Probabilistic Models for Supervised Learning (contd)

CS771: Introduction to Machine Learning

Prob. Models for Supervised Learning

Non-probabilistic supervised learning approaches (e.g., SVM) are usually considered discriminative since $p(\boldsymbol{x})$ is never modeled

- Goal: Learn the conditional distribution $p(y|\boldsymbol{x})$. Broadly, two approaches

Discriminative Approach

$$p(y|\boldsymbol{x}) = p(y|f(\boldsymbol{x}, \boldsymbol{w}))$$

f can be any function which uses inputs and weights $\boldsymbol{w}$ to defines parameters of distr. p

Some examples

$$p(y|\boldsymbol{x}) = \mathcal{N}(y|\boldsymbol{w}^\top \boldsymbol{x}, \beta^{-1})$$

$$p(y|\boldsymbol{x}) = \text{Bernoulli}(y|\sigma(\boldsymbol{w}^\top \boldsymbol{x}))$$

Generative Approach

$$p(y|\boldsymbol{x}) = \frac{p(y, \boldsymbol{x})}{p(\boldsymbol{x})}$$

Requires estimating the joint distribution of inputs and outputs to get the conditional $p(y|\boldsymbol{x})$ (unlike the discriminative approach which directly estimates the conditional $p(y|\boldsymbol{x})$ and does not model the distribution of $\boldsymbol{x}$)

- Both approaches have their pros and cons (discussed later)

The Discriminative Approach

- This approach models $\boldsymbol{p(y|x)}$ directly using a suitable prob. distribution, e.g.,

Regression model likelihood

$$p(y_i|\boldsymbol{x}_i,\boldsymbol{w}) = \mathcal{N}(y_i|\boldsymbol{w}^\top\boldsymbol{x}_i,\beta^{-1}) = \sqrt{\frac{\beta}{2\pi}}\exp\left[-\frac{\beta}{2}(y_i-\boldsymbol{w}^\top\boldsymbol{x}_i)^2\right]$$

Binary classification model likelihood

$\mu_i = \sigma(\boldsymbol{w}^\top\boldsymbol{x}_i)$

$$p(y_i|\boldsymbol{x}_i,\boldsymbol{w}) = \text{Bernoulli}(y_i|\sigma(\boldsymbol{w}^\top\boldsymbol{x}_i)) = \mu_i^{y_i}(1-\mu_i)^{1-y_i}$$

- The negative log-likelihood (assuming i.i.d. outputs) for the above two models

$$NLL(\boldsymbol{w}) = \sum_{i=1}^{N} -\log p(y_i|\boldsymbol{x}_i,\boldsymbol{w}) = \sum_{i=1}^{N} -\log\sqrt{\frac{\beta}{2\pi}}\exp\left[-\frac{\beta}{2}(y_i-\boldsymbol{w}^\top\boldsymbol{x}_i)^2\right]$$

$NLL(\boldsymbol{w}) = \frac{\beta}{2}\sum_{n=1}^{N}(y_i-\boldsymbol{w}^\top\boldsymbol{x}_i)^2$ (same as squared loss ☺)

$$NLL(\boldsymbol{w}) = \sum_{i=1}^{N} -\log p(y_i|\boldsymbol{x}_i,\boldsymbol{w}) = \sum_{i=1}^{N} -\log \mu_i^{y_i}(1-\mu_i)^{1-y_i}$$

$NLL(\boldsymbol{w}) = -\sum_{n=1}^{N}[y_i\log\mu_i + (1-y_i)\log(1-\mu_i)]$ (same as cross-entropy loss ☺)

Thus minimization of NLL (i.e., doing MLE) is equivalent to minimization of the training loss

Lacks regularization (and thus can overfit) but we can use a prior on $\boldsymbol{w}$ to do MAP estimation

A prior over $\boldsymbol{w}$

- For MAP estimation (and to compute full posterior), we need a prior $\boldsymbol{w} \in \mathbb{R}^D$
- A reasonable prior for real-valued vectors can be a multivariate Gaussian

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_0, \boldsymbol{\Sigma})$$

Equivalent to saying that *a priori* we expect the solution to be close to some vector $\boldsymbol{w}_0$

