Lecture 2: Bayesian models

Reading: Sections 2.4, 8.3

GU4241/GR5241 Statistical Machine Learning

Linxi Liu

January 19, 2017

Annoucements

Office Hours

Monday: Haolei Weng, 3:00 – 4:00pm
Tuesday: Shuaiwen Wang, 7:30 – 8:30pm
Wednesday: Shuaiwen Wang, 7:30 – 8:30pm
Thursday: Linxi Liu, 7:30 – 9:30pm
Friday: Haolei Weng, 3:00 – 4:00pm

The location is at the lounge of 10th floor, SSW.

Supervised vs. unsupervised learning

we have response in surp

make prediction;

make reference: understand the structure of the data

$$Y = f(X) + \underbrace{\varepsilon}_{\text{Random error}}$$

Motivations:

- **Prediction:** Useful when the input variable is readily available, but the output variable is not.

  Example: Predict stock prices next week using data from last month.

Supervised vs. unsupervised learning

$$Y = f(X) + \underbrace{\varepsilon}_{\text{Random error}}$$

Motivations:

- **Prediction:** Useful when the input variable is readily available, but the output variable is not.
- **Inference:** A model for f can help us understand the structure of the data — which variables influence the output, and which don't? What is the relationship between each variable and the output, e.g. linear, non-linear?

  Example: What is the influence of genetic variations on the incidence of heart disease.

Prediction error

Training data: $(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)$
Predicted function: $\hat{f}$.

Our goal in supervised learning is to minimize the expected prediction error. Under squared-error loss, this is the *Mean Squared Error*:

$$MSE(\hat{f}) = E(y_0 - \hat{f}(x_0))^2.$$

Unfortunately, this quantity cannot be computed, because we don't know the joint distribution of (X, Y).

minimize the expected prediction error: recoginize as mes

training mse is not good estimation

The bias variance decomposition

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and $\hat{f}$ be estimated from n training samples $(x_1, y_1) \ldots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs $(y_1, \ldots, y_n)$. Then, the Mean Squared Error at x_0 can be decomposed:

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon_0).$$

in order to get better mse: train the model first based on the training model;
after that obtain estimate of the model;
then we get the observed test data set;
calculate the loss function based on the seperate data set
then the eorror is the best

The bias variance decomposition

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and $\hat{f}$ be estimated from n training samples $(x_1, y_1) \ldots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs $(y_1, \ldots, y_n)$. Then, the Mean Squared Error at x_0 can be decomposed:

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon_0).$$

Irreducible error

The bias variance decomposition

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and $\hat{f}$ be estimated from n training samples $(x_1, y_1) \ldots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs $(y_1, \ldots, y_n)$. Then, the Mean Squared Error at x_0 can be decomposed:

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon_0).$$

The variance of the estimate of Y: $E[\hat{f}(x_0) - E(\hat{f}(x_0))]^2$

This measures how much the estimate of $\hat{f}$ at x_0 changes when we sample new training data.

Text

The bias variance decomposition

decomopostion is for the classification

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and $\hat{f}$ be estimated from n training samples $(x_1, y_1) \ldots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs $(y_1, \ldots, y_n)$. Then, the Mean Squared Error at x_0 can be decomposed:

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon_0).$$

The squared bias of the estimate of Y: $[E(\hat{f}(x_0)) - f(x_0)]^2$

This measures the deviation of the average prediction $\hat{f}(x_0)$ from the truth $f(x_0)$.

Implications of bias variance decomposition

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon).$$

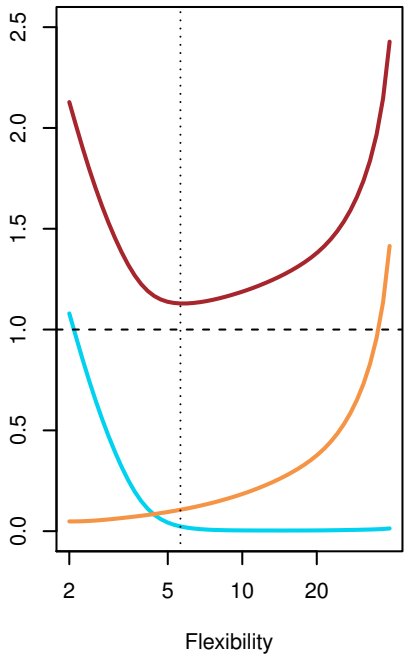
if the model becomes richer, more choice, more flexibility and thus high variance

sometime less variance but high bias. This is a trade off

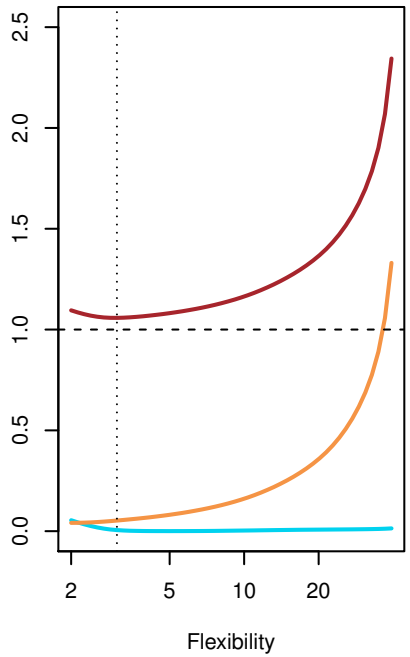
- The MSE is always positive.
- Each element on the right hand side is always positive.
- Therefore, typically when we decrease the bias beyond some point, we increase the variance, and vice-versa.

$$\text{More flexibility} \iff \text{Higher variance} \iff \text{Lower bias.}$$

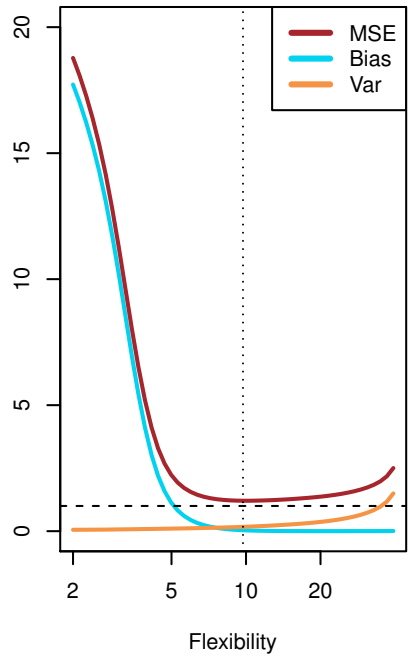
Squiggly f, high noise



Linear f, high noise



Squiggly f, low noise



ISL Figure 2.12

Parametric Models

Models

such that normal dist

A **model** $\mathcal{P}$ is a set of probability distributions. We index each distribution by a parameter value $\theta \in \mathcal{T}$; we can then write the model as

$$\mathcal{P} = \{P_\theta | \theta \in \mathcal{T}\} .$$

The set $\mathcal{T}$ is called the **parameter space** of the model.

