ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Introduction to Machine Learning

A statistical perspective on supervised learning

Prof. Andreas Krause
Learning and Adaptive Systems (las.ethz.ch)

Motivation

- We have seen how we can fit prediction models (linear, non-linear) for regression and classification
- So far, these models do not have any statistical interpretation
- Often we would like to statistically model the data:
  - Quantify uncertainty
  - Express prior knowledge / assumptions about the data
- In the following, we will see how many of the approaches we have discussed can be interpreted as fitting probabilistic models
- This view will allow us to derive new methods

Minimizing the least squares error

- Assuming the data is generated iid according to

$$(\mathbf{x}_i, y_i) \sim P(\mathbf{X}, Y)$$

- The hypothesis h* minimizing $R(h) = \mathbb{E}_{\mathbf{x},y}[(y - h(\mathbf{x}))^2]$ is given by the conditional mean

$$h^*(\mathbf{x}) = \mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}]$$

- This (in practice unattainable) hypothesis is called the Bayes‘ optimal predictor for the squared loss

In practice we have finite data

- We know that

$$h^*(\mathbf{x}) = \mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}]$$

- Thus, one strategy for estimating a predictor from training data is to estimate the conditional distribution

$$\hat{P}(Y \mid \mathbf{X})$$

and then, for test point ***x***, predict label

$$\hat{y} = \hat{\mathbb{E}}[Y \mid \mathbf{X} = \mathbf{x}] = \int \hat{P}(y \mid \mathbf{X} = \mathbf{x})\, y\, dy$$

Estimating conditional distributions

- Common approach: Parametric estimation
  - Choose a particular parametric form $\hat{P}(Y \mid \mathbf{X}, \theta)$
  - Then optimize the parameters. How?

➔ Maximum (conditional) Likelihood Estimation

$$\theta^* = \arg\max_{\theta} \hat{P}(y_1, \ldots, y_n \mid \mathbf{x}_1, \ldots, \mathbf{x}_n, \theta)$$

Ridge regression = MAP estimation

- Ridge regression can be understood as finding the Maximum A Posteriori (MAP) parameter estimate for a linear regression problem, assuming that
  - The noise *P(y|**x**,**w**)* is iid Gaussian and
  - The prior *P(**w**)* on the model parameters ***w*** is Gaussian

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \lambda ||\mathbf{w}||_2^2 \equiv \arg\max_{\mathbf{w}} P(\mathbf{w}) \prod_i P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

Regularization vs. MAP inference

- More generally, regularized estimation can often be understood as MAP inference

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} \ell(\mathbf{w}^T\mathbf{x}_i; \mathbf{x}_i, y_i) + C(\mathbf{w}) = \arg\max_{\mathbf{w}} \prod_i P(y_i \mid \mathbf{x}_i, \mathbf{w}) P(\mathbf{w})$$

$$= \arg\max_{\mathbf{w}} P(\mathbf{w} \mid D)$$

where $C(\mathbf{w}) = -\log P(\mathbf{w})$

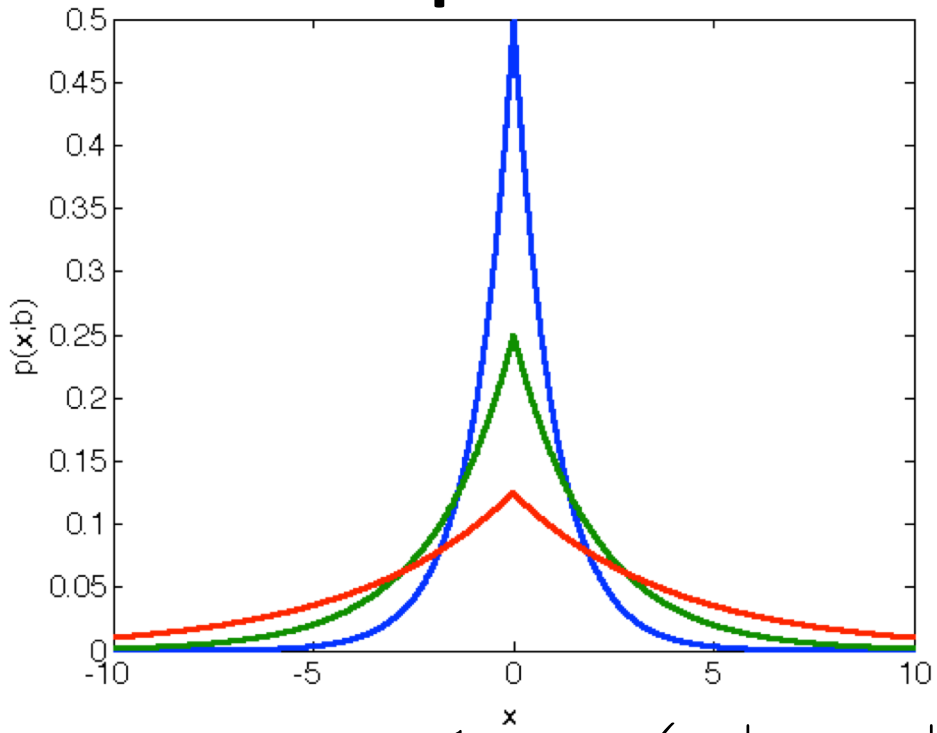
and $\ell(\mathbf{w}^T\mathbf{x}_i; \mathbf{x}_i, y_i) = -\log P(y_i \mid \mathbf{x}_i, \mathbf{w})$

- This perspective allows changing priors (=regularizers) and likelihoods (=loss functions)

