Statistical Machine Learning

Christian Walder

College of Engineering and Computer Science
The Australian National University

Canberra
Semester One, 2020.

(Many figures from C. M. Bishop, "Pattern Recognition and Machine Learning")

Outlines

Part VI

Three Models for Decision Problems

In increasing order of complexity

- Find a discriminant function $f(\mathbf{x})$ which maps each input directly onto a class label.
- Discriminative Models
  1.
  2.
- Generative Models
  1. Solve the inference problem of determining the class-conditional probabilities $p(\mathbf{x} \mid \mathcal{C}_k)$.
  2. *Also, infer the prior class probabilities*
  3. *Use Bayes theorem to find the posterior* $p(\mathcal{C}_k \mid \mathbf{x})$.
  4. Alternatively, model the joint distribution $p(\mathbf{x}, \mathcal{C}_k)$ directly.
  5. Use decision theory to assign each new $\mathbf{x}$ to one of the classes.

Data Generating Process

Given:

- class prior $p(\mathbf{t})$
- class-conditional $p(\mathbf{x} \mid \mathbf{t})$

to genera...

1. Sa...
2. Sa... distribution $p(\mathbf{x} \mid \mathbf{t})$.

(more about sampling later — this is called ...

Thinking about the data generating process is a usef... modelling step, especially when we have more prior knowledge.

Probabilistic Generative Models

- Generative approach: model class-conditional densities $p(\mathbf{x} \mid \mathcal{C}_k)$ and *class* priors (not parameter priors!) $p(\mathcal{C}_k)$ to calculate the posterior probability for class $\mathcal{C}_1$

$$p(\mathcal{C}_1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathcal{C}_1)p(\mathcal{C}_1)}{\quad}$$

where a and the logistic sigmoid function

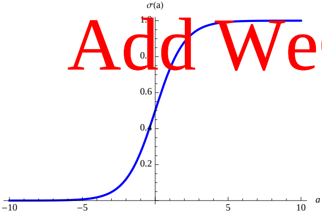
$$a(\mathbf{x}) = \ln \frac{p(\mathbf{x} \mid \mathcal{C}_1)\, p(\mathcal{C}_1)}{p(\mathbf{x} \mid \mathcal{C}_2)\, p(\mathcal{C}_2)} = \ln \frac{\quad}{\quad}$$

$$\sigma(a) = \frac{1}{1 + \exp(-a)}.$$

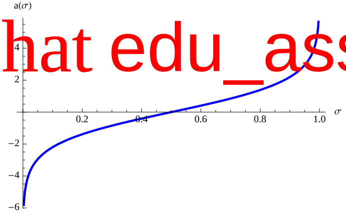
- One point of this re-writing: we may learn $a(\mathbf{x})$ directly as *e.g.* a deep neural network.

Logistic Sigmoid

- The logistic sigmoid function is called a “squashing function” because it squashes the real axis into a finite interval $(0, 1)$.
- Well known properties (derive them):
  - Symmetry: $\sigma(\quad a) = 1 \quad\; \sigma(a)$
  - $-$
- Inv



Sigmoid $\sigma(a) = \frac{1}{1+\exp(-a)}$



Logit $a(\sigma) = \ln\left(\frac{\sigma}{1-\sigma}\right)$

Probabilistic Generative Models - Multiclass

- The normalised exponential is given by

$$p(\mathcal{C}_k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathcal{C}_k)\, p(\mathcal{C}_k)}{\sum_j p(\mathbf{x} \mid \mathcal{C}_j)\, p(\mathcal{C}_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

wh

- Usually called the softmax function as it is a smoo version of the $\arg\max$ function, in particul

$$a_k \gg a_j \; \forall j \neq k \Rightarrow \Big(p(\mathcal{C}_k \mid \mathbf{x}) \approx 1 \wedge$$

- So, softargmax is a more descriptive though less common name.