Parametric model

model can be characterised as specific case, such as gaussian model

The model is called **parametric** if the number of parameters (i.e. the dimension of the vector θ) is (1) finite and (2) independent of the number of data points. Intuitively, the complexity of a parametric model does not increase with sample size.

nonp-parametric model: we can not use finite number to calculate the model

Density representation

For parametric models, we can assume that $\mathcal{T} \subset \mathbb{R}^d$ for some fixed dimension d. We usually represent each P_θ be a density function $p(x|\theta)$.

Maximum Likelihood Estimation

choose mlse estimator.

Setting

- Given: Data $x_1, \ldots, x_n$, parametric model $\mathcal{P} = \{p(x|\theta) \,|\, \theta \in \mathcal{T}\}$.
- Objective: Find the distribution in $\mathcal{P}$ which best explains the data. That means we have to choose a "best" parameter value $\hat{\theta}$.

Maximum Likelihood approach

Maximum Likelihood assumes that the data is best explained by the distribution in $\mathcal{P}$ under which it has the highest probability (or highest density value).

Hence, the **maximum likelihood estimator** is defined as

$$\hat{\theta}_{ML} := \arg\max_{\theta \in \mathcal{T}} p(x_1, \ldots, x_n|\theta)$$

the parameter which maximizes the joint density of the data.

Analytic Maximum Likelihood

The i.i.d. assumption

The standard assumption of ML methods is that the data is **independent and identically distributed (i.i.d.)**, that is, generated by independently sampling repeatedly from the same distrubtion P.

If the density of P is $p(x|\theta)$, that means the joint density decomposes as

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i|\theta)$$

Maximum Likelihood equation

The analytic criterion for a maximum likelihood estimator (under the i.i.d. assumption) is:

$$\nabla_\theta \Big(\prod_{i=1}^{n} p(x_i|\theta) \Big) = 0$$

We use the "logarithm trick" to avoid a huge product rule computation.

Logarithm Trick

Recall: Logarithms turn products into sums

$$\log\Big(\prod_i f_i\Big) = \sum_i \log(f_i)$$

Logarithms and maxima

The logarithm is monotonically increasing on $\mathbb{R}_+$.

Consequence: Application of $\log$ does not change the *location* of a maximum or minimum:

$$\max_y \log(g(y)) \neq \max_y g(y) \qquad \text{The } \textit{value} \text{ changes.}$$

$$\arg\max_y \log(g(y)) = \arg\max_y g(y) \qquad \text{The } \textit{location} \text{ does not change.}$$

Analytic MLE

Likelihood and logarithm trick

$$\begin{aligned}\hat{\theta}_{ML} &= \arg\max_{\theta} \prod_{i=1}^{n} p(x_i|\theta) &&= \arg\max_{\theta} \log\Big(\prod_{i=1}^{n} p(x_i|\theta)\Big) \\ &&&= \arg\max_{\theta} \sum_{i=1}^{n} \log p(x_i|\theta)\end{aligned}$$

Analytic maximality criterion

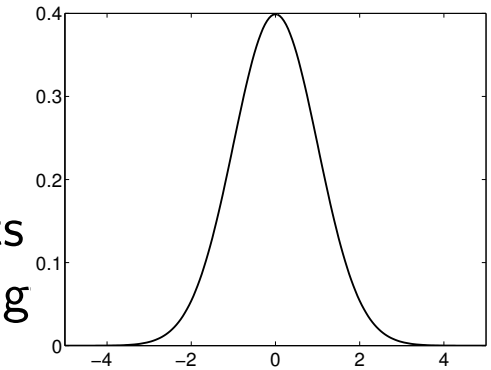
$$0 = \sum_{i=1}^{n} \nabla_\theta \log p(x_i|\theta) = \sum_{i=1}^{n} \frac{\nabla_\theta p(x_i|\theta)}{p(x_i|\theta)}$$

Whether or not we can solve this analytically depends on the choice of the model!

Example: Gaussian Mean MLE

Gaussian density in one dimension

$$g(x;\mu,\sigma) := \frac{1}{\sqrt{2\pi}\sigma}\exp\Big(-\frac{(x-\mu)^2}{2\sigma^2}\Big)$$

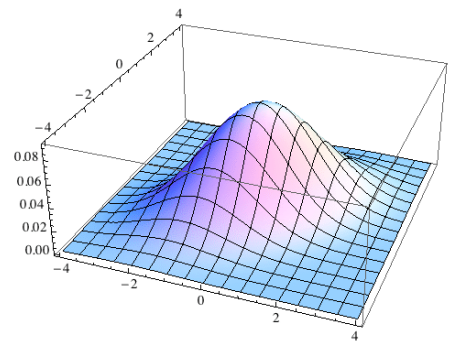


- The quotient $\frac{x-\mu}{\sigma}$ measures deviation of x from its expected value in units of σ (i.e. σ defines the leng scale)

Gaussian density in d dimensions

The quadratric function

$$-\frac{(x-\mu)^2}{2\sigma^2} \quad = \quad -\frac{1}{2}(x-\mu)(\sigma^2)^{-1}(x-\mu)$$



is replaced by a quadratic form:

$$g(\mathbf{x};\boldsymbol{\mu},\Sigma) := \frac{1}{\sqrt{(2\pi)^d\det(\Sigma)}}\exp\Big(-\frac{1}{2}\left\langle(\mathbf{x}-\boldsymbol{\mu}),\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\rangle\Big)$$

Example: Gaussian Mean MLE

Model: Multivariate Gaussians

The model $\mathcal{P}$ is the set of all Gaussian densities on $\mathbb{R}^d$ with *fixed* covariance matrix Σ,

$$\mathcal{P} = \{g(\,.\,|\mu, \Sigma) \,|\, \mu \in \mathbb{R}^d\}\ ,$$

where g is the Gaussian density function. The parameter space is $\mathcal{T} = \mathbb{R}^d$.