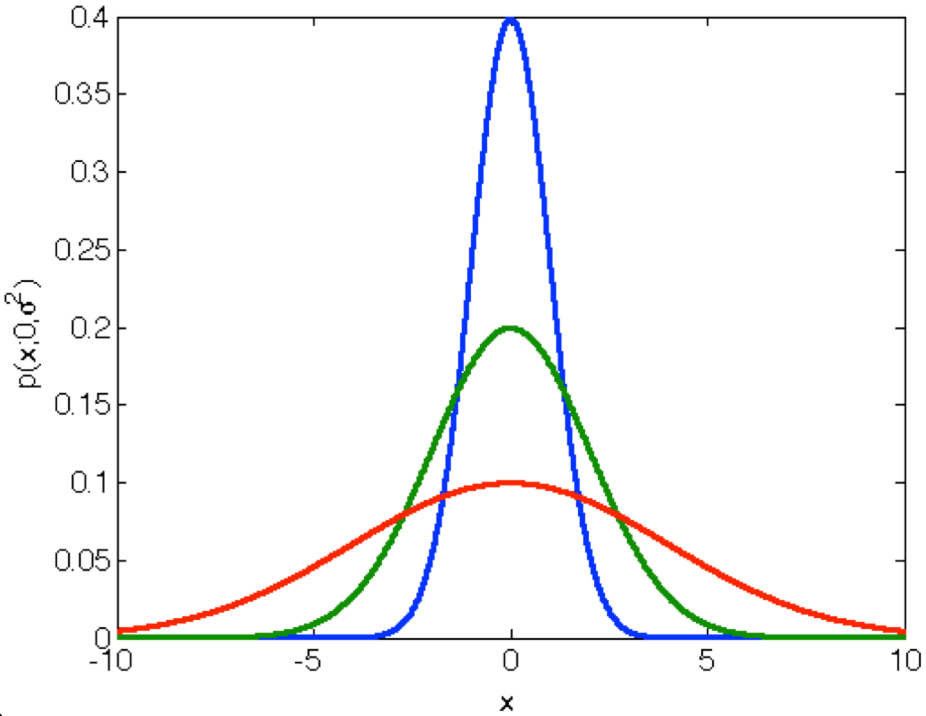
Example: l1-regularization

- Is there a prior that corresponds to l1-regularization?
- **Answer**: The Laplace prior

Laplace



Compare with **Gaussian**

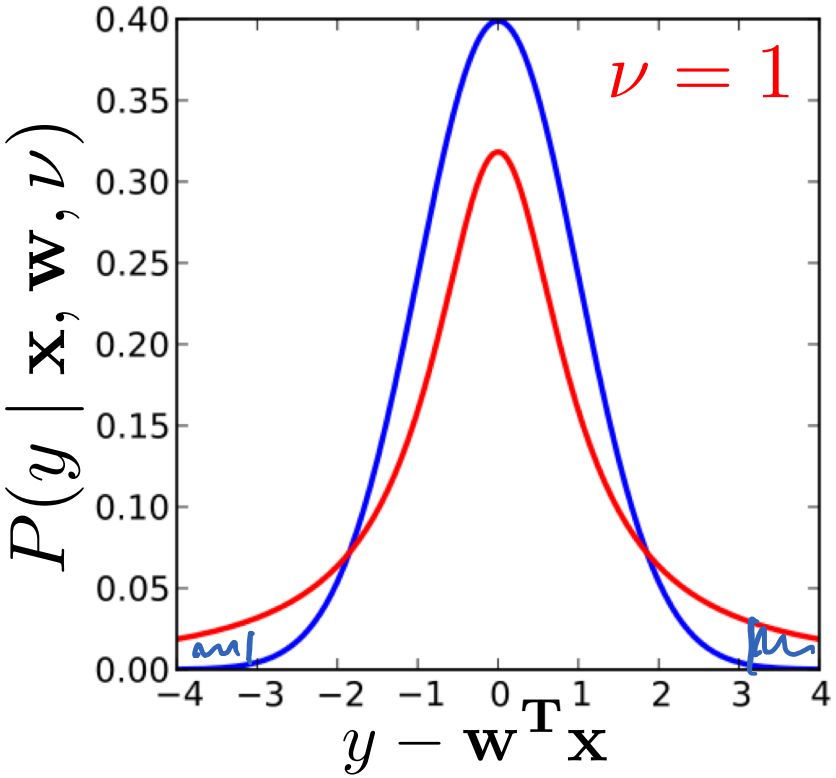


$$p(x;\mu,b) = \frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right)$$

Example: student-t likelihood

- Can introduce robustness by changing the likelihood (=loss) function
- **Example**: (non-standardized) Student's-t likelihood

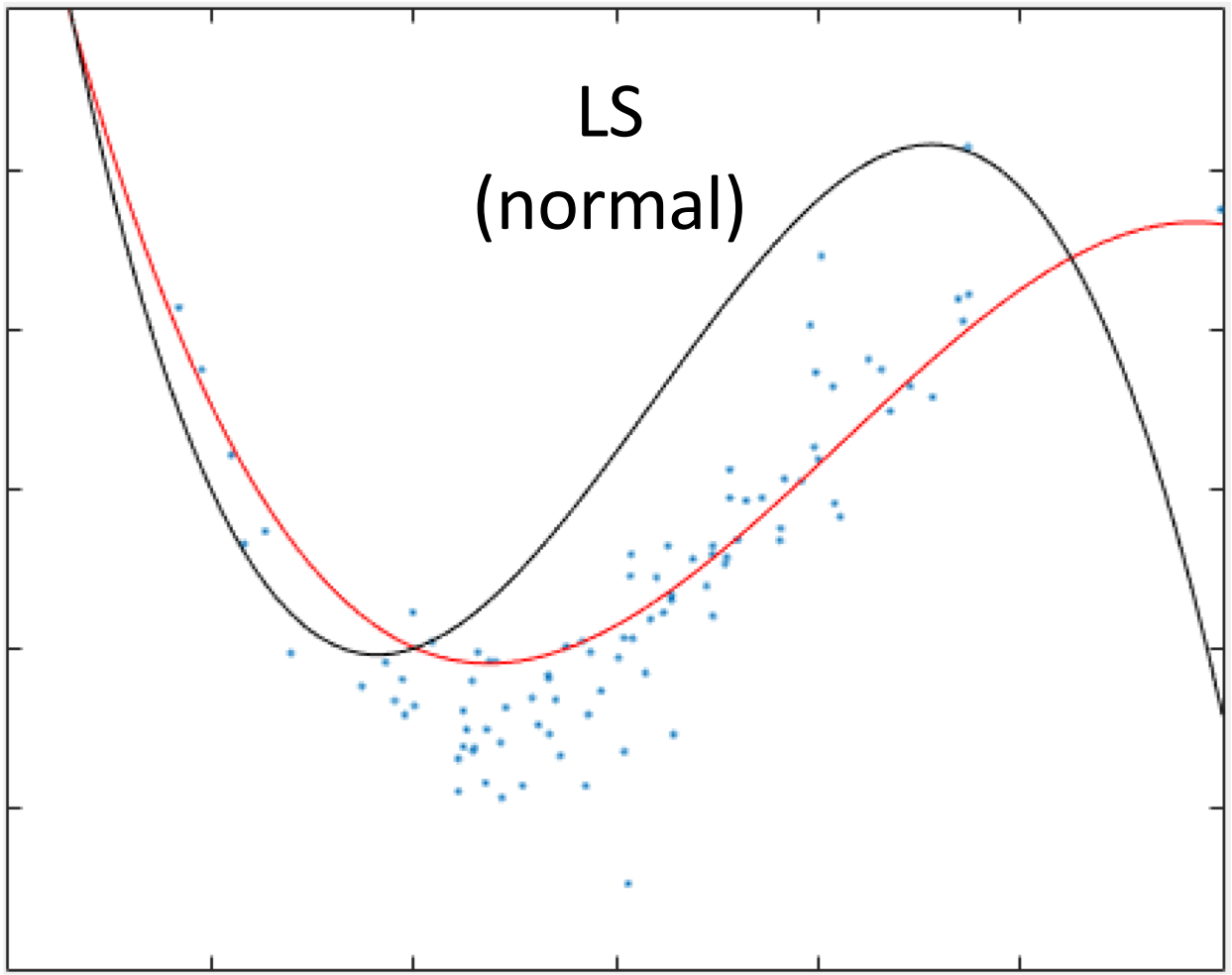
$$P(y \mid \mathbf{x}, \mathbf{w}, \nu, \sigma^2) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi\nu\sigma^2}\Gamma(\frac{\nu}{2})}\left(1 + \frac{(y - \mathbf{w}^T\mathbf{x})^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}$$



