathbf{X}|Y = 1) = 0.03$

To predict a label, you should compute $P(Y = 0|\mathbf{X})$. What is the result?

1. 0.01
2. 0.2
3. 0.4
4. Undetermined as we need to know $P(\mathbf{X})$

Exercise (former exam question)

Solution: $P(Y=0|\mathbf{X}) = 0.4$

$$P(Y=0|\mathbf{X}) = \frac{P(\mathbf{X}|Y=0)P(Y=0)}{P(\mathbf{X})}$$

$$P(Y=0|\mathbf{X}) = \frac{P(\mathbf{X}|Y=0)P(Y=0)}{P(\mathbf{X}|Y=0)P(Y=0) + P(\mathbf{X}|Y=1)P(Y=1)}$$

$$P(Y=0|\mathbf{X}) = \frac{0.02 \cdot 0.5}{0.02 \cdot 0.5 + 0.03 \cdot 0.5} = \frac{0.02}{0.02 + 0.03} = 0.4$$

Naive Bayes — A Generative Model

Model class labels as generated from categorical variable

$$P(Y = y) = p_y \qquad y \in \{1, \dots, m\}$$

Simplification (**naive** assumption): conditional independence

$$P(\mathbf{X}|Y) = \prod_{i=1}^{d} P(X_i|Y)$$

$$P(X_1 = x_1, \dots, X_d = x_d|Y = y) = \prod_{i=1}^{d} P(X_i = x_i|Y = y)$$

Given Y each X_i is independent and $P(X_i|Y) = ?, p_y = ?$ chosen by inspecting the data.

Naive Bayes — p_y

- Categorical distribution (class labels):
  - $P(Y = y) = p_y \iff P(y|\mathbf{p}) = \prod_{j=1}^{m} p_j^{[y=j]}, \quad \sum_{j=1}^{m} p_j = 1$
  - over n samples $D = \{(\mathbf{x_1}, y_1), \dots, (\mathbf{x_n}, y_n)\}$:

$$P(\mathbf{y}|\mathbf{p}) = P(y_1, \dots, y_n | p_1, \dots, p_m) = \prod_{i=1}^{n} \prod_{j=1}^{m} p_j^{[y_i=j]}$$

  - MLE[2] over n samples ($\mathbf{y} = (y_1, \dots, y_n)$) to estimate p_j

$$\frac{\partial P(\mathbf{y})}{\partial p_j} = 0 \iff \frac{\partial(\log P(\mathbf{y}))}{\partial p_j} = 0 \implies \hat{p}_y = \frac{\text{Count}(Y = y)}{n}$$

[2] Maximum Likelihood Estimation

Naive Bayes — p_y - Example from lecture

- Binary case $y = \{0, 1\}$ - Bernoulli distribution:
  - $\sum_{j=0}^{m-1} p_j = 1 \implies P(y=1) = p, P(y=0) = 1-p$

$$P(y) = \prod_{j=0}^{m-1} p_j^{[y=j]} \implies P(y) = p^y (1-p)^{1-y}$$

  - over n samples $D = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)\}$:

$$P(\mathbf{y}|p) = \prod_{i=1}^{n} p^{y_i}(1-p)^{1-y_i}$$

  - MLE over n samples to estimate p

$$\frac{\partial \log P(\mathbf{y})}{\partial p} = 0 \quad \implies \quad \hat{p} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Naive Bayes — $P(X_i|Y)$

- Continuous $X_i \in \mathbb{R}$
  - Gaussian Naive Bayes (GNB) with parameters $\mu_{y,i}, \sigma^2_{y,i}$ (lecture):

$$P(x_i|y) = \mathcal{N}(x_i|\mu_{y,i}, \sigma^2_{y,i})$$

  - Poisson Naive Bayes with parameters $\lambda_{y,i}$ (HW6):

$$P(x_i|y) = e^{\lambda_{y,i}} \frac{\lambda_{y,i}^{x_i}}{x_i!}$$

- Discrete $X_i \in \mathbb{N}$
  - Categorical Naive Bayes with parameters $\theta^{(i)}_{x_i|y}$ (lecture):

$$P(x_i|y) = \theta^{(i)}_{x_i|y}$$

Estimate the parameters of the distribution by MLE!