MLE equation

We have to solve the maximum equation

$$\sum_{i=1}^{n} \nabla_\mu \log g(x_i|\mu, \Sigma) = 0$$

it is unbiased sample covariance, but mle is not unbiased estimator

for μ.

Example: Gaussian Mean MLE

$$
\begin{aligned}
0 &= \sum_{i=1}^{n} \nabla_\mu \log \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\Big(-\frac{1}{2}\left\langle (x_i-\mu), \Sigma^{-1}(x_i-\mu)\right\rangle\Big) \\
&= \sum_{i=1}^{n} \nabla_\mu \Big(\log\Big(\frac{1}{\sqrt{(2\pi)^d |\Sigma|}}\Big) + \log\Big(\exp\Big(-\frac{1}{2}\left\langle (x_i-\mu), \Sigma^{-1}(x_i-\mu)\right\rangle\Big)\Big) \\
&= \sum_{i=1}^{n} \nabla_\mu \Big(-\frac{1}{2}\left\langle (x_i-\mu), \Sigma^{-1}(x_i-\mu)\right\rangle\Big) = -\sum_{i=1}^{n} \Sigma^{-1}(x_i-\mu)
\end{aligned}
$$

Multiplication by $(-\Sigma)$ gives

$$
0 = \sum_{i=1}^{n} (x_i - \mu) \qquad \Rightarrow \qquad \mu = \frac{1}{n}\sum_{i=1}^{n} x_i
$$

Conclusion

The maximum likelihood estimator of the Gaussian expectation parameter for fixed covariance is

$$
\hat{\mu}_{\mathsf{ML}} := \frac{1}{n}\sum_{i=1}^{n} x_i
$$

Example: Gaussian with Unknown Covariance

Model: Multivariate Gaussians

The model $\mathcal{P}$ is now

$$\mathcal{P} = \{g(\,.\,|\mu, \Sigma) \,|\, \mu \in \mathbb{R}^d, \Sigma \in \Delta_d\}\ ,$$

where Δ_d is the set of positive definite $d \times d$-matrices. The parameter space is $\mathcal{T} = \mathbb{R}^d \times \Delta_d$.

ML approach

Since we have just seen that the ML estimator of μ does not depend on Σ, we can compute $\hat{\mu}_{\mathsf{ML}}$ first. We then estimate Σ using the criterion

$$\sum_{i=1}^{n} \nabla_\Sigma \log g(x_i|\hat{\mu}_{\mathsf{ML}}, \Sigma) = 0$$

Solution

The ML estimator of Σ is

$$\hat{\Sigma}_{\mathsf{ML}} := \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu}_{\mathsf{ML}})(x_i - \hat{\mu}_{\mathsf{ML}})^t\ .$$

Bayesian models

true parameter contains uncertainty, should be characterized as random variable

The defining assumption of **Bayesian statistics** is that the distribution P_θ which models the data is a random quantity and itself has a distribution Q. The generative model for data $X_1, X_2, \ldots$ is

$$\begin{aligned} P_\theta &\sim Q \\ X_1, X_2, \ldots &\overset{i.i.d.}{\sim} P_\theta \end{aligned}$$

The rational behind the approach is:

- In any statistical approach (Bayesian or frequentist), the distribution P_θ is unknown.
- Bayesian statistics argues that any form of uncertainty should be expressed by probability distributions.
- We can think of the randomness in Q as a model of the statistician's lack of knowledge regarding P_θ.

Prior and posterior

Q stands for the priori

The distribution Q of P_θ is called the **a priori distribution** (or the **prior** for short). We use q to denote its density if it exists.

Our objective is to determine the conditional probability of P given observed data

$$\Pr(\theta|x_1, \ldots, x_n).$$

be able to calculate this one, next slide

The distribution is called the **a posteriori distribution** or **posterior**.

Bayes' Theorem

Given data $X_1, \ldots, X_n$, we can compute the posterior by

$$\Pr(\theta|x_1, \ldots, x_n) = \frac{(\prod_{i=1}^n p(x_i|\theta))q(\theta)}{p(x_1, \ldots, x_n)} = \frac{(\prod_{i=1}^n p(x_i|\theta))q(\theta)}{\int (\prod_{i=1}^n p(x_i|\theta))\, q(\theta)}.$$

The individual terms have names:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

the marginal probability is the evidence

Example: unknown Gaussian mean

Model

this is a linkelihood model: g(xlu,o^2)

We assume that the data is generated from a Gaussian with fixed variance σ^2. The mean μ is unknown. The model likelihood is $p(x|\mu,\sigma) = g(x|\mu,\sigma)$ (where g is the Gaussian density on the line).

Bayesian model

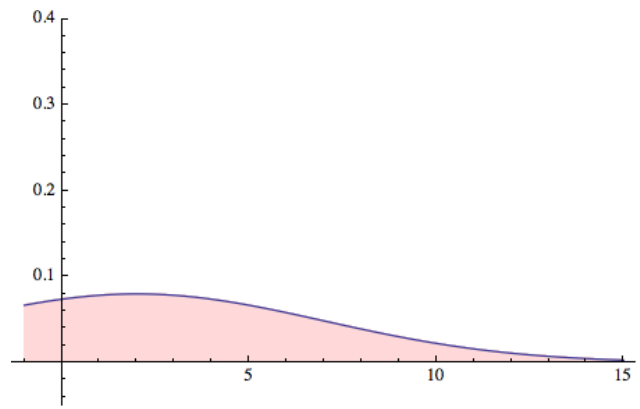
the pirior is also the gaussian distribution: q(xlu0,O0^2)

same distribution, different form

We choose a Gaussian prior on μ,

$$q(\mu) := g(\mu|\mu_0,\sigma_0) \ .$$

In the figure, $\mu_0 = 2$ and $\sigma_0 = 5$. Hence, we assume that $\mu_0 = 2$ is the most probable value of μ, and that $\mu \in [-3,7]$ with a probability ~ 0.68.



