

Gaussian Distributions in Machine Learning

Xiaolei Jiang

School of Control and Computer Engineering, North China Electric Power University, Baoding 071003, China

Abstract: To help teaching and learning of machine learning courses, properties and application examples of Gaussian distributions in machine learning are presented, including classification, regression, and approximation. In each typical cases, emphasis is placed on the assumptions and the conclusions related to Gaussian distributions, so that students can make clear that it is the Gaussian distribution that make the somewhat abstract Bayes' theorem more concrete and yields closed-form analytical predictive distributions.

Keywords: Gaussian Distributions; Bayesian Linear Regression; Gaussian Process Regression; Laplace Approximation; Gaussian Expectation Propagation.

1. Introduction

The machine learning course stands as one of cornerstone components within the curriculum of artificial intelligence [1]. The Gaussian distribution is ubiquitous across the field of machine learning, manifesting itself in myriad contexts and applications [2,3]. For example, if a Gaussian noise model is assumed, then the predictive distribution in supervised learning involves a negative exponent which is in a quadratic form. It is beneficial for a student to understand complex models in machine learning if he is proficient in manipulating Gaussian distributions. However, the Gaussian distribution has many properties and it takes time before a student can feel familiar with applications of Gaussian distributions in machine learning. It is the goal of this article to compile a plenty of cases in which the Gaussian distribution plays an important role in solving machine learning problems, serving as reference for teaching and learning of the machine learning course.

The probability density function of a multidimensional Gaussian distribution is

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \quad (1)$$

where $\mathbf{x}$ in $\mathbb{R}^n$, $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix. The inverse matrix of the covariance matrix $\boldsymbol{\Sigma}$ is called the precision matrix and usually denoted as $\boldsymbol{\Lambda}$. Note that the quadratic form appearing in the negative exponent of the density function, so by completing the square we can easily identify the mean vector and precision matrix from the exponent of a Gaussian distribution. The contours of a Gaussian density function are a family of ellipsoids whose center is the mean vector and whose symmetric axes are eigenvectors of the covariance matrix and the distances to the center along the direction of symmetric axes are proportional to the square root of eigenvalues of the covariance matrix.

The rest of this article is organized as follows. After reviewing some properties of Gaussian distributions in Section 2, we collect typical applications of Gaussian distributions in machine learning in the following sections. In Section 3 the Gaussian density functions are supposed to be the class-conditional probability density functions so as to derive the formula of posterior probability. Using the Gaussian distribution, the predictive distributions for Bayesian linear regression and Gaussian process regression are given in Section 4 and Section 5, respectively. Two

approximation methods closely related to Gaussian distributions are presented in Section 6. We draw our conclusion in Section 7.

2. Properties of Gaussian Distributions

There are many properties associated with Gaussian distributions, among them maybe the followings are the most relevant to machine learning. The affine transform of a Gaussian random variable is still Gaussian. Let $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$, then

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T) \quad (2)$$

Gaussian distributions are closed under condition and marginalization. Let

$$p\left(\begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix}\right) \quad (3)$$

then

$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b), \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}), \quad (4)$$

$$p(\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_b|\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{bb}). \quad (5)$$

In addition, the mean vector and precision matrix of $\mathbf{x}_a$ given $\mathbf{x}_b$ can be written in terms of precision matrices and the precision matrix of $\mathbf{x}_b$ can be written in terms of precision matrices. If the prior distribution and likelihood function are Gaussian, then Bayes' theorem implies that the posterior distribution is also Gaussian. Let $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$, $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{S})$, $\mathbf{L} = \mathbf{S}^{-1}$, then

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{S} + \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T) \quad (6)$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x} | (\boldsymbol{\Lambda} + \mathbf{A}^T\mathbf{L}\mathbf{A})^{-1}(\boldsymbol{\Lambda}\boldsymbol{\mu} + \mathbf{A}^T\mathbf{L}(\mathbf{y} - \mathbf{b})), (\boldsymbol{\Lambda} + \mathbf{A}^T\mathbf{L}\mathbf{A})^{-1}). \quad (7)$$

In addition, the precision matrix of $\mathbf{y}$ can be written in terms of precision matrices and the mean vector and covariance matrix of $\mathbf{x}$ given $\mathbf{y}$ can be written in terms of covariance matrices. It is not easy to remember these formulas, and even if students manage to memorize them, they are likely to forget them quickly. Let's look at a simple example. Suppose $p(x) = \mathcal{N}(x|\mu, \lambda^{-1})$, $p(y|x) = \mathcal{N}(y|x, \beta^{-1})$, then $p(y) = \mathcal{N}(y|\mu, \lambda^{-1} + \beta^{-1})$, $p(x|y) = \mathcal{N}(y|(\lambda + \beta)^{-1}(\lambda\mu + \beta y), (\lambda + \beta)^{-1})$, as shown in Figure 1 in which $\mu = 0$, $\lambda = 3$, $\beta = 6$, and the observed $y = 2$.