For Gaussian: $P(|y - w^T x| > t \cdot \sigma^2) = O(\exp -t)$

For student-t: $\sim$ " —— $= O(t^{-\nu})$

Example fits



Student-t
(ν=2)

Statistical models for classification

- So far, we have focused on *regression*
- Are there natural statistical models for *classification*?

Risk in classification

- In classification, risk is

$$R(h) = \mathbb{E}_{\mathbf{X},Y}[[Y \neq h(\mathbf{X})]]$$

$$\begin{cases} 1 & \text{if } Y \neq h(x) \\ 0 & \text{otw.} \end{cases}$$

- Suppose (unrealistically) we *knew* $P(\mathbf{X}, Y)$
- Which h minimizes the risk then?

$$h^*(x) = \underset{\hat{y}}{\operatorname{argmin}} \underbrace{\mathbb{E}_Y[[Y \neq \hat{y}] \mid X = x]}_{\ell(\hat{y})} \quad = (*)$$

$$\ell(\hat{y}) = \sum_{y=1}^{C} P(Y = y \mid X = x)[y \neq \hat{y}] = \sum_{y : y \neq \hat{y}} P(Y = y \mid X = x) = 1 - P(Y = \hat{y} \mid X = x)$$

$$(*) = \underset{\hat{y}}{\operatorname{argmax}} \; P(Y = \hat{y} \mid X = x)$$

Bayes‘ optimal *classifier*

- Assuming the data is generated iid according to

$$(\mathbf{x}_i, y_i) \sim P(\mathbf{X}, Y)$$

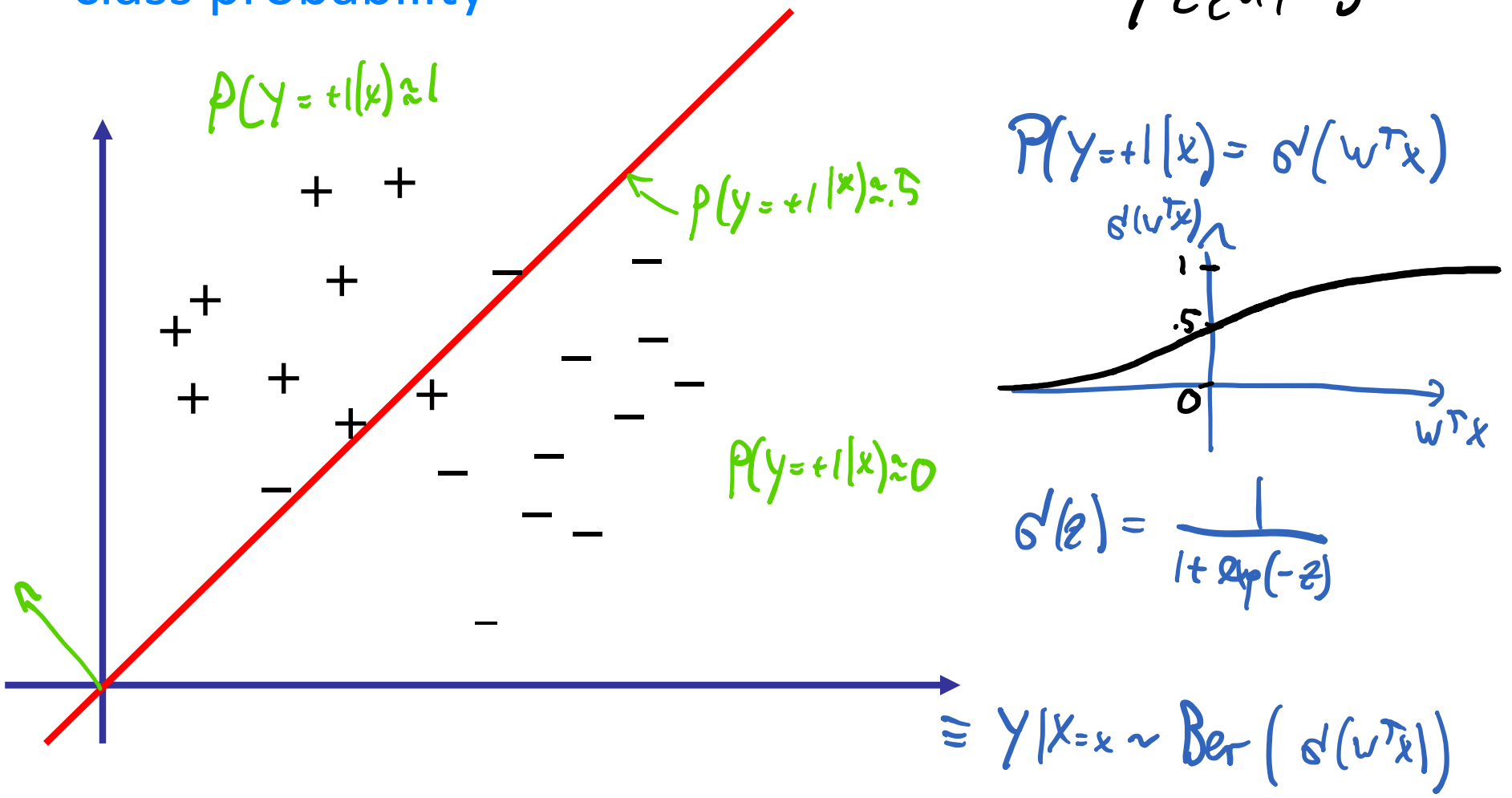
- The hypothesis h* minimizing $R(h) = \mathbb{E}_{\mathbf{X},Y}[[Y \neq h(\mathbf{X})]]$ is given by the most probable class