Example: unknown Gaussian mean

Application of Bayes' formula to the Gaussian-Gaussian model shows the posterior distribution is

is still gaussian, with difference mean and variance

$$\Pr(\mu|x_{1:n}) = g(\mu|\mu_n, \sigma_n),$$

where $\mu_n := \frac{\sigma^2\mu_0 + \sigma_0^2 \sum_{i=1}^n x_i}{\sigma^2 + n\sigma_0^2}$ and $\sigma_n^2 := \frac{\sigma^2\sigma_0^2}{\sigma^2 + n\sigma_0^2}$.

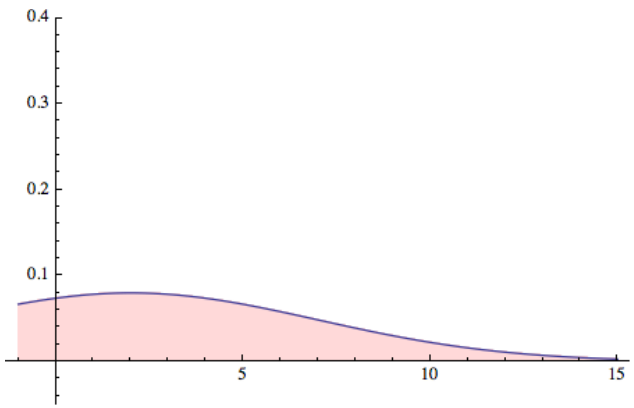
he posterior mean is the compramise between the sample average and prior.
As n increase, the posterior mean will close to the sample average,
and the distribution would be more consentrated around the sample mean

Example: unknown Gaussian mean

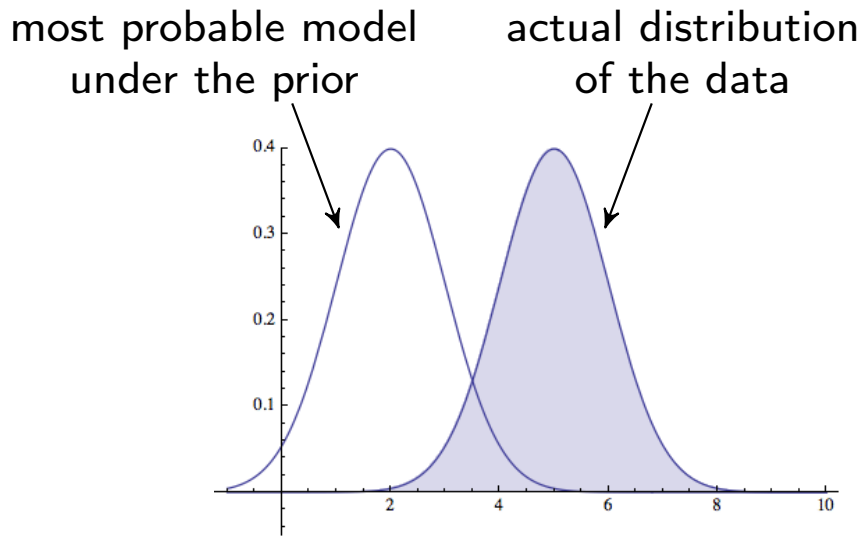
Application of Bayes' formula to the Gaussian-Gaussian model shows the posterior distribution is

$$\Pr(\mu|x_{1:n}) = g(\mu|\mu_n, \sigma_n),$$

where $\mu_n := \frac{\sigma^2\mu_0 + \sigma_0^2 \sum_{i=1}^n x_i}{\sigma^2 + n\sigma_0^2}$ and $\sigma_n^2 := \frac{\sigma^2\sigma_0^2}{\sigma^2 + n\sigma_0^2}$.



Prior



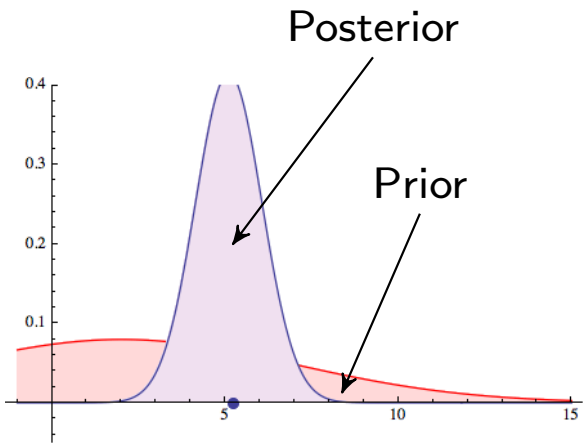
Sampling distribution

Example: unknown Gaussian mean

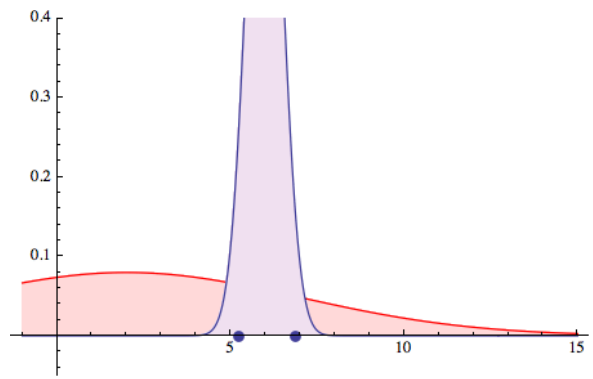
Application of Bayes' formula to the Gaussian-Gaussian model shows the posterior distribution is

$$\Pr(\mu|x_{1:n}) = g(\mu|\mu_n, \sigma_n),$$

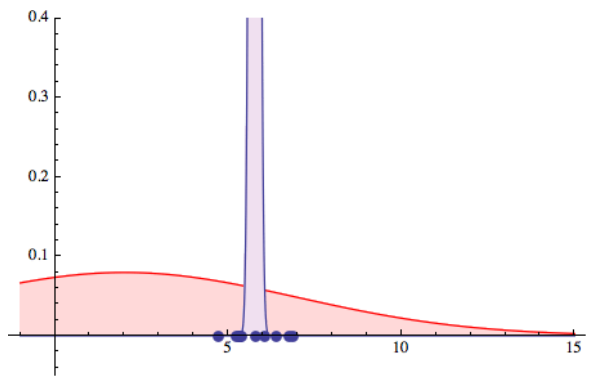
where $\mu_n := \frac{\sigma^2\mu_0 + \sigma_0^2 \sum_{i=1}^n x_i}{\sigma^2 + n\sigma_0^2}$ and $\sigma_n^2 := \frac{\sigma^2\sigma_0^2}{\sigma^2 + n\sigma_0^2}$.