- Assume class-conditional probabilities are Gaussian, with the same covariance and different mean:

- Let's characterise the posterior probabilities.
- We may separate the quadratic and linear term i

$$p(\mathbf{x} \mid \mathcal{C}_k)$$

$$= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right\}$$

$$= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{x}^T \boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k\right\}$$

Probabil. Generative Model - Continuous Input

- For two classes

$$p(\mathcal{C}_1 \mid \mathbf{x}) = \sigma(a(\mathbf{x}))$$

and $a(\mathbf{x})$ is linear because the quadratic terms in $\mathbf{x}$ cancel (*c.f.* the previous slide):

$$\exp\left(\boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-} \mathbf{x} - \tfrac{1}{2}\boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-} \boldsymbol{\mu}\right) \qquad p(\ _2)$$

- Therefore

$$p(\mathcal{C}_1 \mid \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w$$

where

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^T\,\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\,\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

Probabil. Generative Model - Continuous Input

Class-conditional densities for two classes (left). Posterior probability $p(\mathcal{C}_1 \,|\, \mathbf{x})$ (right). Note the logistic sigmoid of a linear function of $\mathbf{x}$.

General Case - K Classes, Shared Covariance

- Use the normalised exponential

$$p(\mathcal{C}_k \,|\, \mathbf{x}) = \frac{p(\mathbf{x} \,|\, \mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x} \,|\, \mathcal{C}_j)p(\mathcal{C}_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

wh

- to ge

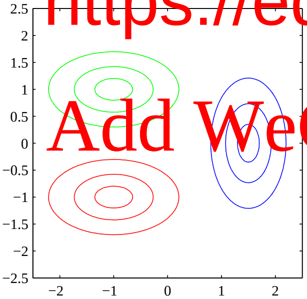
$$a_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0}.$$

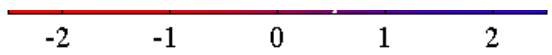
where

$$\mathbf{w}_k = \mathbf{\Sigma}^{-1} \boldsymbol{\mu}_k$$
$$w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^T \mathbf{\Sigma}^{-1} \boldsymbol{\mu}_k + p(\mathcal{C}_k).$$

General Case - K Classes, Different Covariance

- If the class-conditional distributions have different covariances, the quadratic terms $-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}$ do not cancel out.
- We get a quadratic discriminant.





Parameter Estimation

- Given the functional form of the class-conditional densities $p(\mathbf{x} \mid \mathcal{C}_k)$, how can we determine the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and the class prior?

- Given the functional form of the class-conditional densities $p(\mathbf{x} \,|\, \mathcal{C}_k)$, how can we determine the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and the class prior?
- Simplest is maximum likelihood.
- Giv
cod
$t_n =$
- Ass
with the same covariance, but different mean.
- Denote the prior probability $p(\mathcal{C}_1) = \pi$, a
$p(\mathcal{C}_2) = 1 - \pi$.
- Then

$$p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1) p(\mathbf{x}_n \,|\, \mathcal{C}_1) = \pi \, \mathcal{N}(\mathbf{x}_n \,|\, \boldsymbol{\mu}_1, \boldsymbol{\Sigma})$$
$$p(\mathbf{x}_n, \mathcal{C}_2) = p(\mathcal{C}_2) p(\mathbf{x}_n \,|\, \mathcal{C}_2) = (1 - \pi) \, \mathcal{N}(\mathbf{x}_n \,|\, \boldsymbol{\mu}_2, \boldsymbol{\Sigma})$$

Maximum Likelihood Solution

- Thus the likelihood for the whole data set $\mathbf{X}$ and $\mathbf{t}$ is given by

$$p(\mathbf{t}, \mathbf{X} \mid \pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma})$$

$$= \prod_{n=1}^{N} [\pi \, \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} \, [(1-\pi) \, \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$$

- Ma
- The t

$$\sum_{n=1}^{N} (t_n \ln \pi + (1-t_n) \ln(1$$

- which is maximal for (derive it)

$$\pi = \frac{1}{N} \sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2}$$

where N_1 is the number of data points in class $\mathcal{C}_1$.