$$h^*(x) = \arg\max_y P(Y = y \mid \mathbf{X} = \mathbf{x})$$

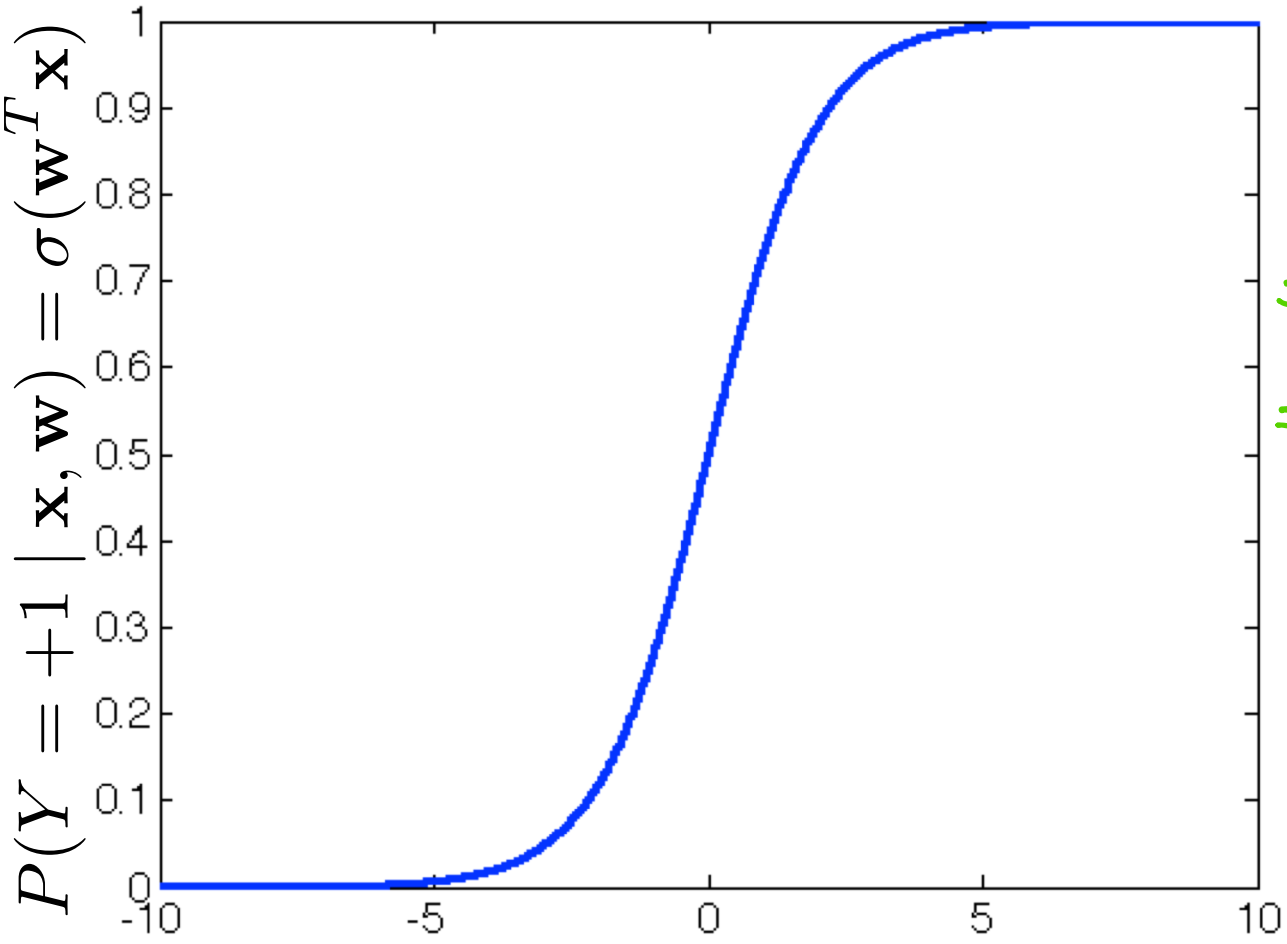
- This (in practice unattainable) hypothesis is called the Bayes‘ optimal predictor for the squared loss
- Thus, natural approach is again to estimate P(Y|X)

Logistic regression

- **Idea**: Use (generalized) linear model for the class probability



Link function for logistic regression



$P(Y=-1|x)$

$= 1 - P(Y=+1|x)$

$= 1 - \frac{1}{1+\exp(-w^Tx)}$

$= \frac{\exp(-w^Tx)}{1+\exp(-w^Tx)}$

$= \frac{1}{1+\exp(w^Tx)}$

$\Rightarrow P(Y=y|x) = \frac{1}{1+\exp(-y \cdot w^Tx)}$

- Link function

$$\sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})}$$

Logistic regression

- Logistic regression (a classification method) replaces the assumption of Gaussian noise (squared loss) by iid Bernoulli noise:

$$P(y \mid \mathbf{x}, \mathbf{w}) = \mathrm{Ber}(y; \sigma(\mathbf{w}^T \mathbf{x}))$$

vs. $P(y \mid x, w) = \mathcal{N}(y; w^T x, \sigma^2)$ for LS regression

- How can we estimate the parameters ***w***?