$n = 1$



$n = 2$



$n = 10$

MAP estimation

Suppse $\Pi(\theta|x_{1:n})$ is the posterior of a Bayesian model. The estimator

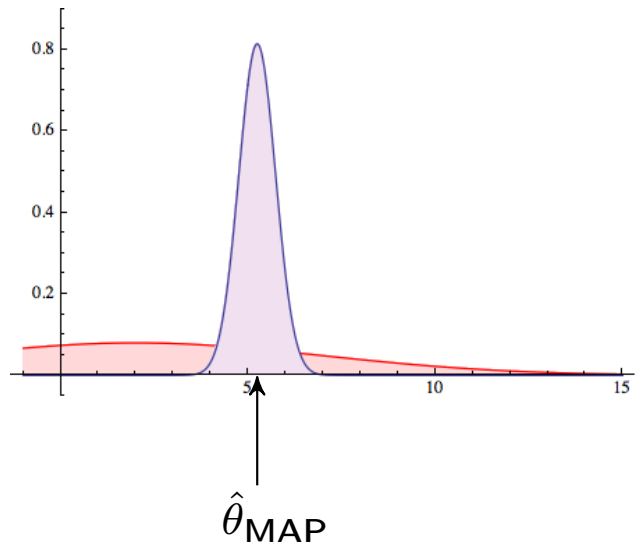
$$\hat{\theta}_{\mathsf{MAP}} = \arg\max_{\theta} \Pi(\theta|x_{1:n})$$

is called the **maximum a posteriori** (or **MAP**) estimator for θ.

Point estimates

The goal of Bayesian inference is to compute the posterior distribution. Contrast this to classical statistics (e.g. maximum likelihood), where we typically estimate a single value for θ (a so-called **point estimate**).

MAP estimation combines aspects of Bayesian methodology (use of a prior) with aspects of classical methodology (since $\hat{\theta}_{\mathsf{MAP}}$ is a point estimate).



MAP and regularization

Logarithmic view

Since the logarithm leaves the maximum invariant,

$$\hat{\theta}_{\mathsf{MAP}} = \arg\max_{\theta} \Pi(\theta|x_{1:n}) = \arg\max_{\theta} \log \Pi(\theta|x_{1:n})$$

Substituting in the Bayes equation gives

$$\log \Pi(\theta|x_{1:n}) = \sum_{i=1}^{n} \log p(x_i|\theta) + \log q(\theta) - \log p(x_1, \ldots, x_n) \ .$$

MAP as regularized ML

Since log-evidence does not depend on θ,

$$\hat{\theta}_{\mathsf{MAP}} = \arg\max_{\theta} \Big\{ \sum_{i=1}^{n} \log p(x_i|\theta) + \log q(\theta) \Big\}$$

Thus, the MAP estimate can be regarded as a regularized version of a maximum likelihood estimator. The regularization term $\log q(\theta)$ favors values where q (and hence $\log q$) is large. MAP is regularized mle

Classification problems

In a classification setting, the output takes values in a discrete set.

For example, if we are predicting the brand of a car based on a number of variables, the function f takes values in the set
$\{$Ford, Toyota, Mercedes-Benz, $\dots\}$.

We will use slightly different notation:

$$P(X, Y) : \text{joint distribution of } (X, Y),$$
$$P(Y \mid X) : \text{conditional distribution of } X \text{ given } Y,$$
$$\hat{y}_i : \text{prediction for } x_i.$$

Loss function for classification

There are many ways to measure the error of a classification prediction. One of the most common is the 0-1 loss:

$$E(\mathbf{1}(y_0 \neq \hat{y}_0))$$

Like the MSE, this quantity can be estimated from training and test data by taking a sample average:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}(y_i \neq \hat{y}_i)$$

Bayes classifier

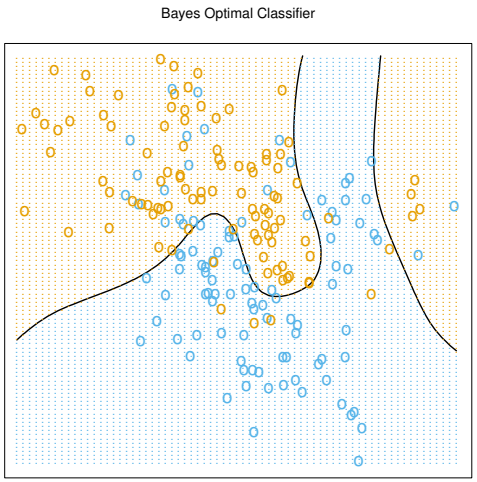


FIGURE 2.5. *The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).*

In practice, we never know the joint probability P. However, we can assume that it exists.

The Bayes classifier assigns:

$$\hat{y}_i = \text{argmax}_k \;\; P(Y = k \mid X = x_i)$$

It can be shown that this is the best classifier under the 0-1 loss.

Def: f(ax+by) = af(x) +bf(y)

X m*n: f(z) = Xz

Images of Linear Mappings (1)

Linear mapping

A matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ defines a linear mapping $f_{\mathbf{x}} : \mathbb{R}^m \to \mathbb{R}^n$.