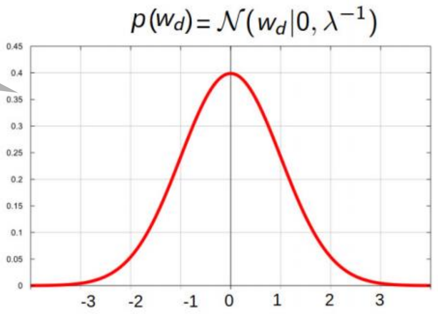
Σ specifies how strong our belief is that $\boldsymbol{w}$ to close to $\boldsymbol{w}_0$

- A specific example of a multivariate Gaussian prior in this problem

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \lambda^{-1}\boldsymbol{I}_D) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda^{-1}) = \prod_{d=1}^{D} p(w_d)$$

Omitting λ for brevity

The precision λ of the Gaussian prior controls how aggressively the prior pushes the elements towards mean (0)



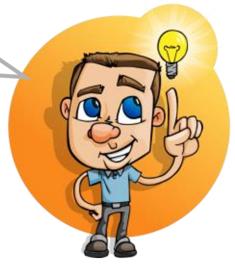
This is essentially like a regularizer that pushes elements of $\boldsymbol{w}$ to be small (we will see shortly)

Equivalent to saying that *a priori* we expect each element of the solution to be close to 0 (i.e., "small")

$$\mathcal{N}(w_d|0, \lambda^{-1}) = \sqrt{\frac{\lambda}{2\pi}} \exp\left[-\frac{\lambda}{2} w_d^2\right]$$

$$\mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \lambda^{-1}\boldsymbol{I}_D) = \left(\frac{\lambda}{2\pi}\right)^{D/2} \exp\left[-\frac{\lambda}{2}\sum_{d=1}^{D} w_d^2\right] = \left(\frac{\lambda}{2\pi}\right)^{D/2} \exp\left[-\frac{\lambda}{2}\boldsymbol{w}^\top\boldsymbol{w}\right]$$

Aha! This $\boldsymbol{w}^\top\boldsymbol{w}$ term reminds me of the ℓ_2 regularizer ☺



That's indeed the case ☺

MAP Estimation with $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0, \lambda^{-1}\boldsymbol{I}_D)$

- The MAP objective (log-posterior) will be the log-likelihood + $\log p(\boldsymbol{w})$
- Ignoring terms that don't depend on $\boldsymbol{w}$ the log-posterior will be

$$NLL(\boldsymbol{w}) - \log\exp\left[-\frac{\lambda}{2}\boldsymbol{w}^\top\boldsymbol{w}\right] = NLL(\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^\top\boldsymbol{w}$$

- Exercise: For regression with likelihood $p(y_i|\boldsymbol{x}_i, \boldsymbol{w}) = \mathcal{N}(y_i|\boldsymbol{w}^\top\boldsymbol{x}_i, \beta^{-1})$

$$\hat{w}_{MAP} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \frac{\beta}{2}\sum_{i=1}^{N}(y_i - \boldsymbol{w}^\top\boldsymbol{x}_i)^2 + \frac{\lambda}{2}\boldsymbol{w}^\top\boldsymbol{w} = (\boldsymbol{X}^\top\boldsymbol{X} + \frac{\lambda}{\beta}\boldsymbol{I}_D)^{-1} \boldsymbol{X}^\top\boldsymbol{y}$$

- Classification with likelihood $p(y_i|\boldsymbol{x}_i, \boldsymbol{w}) = \text{Bernoulli}(y_i|\sigma(\boldsymbol{w}^\top\boldsymbol{x}_i))$ is the same as logistic regression. When also using the above prior, the MAP solution will be the same as cross-entropy loss minimization + L2 regularization on the weights

Prob. Linear Regression: The Full Posterior

- If we want the full posterior distribution over $\boldsymbol{w}$, it can be computed as

$$p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}) = \frac{p(\boldsymbol{w})p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})}{p(\boldsymbol{y}|\boldsymbol{X})}$$

For brevity, we have not shown the dependence of the various distributions here on the hyperparameters λ and β