- Similarly, we can maximise the likelihood $p(\mathbf{t}, \mathbf{X} \mid \pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma})$ w.r.t. the means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, to get

$$\boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1 - t_n)$$

- For each class, this are the means of all input vect assigned to this class.

Maximum Likelihood Solution

- Finally, the log likelihood $\ln p(\mathbf{t}, \mathbf{X} \mid \pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma})$ can be ma

$$\mathbf{S}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n$$

Discrete Features - Naïve Bayes

- Assume the input space consists of discrete features, in the simplest case $x_i \in \{0, 1\}$.
- For a D-dimensional input space, a general distribution would be represented by a table with 2^D entries.
- To
ind
- Gro

- The Naïve Bayes assumption is that, given the class $\mathcal{C}_k$, the features are independent of each other:

$$
\begin{aligned}
p(\mathbf{x} \mid \mathcal{C}_k) &= \prod_{i=1}^{D} p(x_i \mid \mathcal{C}) \\
&= \prod_{i=1}^{D} \mu_{ki}^{x_i} (1 - \mu_{ki})^{1 - x_i}
\end{aligned}
$$

Discrete Features - Naïve Bayes

- With the naïve Bayes

$$p(\mathbf{x} \mid \mathcal{C}_k) = \prod_{i=1}^{D} \mu_{ki}^{x_i}(1 - \mu_{ki})^{1-x_i}$$

- we c exp

$$p(\mathcal{C}_k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathcal{C}_k) p(\mathcal{C}_k)}{\sum_j p(\mathbf{x} \mid \mathcal{C}_j) p(\mathcal{C}_j)} = \frac{\qquad}{\qquad}$$

- as a linear function of the x_i

$$a_k(\mathbf{x}) = \sum_{i=1}^{D} \{x_i \ln \mu_{ki} + (1 - x_i) \ln(1 - \mu_{ki})\} + \ln p(\mathcal{C}_k).$$

In increasing order of complexity

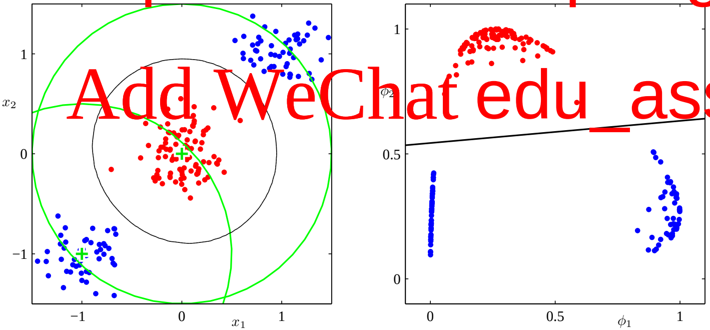
- Find a discriminant function $f(\mathbf{x})$ which maps each input directly onto a class label
- Discriminative Models
  1.
  2.
- Generative Models
  1. Solve the inference problem of determining the class-conditional probabilities $p(\mathbf{x} \mid \mathcal{C}_k)$
  2. *Also, infer the prior class probabilities*
  3. *Use Bayes theorem to find the posterior* $p(\mathcal{C}_k \mid \mathbf{x})$.
  4. Alternatively, model the joint distribution $p(\mathbf{x}, \mathcal{C}_k)$ directly.
  5. Use decision theory to assign each new $\mathbf{x}$ to one of the classes.

Probabilistic Discriminative Models

- Discriminative training: learn only to discriminate between the classes.
- Maximise a likelihood function defined through the conditional distribution $p(\mathcal{C}_k \mid \mathbf{x})$ directly.
- Ty
- As w be b class-conditional density assumptions $p(\mathbf{x} \mid \mathcal{C}_k)$ poorly approximate the true distributions.
- But: discriminative models can not create syn as $p(\mathbf{x})$ is not modelled.
- As an aside: *certain theoretical analyses show that generative models converge faster to their — albeit worse — asymptotic classification performance and are superior in some regimes.*