Image

Recall: The **image** of a mapping f is the set of all possible function values, here

$$\text{image}(f_{\mathbf{x}}) := \{\mathbf{y} \in \mathbb{R}^n \,|\, \mathbf{Xz} = \mathbf{y} \text{ for some } \mathbf{z} \in \mathbb{R}^m\}$$

Image of a linear mapping

- The image of a linear mapping $\mathbb{R}^m \to \mathbb{R}^n$ is a linear subspace of $\mathbb{R}^n$.
- The columns of $\mathbf{X}$ form a basis of the image space:

$$\text{image}(\tilde{\mathbf{X}}) = \text{span}\{\mathbf{X}_1^{\text{col}}, \ldots, \mathbf{X}_m^{\text{col}}\}$$

- This is one of most useful things to remember about matrices, so, again:

The columns span the image.

Images of Linear Mappings (2)

Dimension of the image space

Clearly: The number of linearly independent column vectors. This number is called the **column rank** of $\mathbf{X}$.

Invertible mappings

Recall: A mapping f is invertible if it is one-to-one, i.e. for each function value $\tilde{\mathbf{y}}$ there is exactly one input value with $f(\mathbf{z}) = \tilde{\mathbf{y}}$.

Invertible matrices

The matrix $\tilde{\mathbf{X}}$ is called invertible if $f_{\mathbf{x}}$ is invertible.

- Only square matrices can be invertible.
- For a *linear* mapping: If $\tilde{\mathbf{X}}$ is a square matrix $f_{\mathbf{x}}$ is invertible iff the image has the same dimension as the input space.
- Even if $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times m}$, the matrix $\tilde{\mathbf{X}}^t \tilde{\mathbf{X}}$ is in $\mathbb{R}^{m \times m}$ (a square matrix).
- So: $\tilde{\mathbf{X}}^t \tilde{\mathbf{X}}$ is invertible if $\tilde{\mathbf{X}}$ has full column rank.

Symmetric and Orthogonal Matrices

Recall: Transpose

The transpose A^T of a matrix $A \in \mathbb{R}^m$ is the matrix with entries

$$(A^\mathsf{T})_{ij} := A_{ji}$$

Orthogonal matrices

A matrix $O \in \mathbb{R}^{m \times m}$ is called **orthogonal**

$$O^{-1} = O^T$$

Orthogonal matrices describe two types of operations:

1. Rotations of the coordinate system.
2. Permutations of the coordinate axes.

Symmetric matrices

A matrix $A \in \mathbb{R}^{m \times m}$ is called **symmetric**

$$A = A^\mathsf{T}$$

Note: Symmetric and orthogonal matrices are very different objects. Only the identity is both.

Orthonormal Bases

Recall: ONB

A basis $\{v_1, \ldots, v_m\}$ of $\mathbb{R}^m$ is called an **orthonormal basis** if

$$\langle v_i, v_j \rangle = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

In other words, the v_i are pairwise orthogonal and each of length 1.

Orthogonal matrices

A matrix is orthogonal precisely if its rows form an ONB. Any two ONBs can be transformed into each other by an orthogonal matrix.

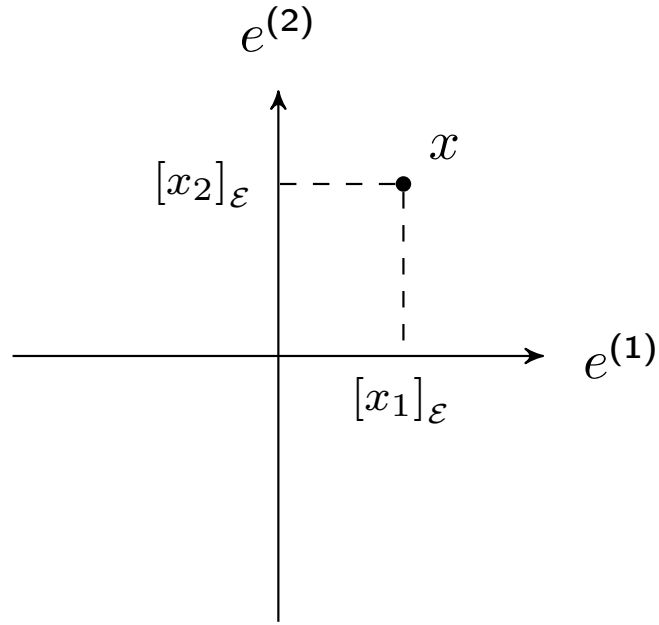
Basis representation

Representation of a vector

Suppose $\mathcal{E} = \{e_1, \ldots, e_d\}$ is a basis of a vector space. Then a vector x is represented as

$$x = \sum_{j=1}^{d} [x_j]_{\mathcal{E}} e^{(j)}$$

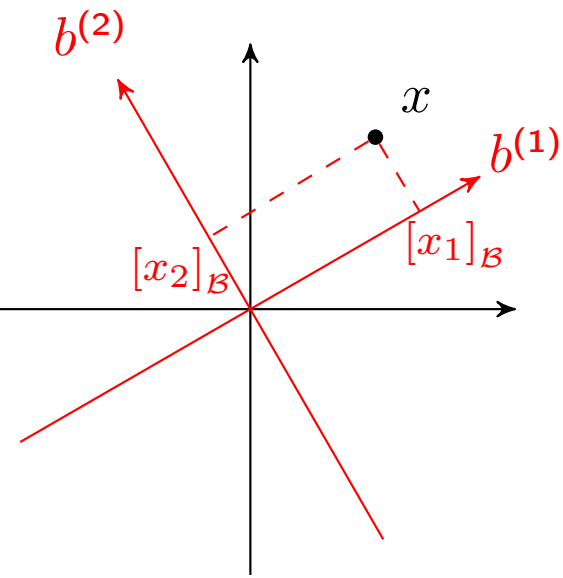
$[x_j]_{\mathcal{E}} \in \mathbb{R}$ are the coordinates of x w.r.t. $\mathcal{E}$.



