Classification

Week goals

- What is classification
- Which loss functions are used to train linear classifiers
- How to measure performance of a classifier
- What is Logistic regression and SVM models and how they are connected to linear classifier with error rate as a loss function

Binary Classification

- $\mathbb{Y} = \{-1, +1\}$
- $\mathbf{-1}$ – negative class
- $\mathbf{+1}$ – positive class
- $a(x)$ should return one of two numbers

Linear Classifier

$$a(x) = w_0 + \sum_{j=1}^{d} w_j x_j$$

- Returns a real number

Linear Classifier

$$a(x) = \operatorname{sign}\left(w_0 + \sum_{j=1}^{d} w_j x_j \right)$$

Linear Classifier

$$a(x) = \mathrm{sign}(\langle w,\ x \rangle)$$

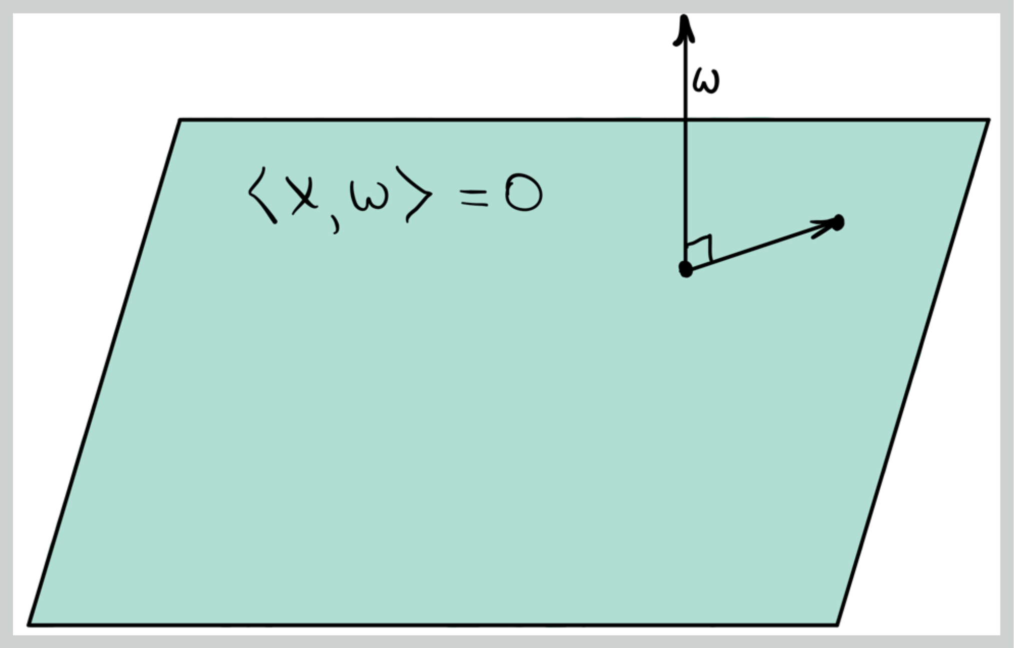
- Assuming that we have a feature, which is always 1