- For regression, with Gaussian likelihood and zero mean Gaussian prior are conjugate (both Gaussians), and the posterior will be Gaussian

Will provide the proof separately but it is straightforward when using properties of Gaussian

$$p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$$

$$\boldsymbol{\mu}_N = (\boldsymbol{X}^\top\boldsymbol{X} + \frac{\lambda}{\beta}\boldsymbol{I}_D)^{-1}\boldsymbol{X}^\top\boldsymbol{y}$$

$$\boldsymbol{\Sigma}_N = (\beta\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I}_D)^{-1}$$

Posterior's mean is the same as the MAP solution since the mean and mode of a Gaussian are the same!

Note: λ and β are assumed to be fixed; otherwise, the problem is a bit harder (beyond the scope of CS771)

- For binary classification with Bernoulli likelihood and zero mean Gaussian prior, we don't have conjugacy, and the posterior can't be computed in closed form

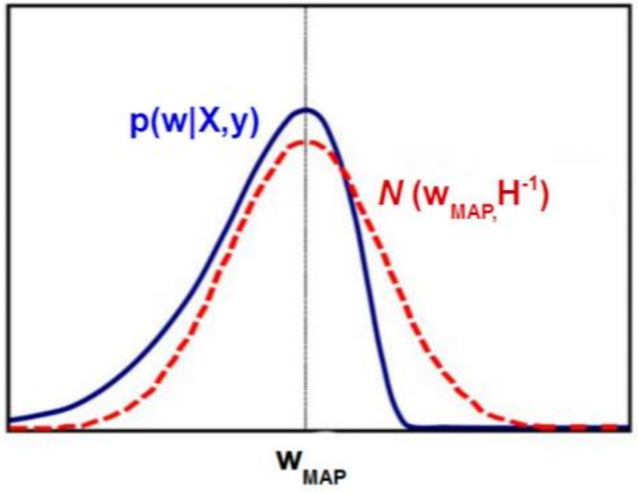
Logistic Regression: The Full Posterior

- The posterior distribution for the logistic regression model

Unfortunately, Gaussian and Bernoulli are not conjugate with each other, so analytic expression for the posterior can't be obtained unlike prob. linear regression

$$p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}) = \frac{p(\boldsymbol{w})p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})}{p(\boldsymbol{y}|\boldsymbol{X})} = \frac{\overbrace{p(\boldsymbol{w})}^{\text{Gaussian}} \prod_{n=1}^{N} \overbrace{p(y_n|\boldsymbol{w},\boldsymbol{x}_n)}^{\text{Bernoulli}}}{\int p(\boldsymbol{w}) \prod_{n=1}^{N} p(y_n|\boldsymbol{w},\boldsymbol{x}_n)\, d\boldsymbol{w}}$$

- Need to approximate the posterior in this case
- We will use a simple approximation called Laplace approximation