➔ Maximum Likelihood Estimation / MAP estimation

MLE for logistic regression

$$\hat{w} \in \arg\max_w P(D|w) \overset{iid}{=} \arg\max_w \prod_{i=1}^n P(y_i|x_i,w)$$

$$= \arg\min_w -\sum_{i=1}^n \log P(y_i|x_i,w) \quad (*)$$

$$P(y_i|x_i,w) = \frac{P(y_i,x_i|w)}{P(x_i|w)} = C_i\, P(y_i,x_i|w)$$

if $P(x_i|w) = P(x_i)$ i.e. x_i indep of w

$$-\log P(y|x,w) = -\log \frac{1}{1+\exp(-yw^Tx)} = \log\left(1+\exp(-yw^Tx)\right)$$

$$(*) = \arg\min_w \sum_{i=1}^n \log\left(1+\exp(-y_i w^T x_i)\right)$$

$$\ell_{logistic}(w; x_i, y_i)$$

$$\hat{R}(w)$$

MLE for logistic regression

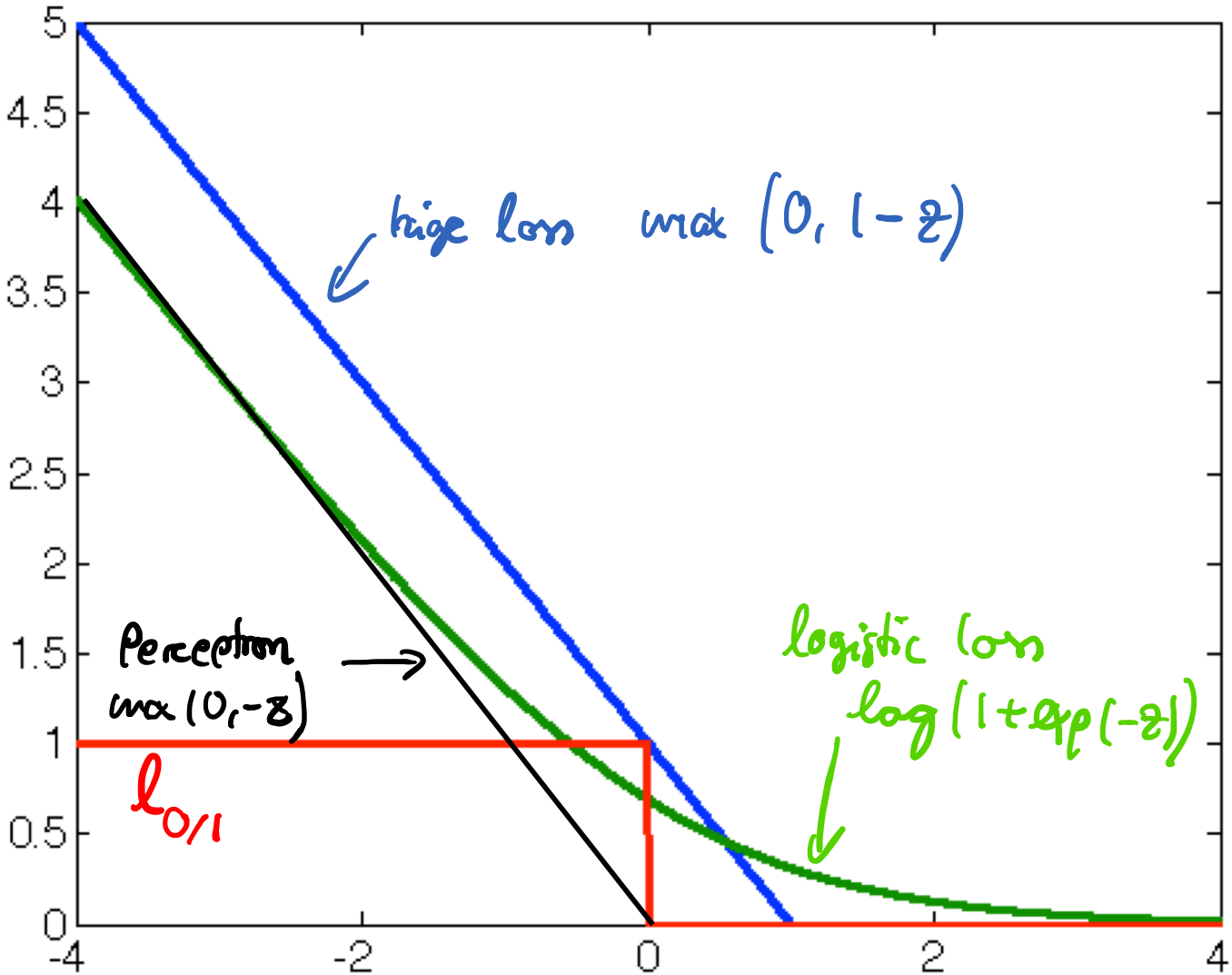
- Negative log likelihood (=objective) function given by

$$\hat{R}(\mathbf{w}) = \sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$$

- The logistic loss is convex

➔ Can use convex optimization techniques (e.g., SGD)

Logistic loss vs other losses



for very large and very small z, logistic ≈ perceptron

Gradient for logistic regression

- Loss for data point (**x**,y)

$$\ell(\mathbf{w}) = \log\left(1 + \exp(-y\mathbf{w}^T\mathbf{x})\right)$$

$$\nabla_w \ell(w) = \frac{1}{1+\exp(-y\,w^Tx)} \cdot \exp(-y\,w^Tx) \cdot (-y \cdot x)$$

$$= \frac{\exp(-y\,w^Tx)}{1+\exp(-y\,w^Tx)} (-y \cdot x)$$

$$= \underbrace{\frac{1}{1+\exp(y \cdot w^Tx)}}_{P(Y \neq y|x)} \cdot (-yx)$$

SGD for logistic regression

- Initialize **w**
- For t = 1,2,...
  - Pick data point ***(x,y)*** uniformly at random from data D
  - Compute probability of misclassification with current model

$$\hat{P}(Y = -y \mid \mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(y\mathbf{w}^T\mathbf{x})}$$