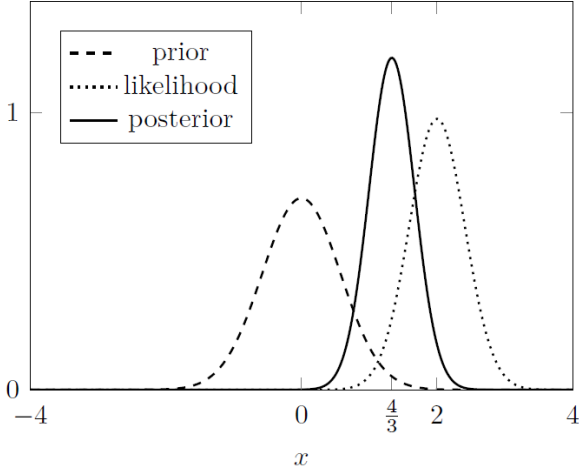


Figure 1. Prior and posterior Gaussian distributions

$$p(C_1|\mathbf{x}) = \frac{p(C_1)\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right)}{p(C_1)\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right) + p(C_2)\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right)}. \quad (8)$$

So Gaussian distributions give us an example illustrating how Bayes' theorem is used for classification problems in supervised learning. Figure 2 illustrates this case for one-

Apart from the formulas mentioned above, Gaussian distributions also exhibit other properties involving convolution and characteristic functions that perhaps have not been found many applications in machine learning.

3. Bayes' Theorem for Classification

For classification problems in supervised learning, posterior probabilities can be obtained in analytical form if the class-conditional probabilities are given, since addition instead of integration is involved when computing the denominator in Bayes' theorem. Suppose there are two classes and the class-conditional probabilities are $p(\mathbf{x}|C_1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $p(\mathbf{x}|C_2) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$, then

dimensional input variables in which $\boldsymbol{\mu}_1 = 0$, $\boldsymbol{\mu}_2 = 2$, $\boldsymbol{\Sigma} = 1$, $p(C_1) = p(C_2) = 0.5$.

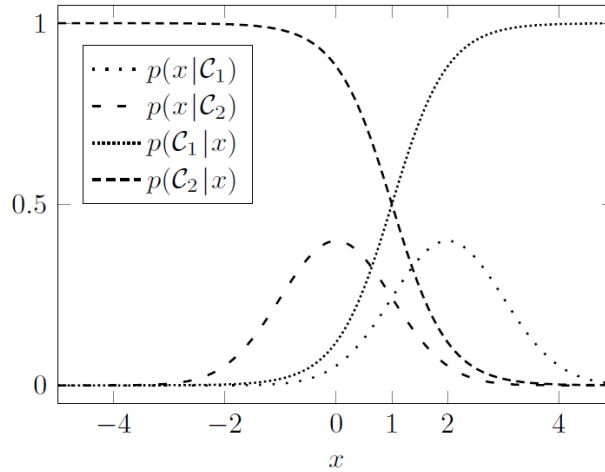


Figure 2. Class-conditional Gaussian densities and posterior Gaussian densities

4. Bayesian Linear Regression

Suppose $t_1, \dots, t_m$ are the target values corresponding to the input vectors $\mathbf{x}_1, \dots, \mathbf{x}_m$, respectively, and our aim is to find the target value s corresponding to the input vector $\mathbf{z}$. The training set $\{t_1, \dots, t_m\}$ can be encapsulated as $\mathbf{t}$. Let the prior for parameters be $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0, \alpha^{-1}\mathbf{I})$, and the

likelihood be $p(\mathbf{t}|\mathbf{w}) = \mathcal{N}(\mathbf{t}|\boldsymbol{\Phi}^T\mathbf{w}, \beta^{-1}\mathbf{I})$, where $\mathbf{w}$ is the parameter vector. Note that $\boldsymbol{\Phi}$ is the design matrix whose columns are feature vectors $\boldsymbol{\phi}(\mathbf{x}_1), \dots, \boldsymbol{\phi}(\mathbf{x}_m)$, where $\boldsymbol{\phi}(\cdot)$ is the feature mapping from input space to feature space. By Equation (7) the posterior distribution of parameter vector $\mathbf{w}$ is given by

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\beta(\alpha\mathbf{I} + \beta\boldsymbol{\Phi}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}\mathbf{t}, (\alpha\mathbf{I} + \beta\boldsymbol{\Phi}\boldsymbol{\Phi}^T)^{-1}), \quad (9)$$

and by Equation (6) the predictive distribution is given by

$$p(s|\mathbf{t}) = \mathcal{N}(s|\beta\boldsymbol{\phi}(\mathbf{z})^T(\alpha\mathbf{I} + \beta\boldsymbol{\Phi}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}\mathbf{t}, \beta^{-1} + \boldsymbol{\phi}(\mathbf{z})^T(\alpha\mathbf{I} + \beta\boldsymbol{\Phi}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\phi}(\mathbf{z})). \quad (10)$$

In a word, Gaussian distributions map probability theory to linear algebra [5].

