Computational Intelligence: Methods and Applications

Lecture 13

Bayesian risks & naive Bayes approach

Włodzisław Duch
SCE, NTU, Singapore
Google: Duch

Bayesian future

The February 2004 issue of MIT Technology magazine showcases Bayesian Machine Learning as one of the 10 emerging technologies that will change your world!

Intel wants computers to be more proactive, learn from their experiences with users and the world around them.

Intel Open Source Machine Learning library OpenML, and Audio-Visual Speech Recognition
http://www.intel.com/technology/computing/applications/avcsr.htm

Now are part of: Open Source Computer Vision Library
http://www.intel.com/technology/computing/opencv/index.htm
http://sourceforge.net/projects/opencvlibrary
http://opencvlibrary.sourceforge.net/

Total risks

Confusion matrix:

$$P\left(C(\mathbf{X})=\omega_k \mid \hat{C}(\mathbf{X})=\omega_l\right)=\mathbf{P}=\begin{bmatrix} P_{11} & P_{12} & \dots & P_{1K} & P_{1K+1} \\ P_{21} & P_{22} & \dots & P_{2K} & P_{2K+1} \\ \dots & \dots & \dots & \dots & \dots \\ P_{K1} & P_{K2} & \dots & P_{KK} & P_{KK+1} \end{bmatrix}$$

Total risk of the decision procedure $\hat{C}$:

$$R\left(\hat{C}\right)=\sum_{k=1}^{K} P(\omega_k)\, R\left(\hat{C},\omega_k\right)=\sum_{k=1}^{K}\sum_{l=1}^{K+1} P(\omega_k)\,\lambda_{kl} P_{kl}$$

If all P_{kl}=0 for $k\neq l$ the risk is zero (assuming λ_{ii}=0).

Minimizing risk means that we try to find decision procedure $\hat{C}$ that minimizes "important" errors. For example, if classes ω_k, k=1,..10, are degrees of severity of heart problems, than mixing $\lambda_{12} < \lambda_{19}$

Bayesian risk

Select the class corresponding to minimum conditional risk:

$$g_k(\mathbf{X})=\sum_{i=1}^{K+1}\lambda_{ik} P(\mathbf{X}\mid\omega_i) P(\omega_i) < g_j(\mathbf{X})=\sum_{i=1}^{K+1}\lambda_{ij} P(\mathbf{X}\mid\omega_i) P(\omega_i)$$

This is equivalent to the following decision procedure $\hat{C}$:

$$R(\omega_l\mid\mathbf{X})=\sum_{i=1}^{K+1}\lambda_{il} P(\omega_i\mid\mathbf{X})$$

Define conditional risk of performing action ω_l given X

$$\hat{C}(\mathbf{X})=\begin{cases}\omega_k & \text{for} \quad k=\arg\min\limits_{l\le K} R(\omega_l\mid\mathbf{X}) \\ \omega_D & \text{for} \quad \text{otherwise}\end{cases}$$

d is the value of the risk of not taking any action.
Discrimination functions $g_k(\mathbf{X})$ provide minimal risk decision boundaries if good approximation to $P(\mathbf{X}|\omega)$ probabilities are found.

Discriminating functions

$g_k(\mathbf{X})$ may also be replaced by posterior probabilities; in the feature space area where the lowest risk corresponds to decision ω_i we have $g_i(\mathbf{X}) > g_j(\mathbf{X})$, and $g_i(\mathbf{X}) = g_j(\mathbf{X})$ at the border.

Any monotonic function of $P(\omega_i|\mathbf{X})$ may be used as discriminating f:

$$g_i(\mathbf{X}) = \ln P(\omega_i \mid \mathbf{X}) = \ln P(\mathbf{X} \mid \omega_i) + \ln P(\omega_i) - \ln P(\mathbf{X})$$

$g_i(\mathbf{X}) = P(\omega_i)$

$g_i(\mathbf{X}) = P(\mathbf{X} \mid \omega_i)$

$g_i(\mathbf{X}) = P(\omega_i \mid \mathbf{X})$

$g_i(\mathbf{X}) = -R(\omega_i \mid \mathbf{X})$

Some choices:

- Majority classifier
- maximum likelihood classifier
- MAP, Maximum A Posterior classifier
- minimum risk classifier (best)

Summary

Posterior probabilities is what we want:

$$P(\omega_i \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid \omega_i) P(\omega_i)}{P(\mathbf{X})}$$

Likelihood x Prior / Evidence

Minimization of error may be done in a number of ways, for ex:

$$E = 1 - \sum_{i=1}^{K} P_{ii}; \quad E = \sum_{\mathbf{X}} \sum_{i=1}^{K} \left\| P(\omega_i \mid \mathbf{X}) - C_i(\mathbf{X}) \right\|$$

where $C_i(X) = 1$ if X is from class ω_i, and 0 otherwise.

Minimization of risk takes into account cost of different types of errors: using Bayesian decision procedure minimizes total/conditional risks.