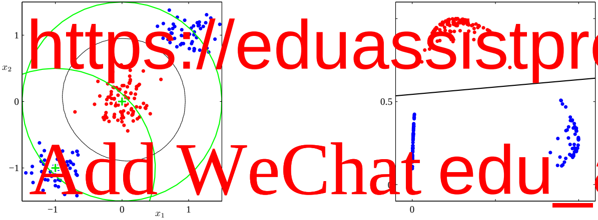
Original Input versus Feature Space

- So far in classification, we used direct input $\mathbf{x}$.
- All classification algorithms work also if we first apply a fixed nonlinear transformation of the inputs using a vector of basis functions $\phi(\mathbf{x})$.
- Example: Use two Gaussian basis functions centered at the g



Original Input versus Feature Space

- Linear decision boundaries in the feature space generally correspond to nonlinear boundaries in the input space.
- Classes which are NOT linearly separable in the input space may become linearly separable in the feature space:



- If classes overlap in input space, they will also overlap in feature space — nonlinear features $\phi(\mathbf{x})$ can not remove the overlap; but they may increase it.

Original Input versus Feature Space

- Fixed basis functions do not adapt to the data and therefore have important limitations (see discussion in Lin
- Un

eas
instead of the original input space.
- Some applications use fixed features succes avoiding the limitations

- We will therefore use ϕ instead of $\mathbf{x}$ fro

Ong & Walder & Webers
Data61 | CSIRO
The Australian National University

Logistic Regression

- Two classes where the posterior of class $\mathcal{C}_1$ is a logistic sigmoid $\sigma()$ acting on a linear function of the input:

$$p(\mathcal{C}_1 \mid \phi) = y(\phi) = \sigma(\mathbf{w}^T \phi)$$

- $p(\mathcal{C}$
- Mo
spa

- Compare this to fitting two Gaussians, which ha
quadratic number of parameters in M:

$$\underbrace{2M}_{\text{means}} + \underbrace{M(M+1)/2}_{\text{shared covariance}}$$

- For larger M, the logistic regression model has a clear advantage.

Logistic Regression

- Determine the parameter via maximum likelihood for data $(\phi_n, t_n), n = 1, \dots, N$, where $\phi_n = \phi(\mathbf{x}_n)$. The class membership is coded as $t_n \in \{0, 1\}$.
- Likelihood function

  where $y_n = p(\mathcal{C}_1 \mid \phi_n)$.
- Error function: negative log likelihood resulting in the cross-entropy error function

$$E(\mathbf{w}) = -\ln p(\mathbf{t} \mid \mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

- Error function (cross-entropy loss)

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

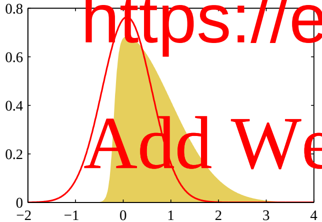
- y_n
- We
rule

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)$$

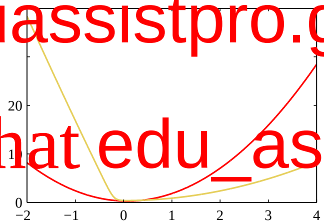
- for each data point error is product of deviation $y_n - t_n$ and basis function ϕ_n.
- We can now use gradient descent.
- We may easily modify this to reduce over-fitting by using regularised error or MAP (how?).

Laplace Approximation

- Given a continous distribution $p(x)$ which is not Gaussian, can we approximate it by a Gaussian $q(x)$?
- Need to find a mode of $p(x)$. Try to find a Gaussian with the same mode:



p.d.f. of :
Non-Gaussian (yellow) and
Gaussian approximation (red).



negative log p.d.f. of :
Non-Gaussian (yellow) and
Gaussian approxmation. (red).