Gaussian Naive Bayes — $P(X_i|Y)$ - Example

MLE of $\mu_{y,i}$:

$$P(\mathbf{x}|y) = \prod_{i=1}^{d} \frac{1}{\sigma_{y,i}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i - \mu_{y,i}}{\sigma_{y,i}}\right)^2}$$

$$P(\mathbf{x}_1, \ldots, \mathbf{x}_n|y) = \prod_{j:y_j=y} \prod_{i=1}^{d} \frac{1}{\sigma_{y,i}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_j, i - \mu_{y,i}}{\sigma_{y,i}}\right)^2}$$

$$\frac{\partial \log(P(\mathbf{x}_1, \ldots, \mathbf{x}_n|y))}{\partial \mu_{y,i}} = \sum_{j:y_j=y} (x_i - \mu_{y,i}) = 0$$

$$\hat{\mu}_{y,i} = \frac{1}{|j : y_j = y|} \sum_{j:y_j=y} x_i$$

Gaussian Naive Bayes - Prediction

Now, we have

- MLE for class prior: $\hat{P}(Y=y) = \hat{p}_y = \frac{\text{Count}(Y=y)}{n} = \frac{|Y=y|}{n}$

- MLE for feature distr.: $\hat{P}(x_i|y) = \mathcal{N}(x_i; \hat{\mu}_{y,i}, \hat{\sigma}^2_{y,i})$

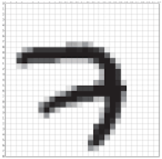
$$\hat{\mu}_{y,i} = \frac{1}{|Y=y|} \sum_{j:y_j=y} x_{j,i} \quad \hat{\sigma}^2_{y,i} = \frac{1}{|Y=y|} \sum_{j:y_j=y} (x_{j,i} - \hat{\mu}_{y,i})^2$$

Prediction: $y = \arg\max_{y'} \hat{P}(y'|\mathbf{x}) = \arg\max_{y'} \hat{P}(y') \prod_{i=1}^{d} \hat{P}(x_i|y')$

Gaussian Naive Bayes — Cond. Indep. on MNIST

The MNIST data set has n, 28×28 ($= 784$) dimensional images and 10 labels 0 to 9.
Formally, let $\mathcal{Y} = \{0, ..., 9\}$ be the set of labels and $\mathcal{X} = \mathbb{R}^{784}$ a 784-dim. feature space, resulting in

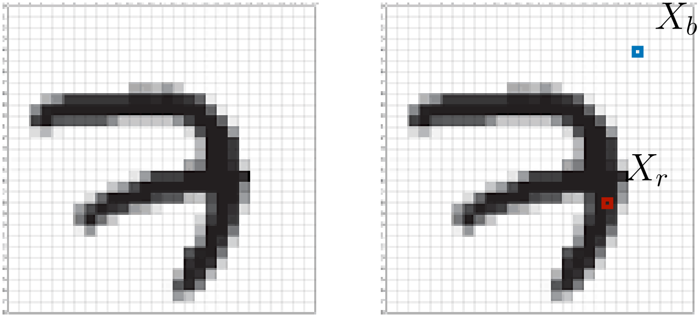
$$D = \{(\mathbf{x}_k, y_k) \in \mathcal{X} \times \mathcal{Y} \,|\, k = 1, \ldots, n\} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$$



MNIST sample with label 7

Gaussian Naive Bayes — Cond. Indep. on MNIST

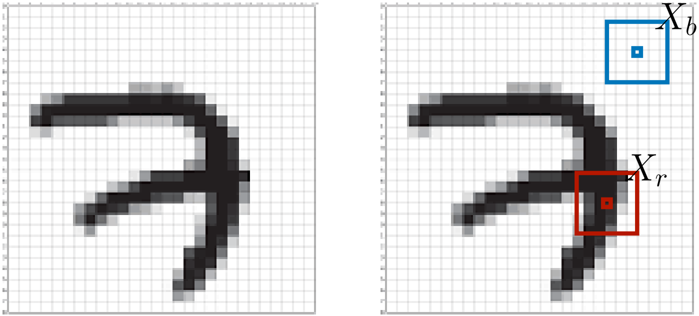
In image $\mathbf{x}_k$ each pixel i corresponds to $X_i = x_i$.
Exercise: Having multiple samples with label 7 for a GNB model, what is the difference between $P(X_b|Y)$ and $P(X_r|Y)$? Further, what does the model not capture? Why?