Try to write risk for 2 class 1D problem with $P(\omega_1)=2/3$, $P(\omega_2)=1/3$, $P(X|\omega_1)$=Gauss(0,1), $P(X|\omega_2)$=Gauss(2,2), $\lambda = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}$

1D Gaussians

Simplest application of Bayesian decisions: Gaussian distributions. 2 distributions, 1D, with the same variance, different means:

$$P(X \mid \omega_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-(X-\mu_i)^2 / 2\sigma^2\right)$$

Discriminating function: log posteriors $g(X) = \ln \frac{P(\omega_1 \mid X)}{P(\omega_2 \mid X)}$

$$g(X) = \ln \frac{P(X \mid \omega_1)}{P(X \mid \omega_2)} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

$$= \ln \frac{\exp\left(-(X-\mu_1)^2 / 2\sigma^2\right)}{\exp\left(-(X-\mu_2)^2 / 2\sigma^2\right)} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

1D Gaussians solution

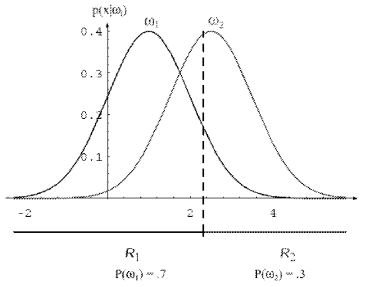
After simplifications:

$$g(X) = \frac{\mu_1 - \mu_2}{\sigma^2} X - \frac{\mu_1^2 - \mu_2^2}{2\sigma^2} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

$g(X)>0$ for class ω_1, $g(X)<0$ for class ω_2

For equal a priori probabilities $g(X_0)=0$ in the middle $X_0 = \frac{\mu_1 + \mu_2}{2}$ between the means of the two distributions. Otherwise it is shifted towards class with smaller prior:

$$X_0 = \frac{\mu_1 + \mu_2}{2} - \frac{\sigma^2}{\mu_1 - \mu_2} \ln \frac{P(\omega_1)}{P(\omega_2)}$$



dD Gaussians

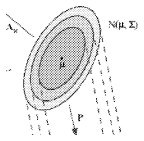
Multivariate Gaussians:

$$P(\mathbf{X} \mid \omega_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\mathbf{X}-\boldsymbol{\mu}_i)\right)$$

Non-diagonal covariance matrix rotates the distributions.

Discriminant functions are quadratic:

$$g_i(\mathbf{X}) = -\frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i) - \frac{1}{2}(\mathbf{X}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\mathbf{X}-\boldsymbol{\mu}_i)$$



$D_M(\mathbf{X},\boldsymbol{\mu})$=const

Discriminant function is thus Mahalanobis distance from the mean + terms that take into account the prior probability and the variance.

Linear classifiers

Identical covariance matrices => linear discrimination function:

$$g_i(\mathbf{X}) = \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1}\mathbf{X} - \frac{1}{2}\boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln P(\omega_i)$$

$$= \sum_{j=1} W_{ij} X_j + W_{i0} = \mathbf{W}_i^{\mathrm{T}}\mathbf{X} + W_{i0}$$

where:

$$\mathbf{W}_i^{\mathrm{T}} = \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1}$$

$$W_{i0} = -\frac{1}{2}\boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln P(\omega_i)$$

This type of linear discrimination functions are used quite frequently, not only in statistical theories but also in neural networks, where:
X are input signals,
W are synaptic weights, and
W_0 is the neuron activation threshold.

Special cases

If identical covariance matrices and identical *a priori* probabilities are assumed then Mahalanobis distance is obtained:

$$g_i(\mathbf{X}) = -(\mathbf{X}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}^{-1}(\mathbf{X}-\boldsymbol{\mu}_i) = -D(\mathbf{X},\boldsymbol{\mu}_i)$$

Note: max $g(\mathrm{X})$ ⇔ min $D(\mathbf{X},\boldsymbol{\mu})$

For diagonal covariance matrices distance becomes:

$$D(\mathbf{X},\boldsymbol{\mu}_i) = \sum_j s_j (X_j - \mu_{ij})^2; \quad s_j = 1/\sigma_j^2$$

Weighted Euclidean distance function from the center of the Gaussian distribution serves as discriminating function.

Bayesian classifier becomes now a nearest prototype classifier:

$$\mathbf{X} \in \omega_k \text{ if } k = \arg\min_i D(\mathbf{X},\boldsymbol{\mu}_i)$$

Illustration

Influence of priors on decision borders:

in 1 D

In 2D

In 3D

Note that decision borders are not between means.

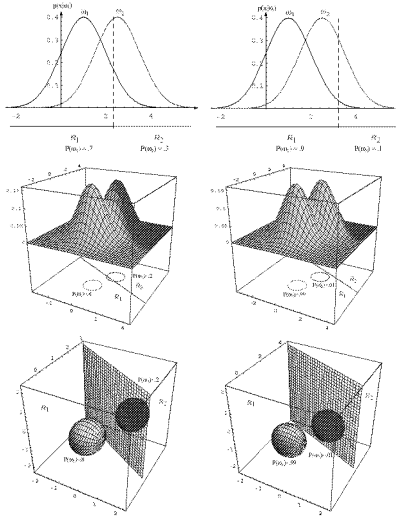


Fig. 2.11, Duda, Hart, Stork

Naive Bayesian classifier

How to estimate probability densities required by Bayesian approach?
$P(X|\omega)=P(X_1,X_2..X_d|\omega)$ in d-dimensions and b values/feature,
requires b^d bins to estimated numerically density of the vectors!
In practice there is never enough data for that!