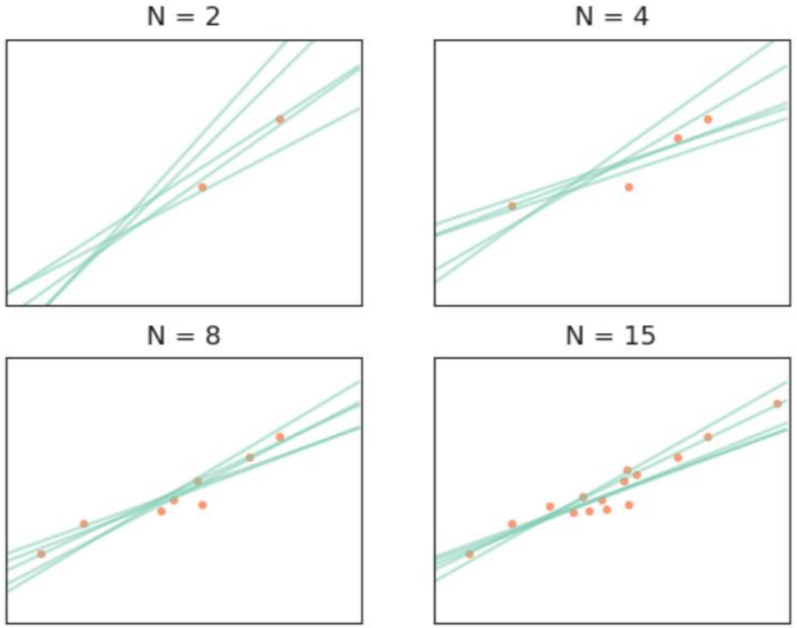


Approximates the posterior of $\boldsymbol{w}$ by a Gaussian whose mean is the MAP solution $\hat{\boldsymbol{w}}_{MAP}$ and covariance matrix is the inverse of the Hessian (Hessian: second derivative of the negative log-posterior of the LR model)

Can also employ more advanced posterior approximation methods, like MCMC and variational inference (beyond the scope of CS771)

What does posterior of a linear model look like ?

- Each sample from posterior $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$ will give a weight vector $\boldsymbol{w}$
  - In case of lin. reg., each weight vector corresponds to a regression line



The posterior sort of represents an ensemble of solutions (not all are equally good but we can use all of them in an “importance-weighted” fashion to make the prediction using the posterior predictive distribution)

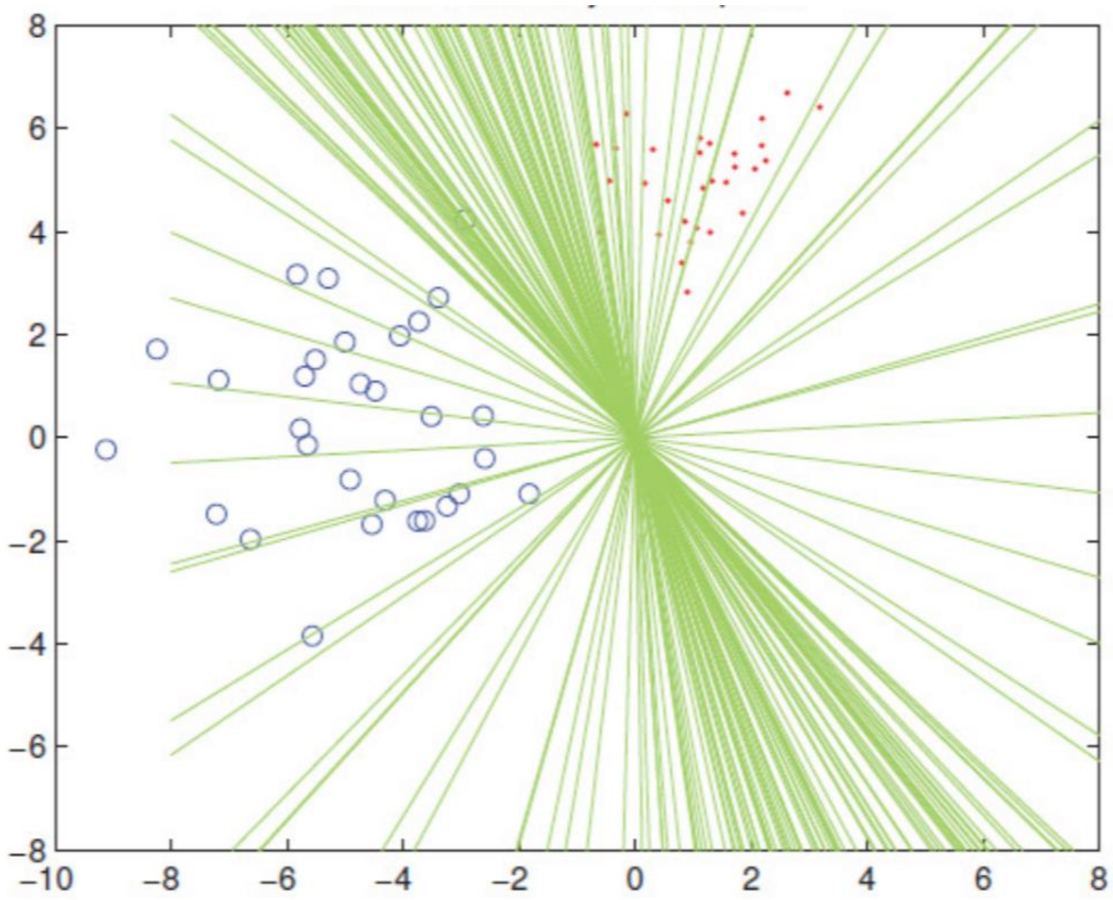
Importance of each solution in this ensemble is its posterior probability $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y})$