Gaussian Naive Bayes — Cond. Indep. on MNIST

Recall for GNB: $P(X_i = x_i|Y) = \mathcal{N}(x_i|\mu_{y,i}, \sigma^2_{y,i})$

$\sigma^2_{y,b} \sim 0$ and $\sigma^2_{y,r} > 0$. GNB misses neighborhood information, because of cond. independence!



Gaussian Bayes Classifier (GBC)

GBC takes correlation of features into account (**not** cond. indep.)!

$$P(\mathbf{x}|y) = \mathcal{N}(\mathbf{x}; \mu_y, \Sigma_y)$$

where Σ_y is a **non-diagonal** matrix and MLE yields $\hat{\mu}_y, \hat{\Sigma}_y$!
For binary classification ($y \in \{+1, -1\}$): $y = \text{sign}(f(\mathbf{x}))$

$$f(\mathbf{x}) = \log\left(\frac{P(Y=1|\mathbf{x})}{P(Y=-1|\mathbf{x})}\right) = \log\left(\frac{p}{1-p}\right) + \frac{1}{2}\log\left(\frac{|\hat{\Sigma}_-|}{|\hat{\Sigma}_+|}\right)$$
$$+ \frac{1}{2}(\mathbf{x} - \hat{\mu}_-)^\top \hat{\Sigma}_-^{-1}(\mathbf{x} - \hat{\mu}_-) - \frac{1}{2}(\mathbf{x} - \hat{\mu}_+)^\top \hat{\Sigma}_+^{-1}(\mathbf{x} - \hat{\mu}_+)$$

Linear discriminant analysis (LDA)

- Special case of Gaussian Bayes Classifiers
  - Same co-variance matrix across classes (for binary: $\hat{\Sigma}_- = \hat{\Sigma}_+ = \hat{\Sigma}$)
  - it's called linear because $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$ is linear in $\mathbf{x}$, where for the binary case and $p = 0.5$ (Fisher's LDA):

$$\mathbf{w} = \hat{\Sigma}^{-1}(\hat{\mu}_+ - \hat{\mu}_-) \qquad w_0 = \frac{1}{2}(\hat{\mu}_-^\top \hat{\Sigma}^{-1}\hat{\mu}_- - \hat{\mu}_+^\top \hat{\Sigma}^{-1}\hat{\mu}_+)$$

Quadratic discriminant analysis (QDA)

- Special case of Gaussian Bayes Classifiers
  - Co-variance matrix across classes not *necessarily* equal $\hat{\Sigma}_- \neq \hat{\Sigma}_+$
  - $f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} + \mathbf{w}^\top \mathbf{x} + w_0$ is quadratic in $\mathbf{x}$, where for the binary case and $p = 0.5$:

$$
\begin{aligned}
A &= \frac{1}{2}(\hat{\Sigma}_-^{-1} - \hat{\Sigma}_+^{-1}) \\
\mathbf{w} &= (\hat{\Sigma}_+^{-1}\hat{\mu}_+ - \hat{\Sigma}_-^{-1}\hat{\mu}_-) \\
w_0 &= \frac{1}{2}\log\left(\frac{|\hat{\Sigma}_-|}{|\hat{\Sigma}_+|}\right) + \frac{1}{2}(\hat{\mu}_-^\top \hat{\Sigma}^{-1}\hat{\mu}_- - \hat{\mu}_+^\top \hat{\Sigma}^{-1}\hat{\mu}_+)
\end{aligned}
$$

Regularization

- MLE of distribution parameters is prone to overfitting. Options to prevent that
  - Restrict model class (e.g GNB)
  - Priors
    - $P(Y = 1) = \theta$
    - Compute posterior on previous data $P(\theta | y_1, \ldots, y_n)$
    - Conjugate priors - prior and posterior are in the same family
    - Prior: Beta$(\theta, \alpha_+, \alpha_-)$
    - Observe additional data (n_+, n_-)
    - Posterior: Beta$(\theta, \alpha_+ + n_+, \alpha_- + n_-)$

END OF PRESENTATION