Assume (naively) that all features are conditionally independent.

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

Instead of multidimensional function the problem is reduced to estimation of d one-dimensional P_i functions.

If the class ω is a disease than symptoms X_1, X_2, depends on the class but not directly on each other, so usually this may work.

It works for Gaussian density distributions for classes; perhaps feature selection has already removed correlated features.

NB

Estimate (learn from data) the posterior probability:

$$P(\omega_k\,|\,\mathbf{X})=\prod_{j=1}^{d}P_i(X_i\,|\,\omega_k)\frac{P(\omega_k)}{P(\mathbf{X})}$$

Use the log-likelihood ratio:

$$\ln\frac{P(\omega_1\,|\,\mathbf{X})}{P(\omega_2\,|\,\mathbf{X})}=\ln\frac{P(\omega_1)}{P(\omega_2)}+\sum_{i=1}^{d}\ln\frac{P_i(X_i\,|\,\omega_1)}{P_i(X_i\,|\,\omega_2)}$$

If the overall result is positive than class ω_1 is more likely.
Each feature has an additive, positive or negative, contribution.

For discrete or symbolic features $P(X_i|\omega)$ is estimated from histograms, for continuous features a single Gaussian or a combination of Gaussians is frequently fit to estimate the density.

NB Iris example

nbc(iris_type) = { prob(iris_type) = {
Iris-setosa : 50, Iris-versicolor: 50, Iris-virginica : 50 };
prob(sepal_length|iris_type) = {
Iris-setosa : N(5.006, 0.124249) [50], (Normal distribution)
Iris-versicolor: N(5.936, 0.266433) [50], (mean, std)
Iris-virginica : N(6.588, 0.404343) [50] };
prob(sepal_width|iris_type) = {
Iris-setosa : N(3.428, 0.14369) [50],
Iris-versicolor: N(2.77, 0.0984694) [50],
Iris-virginica : N(2.974, 0.104004) [50] };
prob(petal_length|iris_type) = {
Iris-setosa : N(1.462, 0.0301592) [50],
Iris-versicolor: N(4.26, 0.220816) [50],
Iris-virginica : N(5.552, 0.304588) [50] };
prob(petal_width|iris_type) = {
Iris-setosa : N(0.246, 0.0111061) [50],
Iris-versicolor: N(1.326, 0.0391061) [50],
Iris-virginica : N(2.026, 0.0754327) [50] };

Use WEKA or applet: http://www.cs.technion.ac.il/~rani/LocBoost/

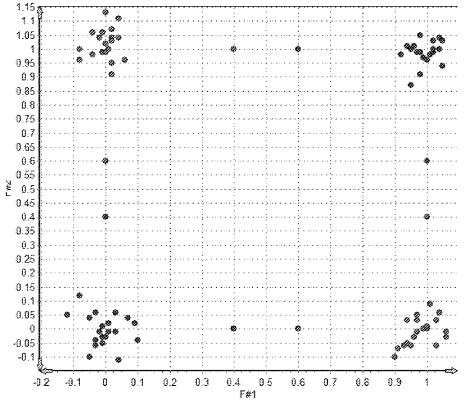
NB on fuzzy XOR example

For demonstrations one may use WEKA, GhostMiner 3
or an applet: http://www.cs.technion.ac.il/~rani/LocBoost/

NB fails completely on such data – why?

$P(X_1,X_2|\omega_k)$ probabilities are needed, but they are more difficult to estimate.

Use local probability around your query point.



NB generalization

If the first-order (independent) feature model is not sufficient for strongly interacting features one can introduce second and third-order dependencies (to find interacting features look at covariance matrix).

$$P(\mathbf{X} \mid \omega_k) = P_i(X_1, X_2 \mid \omega_k) P_i(X_3 \mid \omega_k) P_i(X_4, X_5, X_6 \mid \omega_k) \ldots$$

This leads to graphical causal Bayesian network models.

Surprisingly:

- frequently adding explicit correlations makes things worse, presumably because difficulty in reliable estimation of two or three dimensional densities.
- NB is quite accurate on many data sets, and fairly popular, has many free software packages and numerous applications.
- Local approach: make all estimations only around X!

Literature

Duda, Hart & Stork Chap. 2 has a good introduction to the Bayesian decision theory, including some more advanced sections:

- on min-max, or overall risk minimization which is independent of prior probabilities;
- or Neyman-Pearson criteria that introduce constraints on acceptable risk for selected classes.

More detailed exposition of Bayesian decision and risk theory may be found in chapter 2 of:

S. Theodoridis, K. Koutroumbas, Pattern recognition (2nd ed)

Decision theory is obviously connected with hypothesis testing.

Here chapter 3 of K. Fukunaga, Introduction to statistical pattern recognition (2nd ed, 1990) is the best source, with numerous examples.