Other bases

If $\mathcal{B} = \{b_1, \ldots, b_d\}$ is another basis, x can be represented alternatively as

$$x = \sum_{j=1}^{d} [x_j]_{\mathcal{B}} b^{(j)}$$



Changing bases

Change-of-basis matrix

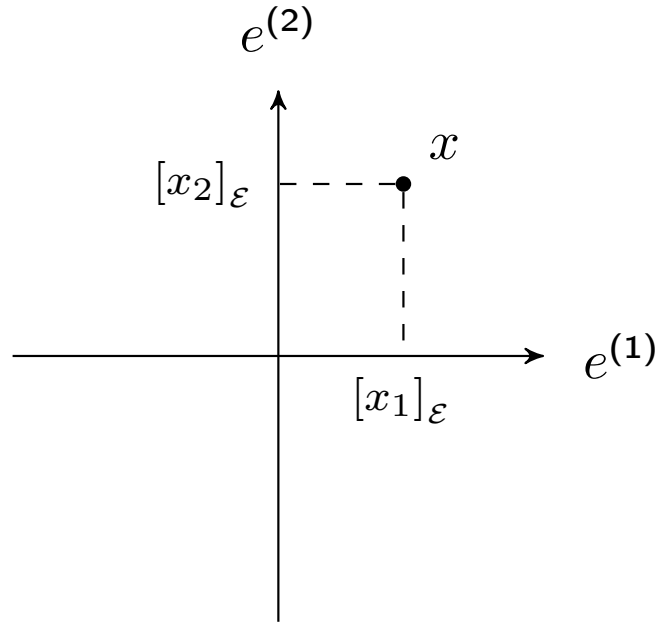
The matrix

$$M := \left([e^{(1)}]_{\mathcal{B}}, \ldots, [e^{(d)}]_{\mathcal{B}} \right)$$

transforms between the bases, i.e.

$$M[x]_{\mathcal{E}} = [x]_{\mathcal{B}} \ .$$

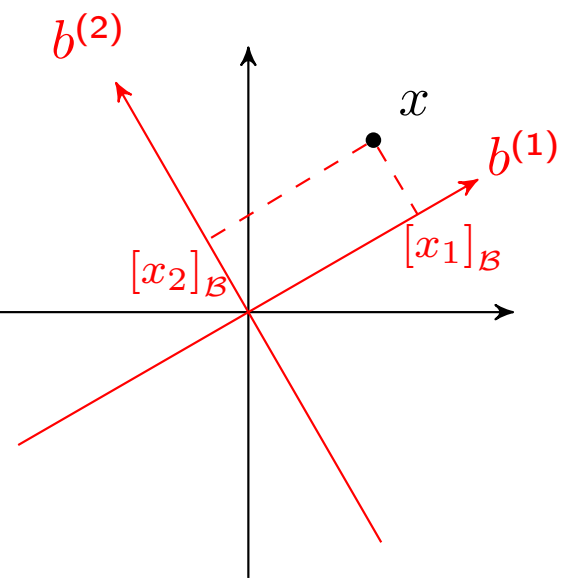
If both $\mathcal{E}$ and $\mathcal{B}$ are ONBs, M is orthogonal.



Representation of matrices

The matrix representing a linear mapping $A : \mathbb{R}^d \to \mathbb{R}^d$ in the basis $\mathcal{E}$ is computed as

$$[A]_{\mathcal{E}} := \left([A(e^{(1)})]_{\mathcal{E}}, \ldots, [A(e^{(d)})]_{\mathcal{E}} \right)$$



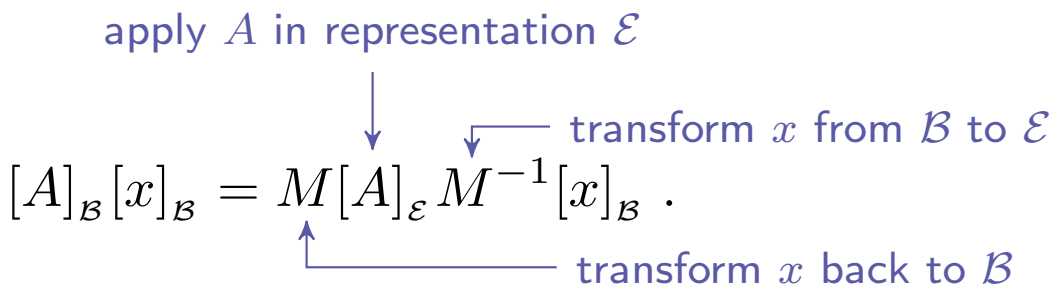
Basis Change for Linear Mappings

Transforming matrices

The matrix representing a linear mapping also changes when we change bases:

$$[A]_{\mathcal{B}} = M[A]_{\mathcal{E}} M^{-1} \ .$$

Applied to a vector x, this means:



Transforming between ONBs

If $\mathcal{V} = \{v_1, \ldots, v_m\}$ and $\mathcal{W} = \{w_1, \ldots, w_m\}$ are any two ONBs, there is an orthogonal matrix O such that

$$[A]_{\mathcal{V}} = O[A]_{\mathcal{W}} O^{-1}$$

for any linear mapping A.

Eigenvalues

We consider a square matrix $A \in \mathbb{R}^{m \times m}$.

Definition

A vector $\xi \in \mathbb{R}^m$ is called an **eigenvector** of A if the direction of ξ does not change under application of A. In other words, if there is a scalar λ such that

$$A\xi = \lambda\xi \ .$$

λ is called an **eigenvalue** of A for the eigenvector ξ.

Properties in general

- In general, eigenvalues are complex numbers $\lambda \in \mathbb{C}$.
- The class of matrices with the nicest eigen-structure are symmetric matrices, for which all eigenvectors are mutually orthogonal.

Eigenstructure of symmetric matrices

If a matrix is symmetric:

- There are rank(A) distinct eigendirections.
- The eigenvectors are pair-wise orthogonal.
- If rank$(A) = m$, there is an ONB of $\mathbb{R}^m$ consisting of eigenvectors of A.

  full rank of the eigen vector, all the eigen vectors form the dimension of the space

Definiteness

type	if ...
positive definite	all eigenvalues > 0
positive semi-definite	all eigenvalues ≥ 0
negative semi-definite	all eigenvalues ≤ 0
negative definite	all eigenvalues < 0
indefinite	none of the above

Orthonormal Bases

both orthogonal and normal

Recall: ONB

A basis $\{v_1, \ldots, v_m\}$ of $\mathbb{R}^m$ is called an **orthonormal basis** if

$$\langle v_i, v_j \rangle = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

In other words, the v_i are pairwise orthogonal and each of length 1.