Laplace Approximation

Ong & Walder & Webers
Data61 | CSIRO
The Australian National University

- Cheap and nasty but sometimes effective.
- Assume $p(x)$ can be written as

$$p(z) = \frac{1}{Z} f(z)$$

wit

- We
  app
- A mode of $p(z)$ is at a point z_0 where p'
- Taylor expansion of $\ln f(z)$ at z_0

$$\ln f(z) \simeq \ln f(z_0) - \frac{1}{2} A (z - z_0)^2$$

where

$$A = -\frac{d^2}{dz^2} \ln f(z) \mid_{z=z_0}$$

Laplace Approximation

- Exponentiating

$$\ln f(z) \simeq \ln f(z_0) - \frac{1}{2}A(z - z_0)^2$$

- we g

- And after normalisation we get the Laplace app

$$q(z) = \left(\frac{A}{2\pi}\right)^{1/2} \exp\{-\frac{A}{2}(z$$

- Only defined for precision $A > 0$ as only then $p(z)$ has a maximum.

Laplace Approximation - Vector Space

- Approximate p($\mathbf{z}$) for $z \in \mathbb{R}^M$

$$p(\mathbf{z}) = \frac{1}{Z} f(\mathbf{z})$$

- we get the Taylor expansion

- where the Hessian $\mathbf{A}$ is defined as

$$\mathbf{A} = -\nabla\nabla \ln f(\mathbf{z})\,|_{\mathbf{z}=\mathbf{z}_0}$$

- The Laplace approximation of $p(\mathbf{z})$ is th

$$q(\mathbf{z}) \propto \exp\left\{-\frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^T \mathbf{A}(\mathbf{z}-\mathbf{z}_0)\right\}$$

$$\Rightarrow q(\mathbf{z}) = \mathcal{N}(\mathbf{z}\,|\,\mathbf{z}_0, \mathbf{A}^{-1})$$

- Exact Bayesian inference for the logistic regression is intractable.
- Wh and l sig

- Eva
- Therefore we will use the Laplace approximati
- The predictive distribution remains intractabl the Laplace approximation to the posterior dis it can be approximated.

- Assume a Gaussian prior

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_0, \mathbf{S}_0)$$

for fi

- Hy
con

- For a set of training data $(\mathbf{x}_n, t_n)$, where
posterior is given by

$$p(\mathbf{w} \mid \mathbf{t}) \propto p(\mathbf{w}) p(\mathbf{t} \mid \mathbf{w}$$

where $\mathbf{t} = (t_1, \dots, t_N)^T$.

- Using our previous result for the cross-entropy function

$$E(\mathbf{w}) = -\ln p(\mathbf{t} \,|\, \mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

we c

using the notation $y_n = \sigma(\mathbf{w}^T \phi_n)$ as

$$\ln p(\mathbf{w} \,|\, \mathbf{t}) = -\frac{1}{2} (\mathbf{w} - \mathbf{m}_0)^T \mathbf{S}_0^{-1} (\mathbf{w}$$

$$+ \sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

- To obtain a Gaussian approximation to

$$\ln p(\mathbf{w} \mid \mathbf{t})$$

$$= -\frac{1}{2}(\mathbf{w} - \mathbf{m}_0)^T \mathbf{S}_0^{-1}(\mathbf{w} - \mathbf{m}_0) + \sum^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

1. nonlinear function in $\mathbf{w}$ because $y_n = \quad ^T$
2. *Calculate the second derivative of the negati to get the inverse covariance of the Laplace ap*

$$\mathbf{S}_N = -\nabla\nabla \ln p(\mathbf{w} \mid \mathbf{t}) = \mathbf{S}_0^{-1} + \sum_{n=1}^{N} \quad {}_n \quad - \quad {}_n \quad {}_n \quad {}_n$$

Nowadays the gradient and Hessian would be computed with automatic differentiation; one need only implement $\ln p(\mathbf{w} \mid \mathbf{t})$.

Bayesian Logistic Regression

- The approximated Gaussian (via Laplace approximation) of the posterior distribution is now

wh

$$\mathbf{S}_N = -\nabla\nabla \ln p(\mathbf{w} \mid \mathbf{t}) = \mathbf{S}_0^{-1} + \sum_{n=1}^{N}$$