5. Gaussian Processes for Regression

Gaussian processes are generalized Gaussian distributions when the dimension goes to infinity [4,5]. After choosing an appropriate kernel function $k(\cdot, \cdot)$, the model of Gaussian

processes for regression is given by

$$p\left(\begin{bmatrix} s \\ \mathbf{t} \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} s \\ \mathbf{t} \end{bmatrix} \middle| \begin{bmatrix} 0 \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} u + \beta^{-1} & \mathbf{k}^T \\ \mathbf{k} & \mathbf{V} + \beta^{-1}\mathbf{I} \end{bmatrix}\right) \quad (11)$$

where $u = k(\mathbf{z}, \mathbf{z})$, $\mathbf{k} = [k(\mathbf{x}_1, \mathbf{z}) \dots k(\mathbf{x}_m, \mathbf{z})]^T$, $[\mathbf{V}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. From Equation (4) it follows that the predictive distribution is

$$p(s|\mathbf{t}) = \mathcal{N}(s|\mathbf{k}^T(\mathbf{V} + \beta^{-1}\mathbf{I})^{-1}\mathbf{t}, u + \beta^{-1} - \mathbf{k}^T(\mathbf{V} + \beta^{-1}\mathbf{I})^{-1}\mathbf{k}) \quad (12)$$

Bayesian linear regression can be seen a special case of Gaussian process regression since when the kernel function is taken to be the inner product of feature vectors Gaussian process regression is reduced to Bayesian linear regression.

6. Approximate Inference

Bayesian learning involves intractable integration, so approximate inference is often employed to arrive at the final result. Both Laplace approximation and Gaussian expectation propagation use the Gaussian distribution as the building block to approximate another distribution.

6.1. Laplace Approximation

Suppose we want to use a Gaussian density $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to approximate another density $p(\mathbf{x})$. Let $\mathbf{x}_0$ be one maximal point of $p(\mathbf{x})$. It should be emphasized that for a given density $p(\mathbf{x})$, there are different Laplace approximations associated with different maximal points of $p(\mathbf{x})$. Let the second order Taylor expansion of $\ln p(\mathbf{x})$ around $\mathbf{x}_0$ be

$$\ln p(\mathbf{x}) \approx \ln p(\mathbf{x}_0) - \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{A}(\mathbf{x} - \mathbf{x}_0) \quad (13)$$

where $\mathbf{A} = -\nabla^2 \ln p(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0}$ is the Hessian matrix of $-\ln p(\mathbf{x})$ evaluated at $\mathbf{x}_0$. The Laplace approximation is given by $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{x}_0, \mathbf{A}^{-1})$. It can be seen that $p(\mathbf{x})$ need not be normalized properly, that is $p(\mathbf{x})$ can be known up to a scaling constant, as is often the case in many practical applications of Bayesian machine learning. Equation (13) can be rewritten as

$$p(\mathbf{x}) \approx p(\mathbf{x}_0) e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T \mathbf{A}(\mathbf{x}-\mathbf{x}_0)} \quad (14)$$

so as a byproduct we have

$$\int p(\mathbf{x}) d\mathbf{x} \approx p(\mathbf{x}_0) \frac{(2\pi)^{n/2}}{|\mathbf{A}|^{1/2}} \begin{cases} \Lambda_j = \Lambda^{new} - \Lambda_{-j} \\ \Lambda_j \boldsymbol{\mu}_j = \Lambda^{new} \boldsymbol{\mu}^{new} - \Lambda_{-j} \boldsymbol{\mu}_{-j} \\ s_j = Z_j \frac{|\Lambda^{new}|^{1/2}}{|\Lambda_{-j}|^{1/2}} \exp \left(\frac{1}{2} \boldsymbol{\mu}^{newT} \Lambda^{new} \boldsymbol{\mu}^{new} - \frac{1}{2} \boldsymbol{\mu}_j^T \Lambda_j \boldsymbol{\mu}_j - \frac{1}{2} \boldsymbol{\mu}_{-j}^T \Lambda_{-j} \boldsymbol{\mu}_{-j} \right) \end{cases} \quad (15)$$

7. Conclusion

Properties of Gaussian distributions are reviewed and typical applications of Gaussian distributions in machine learning are demonstrated, including Bayes' theorem for classification, Bayesian linear regression, Gaussian process regression, Laplace approximation, and Gaussian expectation propagation. In all these cases, closed-form analytical formulas are obtained, owing to the algebraic properties of the Gaussian distribution. We hope that after becoming familiar with Gaussian distributions, students will be well prepared to study more complex models in machine learning.