Orthogonal matrices

A matrix is orthogonal precisely if its rows form an ONB. Any two ONBs can be transformed into each other by an orthogonal matrix.

Transforming between ONBs

If $\mathcal{V} = \{v_1, \ldots, v_m\}$ and $\mathcal{W} = \{w_1, \ldots, w_m\}$ are ONBs, there is an orthogonal matrix O such that

Text

$$A_{[\mathcal{V}]} = O A_{[\mathcal{W}]} O^{-1}$$

for any matrix A. By $A_{[\mathcal{V}]}$, we denote the representation of A in $\mathcal{V}$.

Eigenvector ONB

Setting

- Suppose A symmetric, $\xi_1, \ldots, \xi_m$ are eigenvectors and form an ONB.
- $\lambda_1, \ldots, \lambda_m$ are the corresponding eigenvalues.

How does A act on a vector $v \in \mathbb{R}^m$?

1. Represent v in basis $\xi_1, \ldots, \xi_m$:

$$v = \sum_{j=1}^{m} v_j^{\mathsf{A}} \xi_j \qquad \text{where } v_j^{\mathsf{A}} \in \mathbb{R}$$

2. Multiply by A: Eigenvector definition (recall: $A\xi_j = \lambda\xi_j$) yields

$$Av \quad = \quad A\Big(\sum_{j=1}^{m} v_j^{\mathsf{A}} \xi_j\Big) \quad = \quad \sum_{j=1}^{m} v_j^{\mathsf{A}} A\xi_j \quad = \quad \sum_{j=1}^{m} v_j^{\mathsf{A}} \lambda_j \xi_j$$

Conclusion

This is the image of the eighen vectors

A symmetric matrix acts by scaling the directions ξ_j.

Illustration

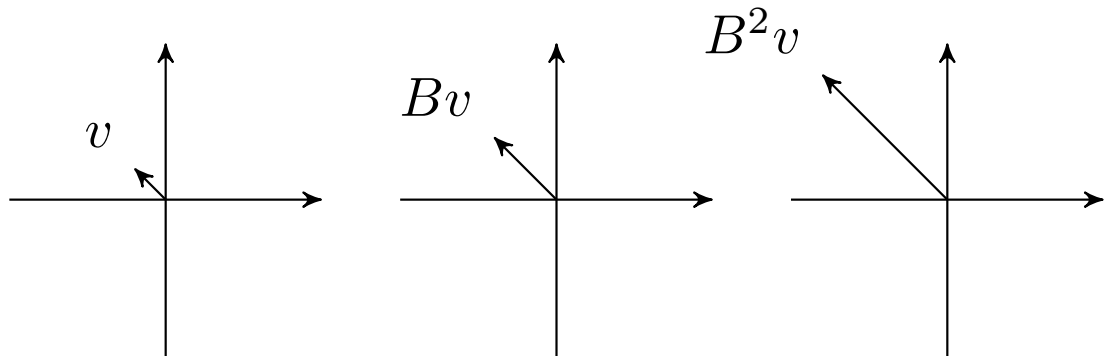
Setting

We *repeatedly* apply a symmetric matrix B to some vector $v \in \mathbb{R}^m$, i.e. we compute

$$Bv, \quad B(Bv) = B^2v, \quad B(B(Bv))) = B^3v, \quad \ldots$$

How does v change? it will converges to get the largest eigen vector

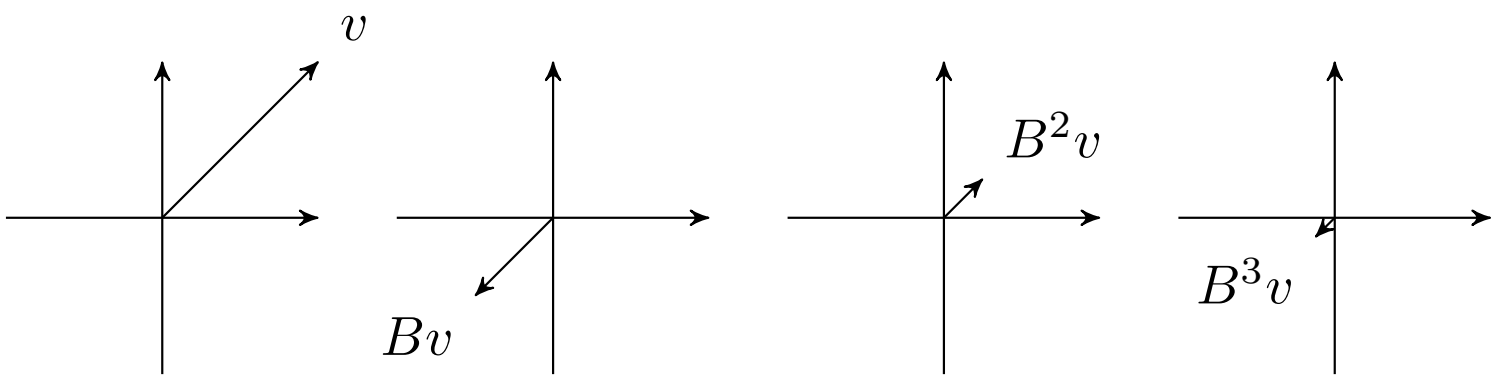
Example 1: v is an eigenvector with eigenvalue 2



The direction of v does not change, but its length doubles with each application of B.

Illustration

Example 2: v is an eigenvector with eigenvalue $-\frac{1}{2}$



For an arbitrary vector v

$$B^n v = \sum_{j=1}^{m} v_j^{\mathsf{B}} \lambda_j^n \xi_j$$

- The weight λ_j^n grows most rapidly for eigenvalue with largest absolute value.
- Consequence:
  The direction of $B^n v$ converges to the direction of the eigenvector with largest eigenvalue as n grows large.

Thanks to Sergio Bacallado and Peter Orbanz
for sharing the slides.