  - Take gradient step

$$\mathbf{w} \leftarrow \mathbf{w} + \eta_t \, y \, \mathbf{x} \, \hat{P}(Y = -y \mid \mathbf{w}, \mathbf{x})$$

Logistic regression and regularization

- Similar to SVMs and linear regression, want to use **regularizer** to control model complexity
- Thus, instead of solving MLE

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \log\Big(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\Big)$$

  estimate MAP/solve regularized problem

- **L2 (Gaussian prior)**:

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \log\Big(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\Big) + \lambda ||\mathbf{w}||_2^2$$

- **L1 (Laplace)**:

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \log\Big(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\Big) + \lambda ||\mathbf{w}||_1$$

SGD for L2-regularized logistic regression

- Initialize **w**
- For t = 1,2,...
  - Pick data point ***(x,y)*** uniformly at random from data D
  - Compute probability of misclassification with current model

$$\hat{P}(Y = -y \mid \mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(y\mathbf{w}^T\mathbf{x})}$$

  - Take gradient step

$$\mathbf{w} \leftarrow \mathbf{w}(1 - 2\lambda\eta_t) + \eta_t \, y \, \mathbf{x} \, \hat{P}(Y = -y \mid \mathbf{w}, \mathbf{x})$$

Regularized logistic regression

- Learning:
  - Find optimal weights by minimizing logistic loss + regularizer

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right) + \lambda ||\mathbf{w}||_2^2$$

$$= \arg\max_{\mathbf{w}} P(\mathbf{w} \mid \mathbf{x}_1, \ldots, \mathbf{x}_n, y_1, \ldots, y_n)$$

- Classification:
  - Use conditional distribution

$$P(y \mid \mathbf{x}, \hat{\mathbf{w}}) = \frac{1}{1 + \exp(-y \hat{\mathbf{w}}^T \mathbf{x})}$$

  - E.g., predict more likely class label

(*) $\hat{y} = \text{Sign}(\hat{w}^T x)$

(*) $\operatorname{argmax}_{\hat{y}} P(\hat{y} \mid x, \hat{w})$

$= \operatorname{argmax}_{\hat{y}} \frac{1}{1 + \exp(-\hat{y} \hat{w}^T x)}$

$= \operatorname{argmin}_{\hat{y}} \exp(-\hat{y} \hat{w}^T x)$

$= \operatorname{argmin}_{\hat{y}} -\hat{y} \hat{w}^T x$

$= \operatorname{argmax}_{\hat{y} \in \{+1, -1\}} \hat{y} w^T x = \text{Sign}(w^T x)$

Logistic regression demo

More remarks on logistic regression

- Can kernelize (kernelized logistic regression)
- Can apply logistic loss function to neural networks, in order to have them output probabilities
- Natural multi-class variants
- ...

Kernelized logistic regression

- Learning:
  - Find optimal weights by minimizing logistic loss + regularizer

$$\hat{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \alpha^T \mathbf{K}_i\right)\right) + \lambda \alpha^T \mathbf{K} \alpha$$

$K = \left(k_1 \middle| \ldots \middle| k_n\right)$

- Classification:
  - Use conditional distribution

$$\hat{P}(y \mid \mathbf{x}, \hat{\alpha}) = \frac{1}{1 + \exp\left(-y \sum_{j=1}^{n} \alpha_j k(\mathbf{x}_j, \mathbf{x})\right)}$$

  - E.g., predict more likely class label

$w^T x$ assuming $w = \sum_i \alpha_i x_i$

Multi-class logistic regression

- Can extend logistic regression to multi-class setting
- Maintain one weight vector per class and model

$$P(Y = i \mid \mathbf{x}, \mathbf{w}_1, \ldots, \mathbf{w}_c) = \frac{\exp(\mathbf{w}_i^T \mathbf{x})}{\sum_{j=1}^{c} \exp(\mathbf{w}_j^T \mathbf{x})}$$

$\sum_{i=1}^{c} \hat{p}_i = 1$

$\hat{p}_i$

$$\frac{\exp(w_i^T x + z)}{\sum_j \exp(w_j^T x + z)} = \frac{\exp(w_i^T x) \cdot e^z}{\sum_j \exp(w_j^T x) \cdot e^z}$$