Acknowledgments

The author thanks his colleges and students for their kindness and support.

This approximation can be useful in Bayesian model selection.

6.2. Gaussian Expectation Propagation

The Laplace approximation focuses on local behaviors near maximal points, while expectation propagation plays more emphasis on global properties [6]. Suppose the unnormalised probability density function we want to approximate has a multiplicative form $p(\mathbf{x}) = \prod_{i=1}^m f_i(\mathbf{x})$. In Gaussian expectation propagation we use $q(\mathbf{x}) = \prod_{i=1}^m \tilde{f}_i(\mathbf{x})$ to approximate $p(\mathbf{x})$, where $\tilde{f}_i(x) = s_i \exp \left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \Lambda_i(\mathbf{x} - \boldsymbol{\mu}_i) \right)$. The remaining task is to find $\{s_i, \boldsymbol{\mu}_i, \Lambda_i\}_{i=1}^m$ such that $q(\mathbf{x})$ is close to $p(\mathbf{x})$.

Gaussian expectation propagation computes $\{s_i, \boldsymbol{\mu}_i, \Lambda_i\}_{i=1}^m$ in an alternative way. Specifically, each factor $\tilde{f}_j$ is updated with other factors in $q(\mathbf{x})$ fixed. When updating $\tilde{f}_j$, $\tilde{f}_i$, where $i \neq j$ is fixed temporarily, so we need to find $s_j, \boldsymbol{\mu}_j, \Lambda_j$ such that $\tilde{f}_j(\mathbf{x}) \prod_{i \neq j} \tilde{f}_i(\mathbf{x})$ is close to $f_j(\mathbf{x}) \prod_{i \neq j} \tilde{f}_i(\mathbf{x})$. Since $q(\mathbf{x}) \propto \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \Lambda^{-1})$, where $\Lambda = \sum_{i=1}^m \Lambda_i$, $\Lambda \boldsymbol{\mu} = \sum_{i=1}^m \Lambda_i \boldsymbol{\mu}_i$, $\prod_{i \neq j} \tilde{f}_i(\mathbf{x}) \propto q_{-j}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{-j}, \Lambda_{-j}^{-1})$, where $\Lambda_{-j} = \Lambda - \Lambda_j$, $\Lambda_{-j} \boldsymbol{\mu}_{-j} = \Lambda \boldsymbol{\mu} - \Lambda_j \boldsymbol{\mu}_j$. If the generalized KL divergence is used as the disparity between $\tilde{f}_j(\mathbf{x}) q_{-j}(\mathbf{x})$ and $f_j(\mathbf{x}) q_{-j}(\mathbf{x})$, then they must share the same zeroth, first, and second order moments to minimize the disparity. Let $\int f_j(\mathbf{x}) q_{-j}(\mathbf{x}) d\mathbf{x} = Z_j$, $\boldsymbol{\mu}^{new} = \mathbb{E}[\mathbf{x}]$, $\Lambda^{new} = (\mathbb{V}[\mathbf{x}])^{-1}$, where $\mathbb{E}$ and $\mathbb{V}$ are operators returning expectation vector and covariance matrix respectively and the associated expectation is with respect to the distribution $f_j(\mathbf{x}) q_{-j}(\mathbf{x}) / Z_j$, then the update formulas for $s_j, \boldsymbol{\mu}_j, \Lambda_j$ are

References

- [1] Hui-Tzu Chang, Chia-Yu Lin (2024). Applying competition-based learning to stimulate students' practical and competitive AI ability in a machine learning curriculum. IEEE Transactions on Education, p.256-265.
- [2] Christopher M. Bishop and Hugh Bishop (2023). Deep learning: foundations and concepts. Springer.
- [3] Christopher M. Bishop (2006). Pattern recognition and machine learning. Springer.
- [4] Carl Edward Rasmussen and Christopher K. I. Williams (2006). Gaussian processes for machine learning. MIT Press.
- [5] Philipp Hennig, Michael A Osborne, and Hans P Kersting (2022). Probabilistic Numerics. Cambridge University Press.
- [6] Kevin Patrick Murphy (2023). Probabilistic machine learning: advanced topics. MIT Press.