- Each weight vector will give a different set of predictions on test data
  - These different predictions will give us a variance (uncertainty) estimate in model’s prediction
  - The uncertainty decreases as N increases (we become more sure when we see more training data)

Pic source: https://waterprogramming.wordpress.com/

What does posterior of a linear model look like ?

- Can also sample from the Laplace approximate posterior of logistic regression
- Each sample will give a weight vec defining a hyperplane separator



Not all separators are equally good; their goodness depends on their posterior probabilities

When making predictions, we can still use all of them but weighted by their importance based on their posterior probabilities

That's exactly what we do when computing the predictive distribution

Prob. Linear Regression: Predictive Distribution

- Want the predictive distribution $p(y_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y})$ of the output y_* for a new input $\boldsymbol{x}_*$
- With MLE/MAP estimate of $\boldsymbol{w}$, we will use the plug-in predictive

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) \approx p(y_*|\mathbf{x}_*, \mathbf{w}_{MLE}) = \mathcal{N}(\mathbf{w}_{MLE}^\top \mathbf{x}_*, \beta^{-1})$$ - MLE prediction

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) \approx p(y_*|\mathbf{x}_*, \mathbf{w}_{MAP}) = \mathcal{N}(\mathbf{w}_{MAP}^\top \mathbf{x}_*, \beta^{-1})$$ - MAP prediction

- If we have the full posterior over $\boldsymbol{w}$, can compute the posterior predictive dist.

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(y_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|\mathbf{X}, \mathbf{y})d\mathbf{w}$$

Assuming the hyperparameters λ and β are known (otherwise, the PPD can't be computed exactly)

- Requires an integral but has a closed form

Mean prediction

Will provide the proof separately but it is straightforward when using properties of Gaussian

$$\boxed{p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_N^\top \mathbf{x}_*, \beta^{-1} + \mathbf{x}_*^\top \boldsymbol{\Sigma}_N \mathbf{x}_*)}$$

Input-specific predictive variance unlike the MLE/MAP based predictive where it was β^{-1} (and was same for all test inputs)

- Input-specific predictive uncertainty useful in problems where we want confidence estimates of the predictions made by the model (e.g., Active Learning)

Logistic Regression: Predictive Distribution

- When using MLE/MAP solution $\hat{\boldsymbol{w}}_{opt}$, can use the plug-in predictive distribution

$$p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \int p(y_* = 1|\boldsymbol{w}, \boldsymbol{x}_*)p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y})d\boldsymbol{w}$$
$$\approx p\left(y_* = 1\middle|\hat{\boldsymbol{w}}_{opt}, \boldsymbol{x}_*\right) = \sigma(\hat{\boldsymbol{w}}_{opt}{}^{\top}\boldsymbol{x}_n)$$

$$p(y_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \text{Bernoulli}[\sigma(\hat{\boldsymbol{w}}_{opt}{}^{\top}\boldsymbol{x}_n)]$$

- When using fully Bayesian inference, we must compute the posterior predictive

$$p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \int p(y_* = 1|\boldsymbol{w}, \boldsymbol{x}_*)p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y})d\boldsymbol{w}$$

Integral not tractable and must be approximated

sigmoid

Gaussian (if using Laplace approx.)

Monte-Carlo approximation of this integral is one possible way

Generate M samples $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_M$, from the Gaussian approx. of posterior and use $p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) \approx \frac{1}{M}\sum_{m=1}^{M} p(y_* = 1|\boldsymbol{w}_{\mathrm{m}}, \boldsymbol{x}_*) = \frac{1}{M}\sum_{m=1}^{M} \sigma(\boldsymbol{w}_m^{\top}\boldsymbol{x}_n)$