$\Rightarrow$ w.l.o.g., can pick z to be $-w_c^T x$

$\hookrightarrow$ $w_i \mapsto w_i - w_c$, and $w_c \mapsto 0$

Multi-class logistic regression

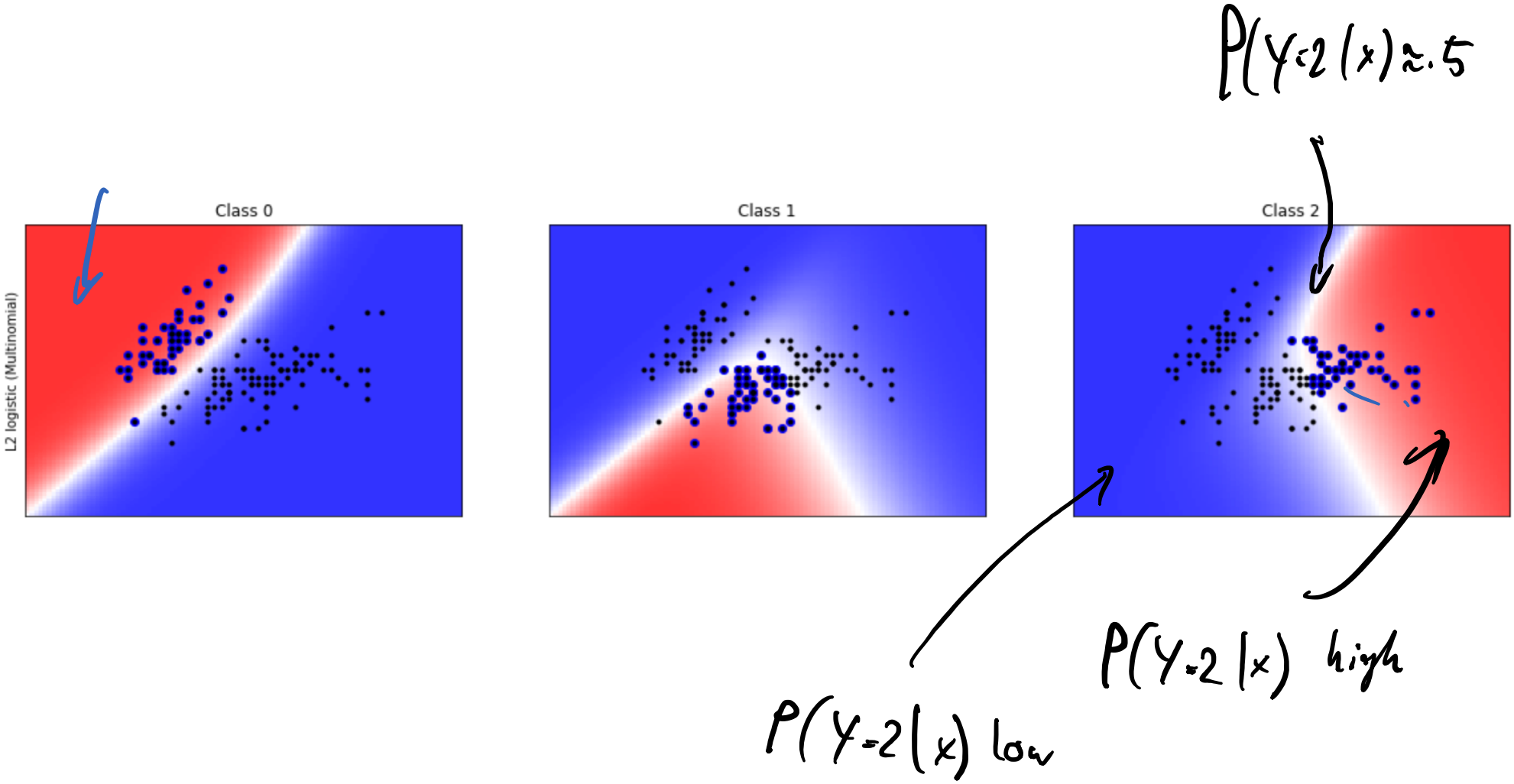
- Maintain one weight vector per class and model

$$P(Y = i \mid \mathbf{x}, \mathbf{w}_1, \ldots, \mathbf{w}_c) = \frac{\exp(\mathbf{w}_i^T \mathbf{x})}{\sum_{j=1}^{c} \exp(\mathbf{w}_j^T \mathbf{x})}$$

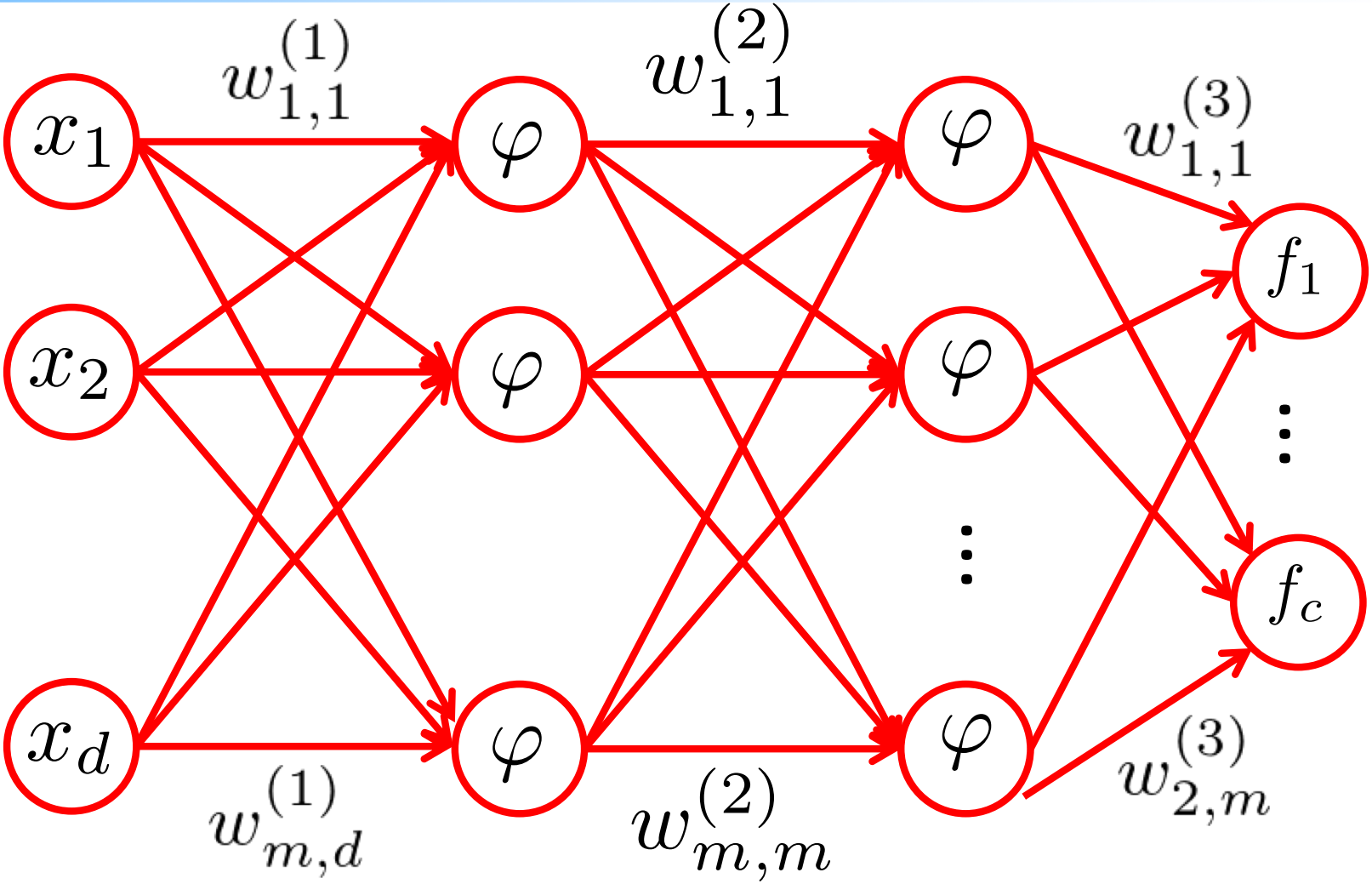
- Not unique – can force uniqueness by setting $\mathbf{w}_c = 0$ (this recovers logistic regression as special case)
- Corresponding loss function (cross-entropy loss):

$$\ell(y; \mathbf{x}, \mathbf{w}_1, \ldots, \mathbf{w}_c) = -\log P(Y = y \mid \mathbf{x}, \mathbf{w}_1, \ldots, \mathbf{w}_c)$$

Illustration



Training neural nets for multi-class



Loss: $\ell(Y = i; f_1, \ldots, f_c) = -\log \frac{\exp(f_i)}{\sum_{j=1}^{c} \exp(f_j)}$

SVM vs. Logistic regression

Method	*SVM / Perceptron*	*Logistic regression*
Advantages	Sometimes higher classification accuracy; Sparse sol‘s	Can obtain class probabilities
Disadvantages	Can‘t (easily) get class probabilities	Dense solutions

Outlook: Bayesian learning

"Optimization" based learning (MAP, MLE, …):

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} P(\mathbf{w} \mid D) \qquad P(y \mid \mathbf{x}, \hat{\mathbf{w}})$$

Ignores uncertainty in model

Optimization typically efficient

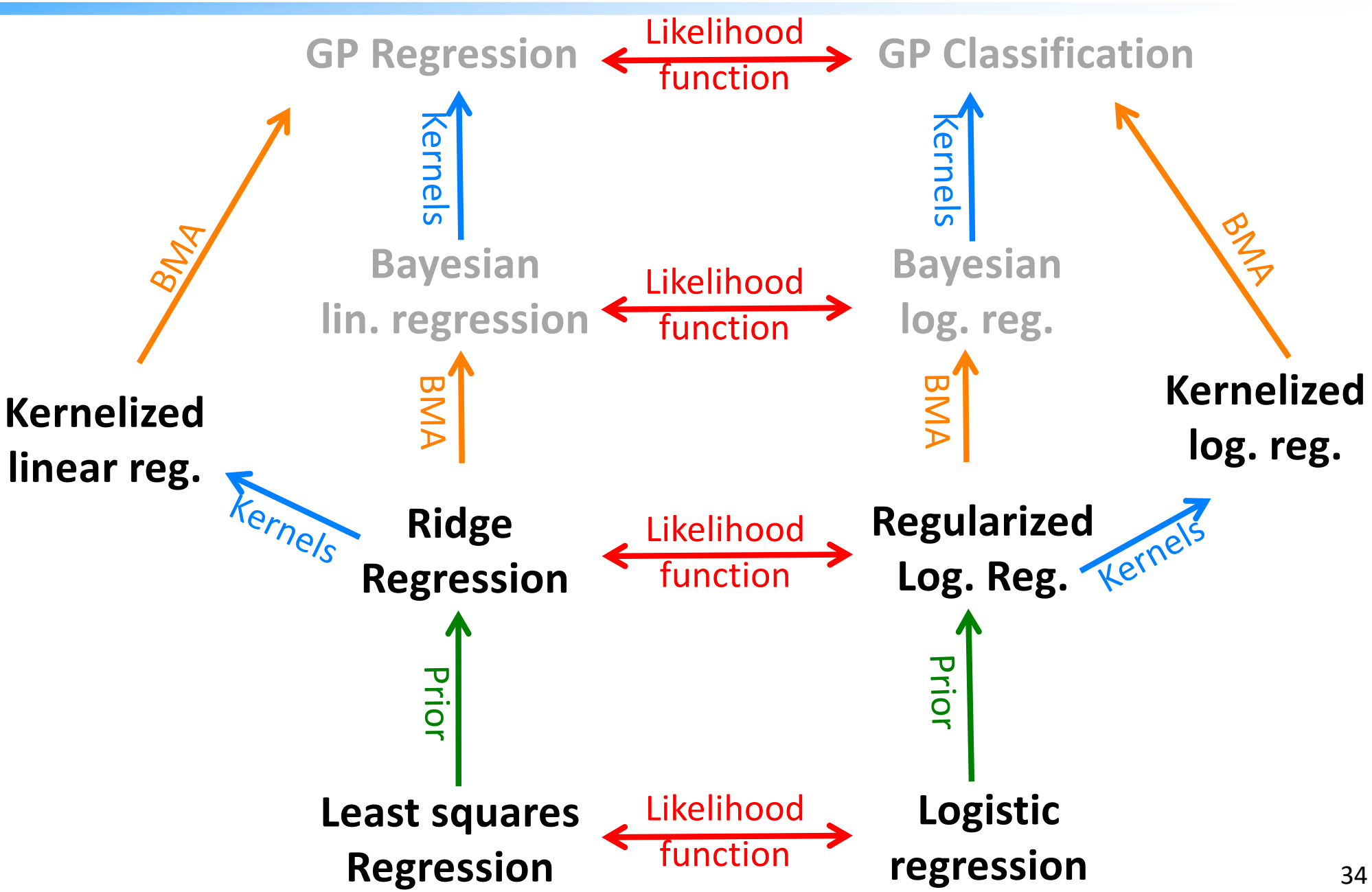
"Integration" based learning / Bayesian model averaging:

$$P(y \mid \mathbf{x}, D) = \int P(y \mid \mathbf{x}, \mathbf{w}) P(\mathbf{w} \mid D)$$

Quantifies uncertainty in model

integration typically intractable

Probabilistic modeling big picture so far



Representation/ features	Linear hypotheses; nonlinear hypotheses with nonlinear feature transforms, kernels, learn nonlinear features via neural nets		
Probabilistic / Optimization Model:	Likelihood	*	Prior
	Loss-function	+	Regularization
	Squared loss = Gaussian lik., 0/1, Perceptron, Hinge, cost sensitive, multi-class hinge, reconstruction error, logistic loss=Bernoulli lik., cross-entropy loss=Categorical lik.		L^2 norm (=Gaussian prior), L^1 norm (=Laplace prior), early stopping, dropout
Method:	Exact solution, Gradient Descent, (mini-batch) SGD, Reductions, Lloyd's heuristic, Bayesian model averaging		
Evaluation metric:	Mean squared error, Accuracy, F1 score, AUC, Confusion matrices, compression performance, log-likelihood on validation set		
Model selection:	K-fold Cross-Validation, Monte Carlo CV, Bayesian model selection		