Linear Classifier

Hyperplane

$\langle w,\ x \rangle = 0$

- w – normal vector



Linear Classifier

Hyperplane

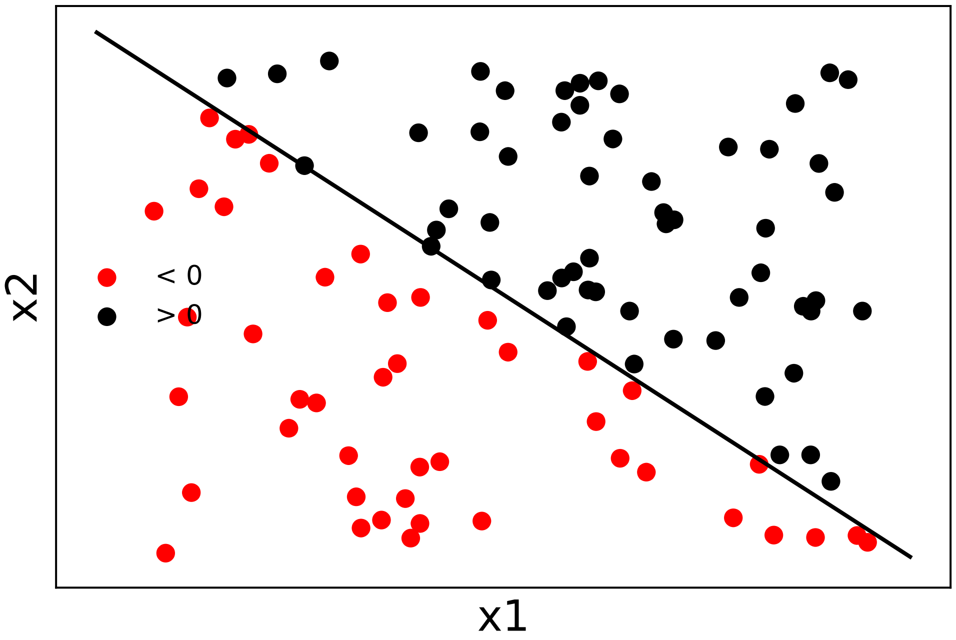
$\langle w,\ x \rangle = 0$

- w – normal vector
- If x lies on hyperplane, then $\langle w,\ x \rangle = 0$
- $\langle w,\ x \rangle < 0$ – object lies «to the left» from hyperplane
- $\langle w,\ x \rangle > 0$ – object lies «to the right» from hyperplane

Linear Classifier

Hyperplane

$$\langle w,\ x \rangle = 0$$



Margin

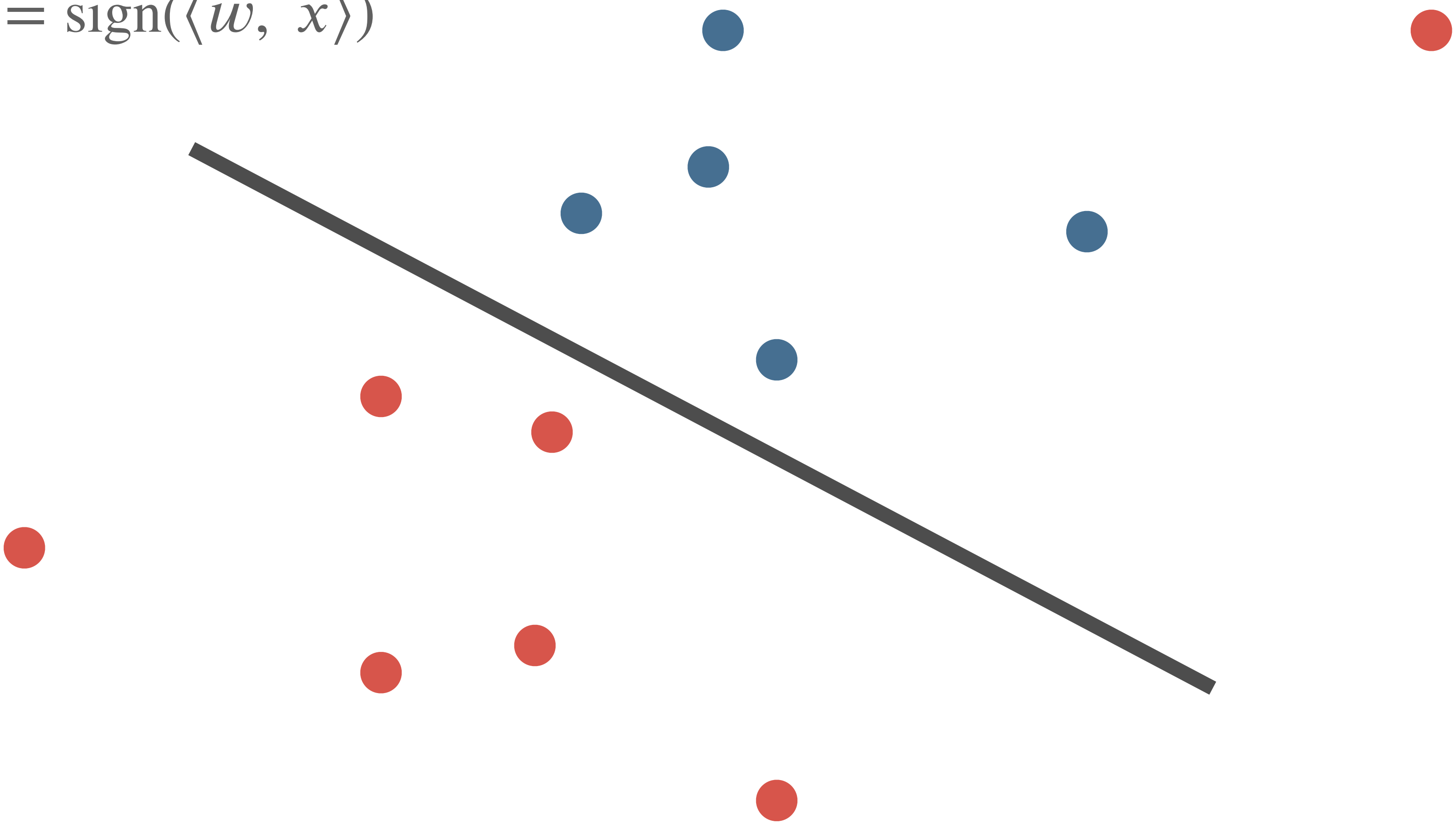
- Distance from the point to hyperplane $\langle w,\ x\rangle = 0$:

$$\frac{\left|\langle w,\ x\rangle\right|}{\|w\|}$$

- The lager $\langle w,\ x\rangle$ value is, further the object is from a hyperplane

Margin

$$a(x) = \mathrm{sign}(\langle w,\ x \rangle)$$



Margin

$$M_i = y_i \langle w, \; x_i \rangle$$

- $M_i > 0$ – classifier give the right answer
- $M_i < 0$ – classifier give the wrong answer
- Distance from zero indicates the confidence of the classifier (large absolute values mean that the classifier is more confident)

Threshold

$$a(x) = \mathrm{sign}(\langle w,\ x\rangle - t)$$

- t – threshold
- We can choose threshold using loss function which is different from the one used for training

Summary

- Linear classifier separates two classes using a hyperplane

$$a(x) = \text{sign}(\langle w,\ x \rangle - t)$$

- Sing of a scalar product shows, on which side compared to hyperplane the object lies

- Margin reflects confidence of a classifier on a given object

$$M_i = y_i \left\langle w,\ x_i \right\rangle$$

Training Linear Classifiers

Loss function in classification

Loss function – error rate

$$L(a,\ X) = \frac{1}{N}\sum_{i=1}^{N}[a(x_i) \neq y_i]$$

Sometimes accuracy is measured:

$$L(a,\ X) = \frac{1}{N}\sum_{i=1}^{N}[a(x_i) = y_i]$$

Indicator function:

$$[A] = \begin{cases} 1, & if\ A\ is\ True \\ 0, & if\ A\ is\ False \end{cases}$$

Margin

Loss function

$$L(w,\ X) = \frac{1}{N}\sum_{i=1}^{N}\left[\operatorname{sign}\left(\langle w,\ x_i\rangle\right) \neq y_i\right]$$

Alternative formulation:

$$L(w,\ X) = \frac{1}{N}\sum_{i=1}^{N}\left[\underbrace{y_i\langle w,\ x_i\rangle}_{M_i} < 0\right]$$

Margin

Loss function

$$L(w,\ X) = \frac{1}{N} \sum_{i=1}^{N} \left[\operatorname{sign}\left(\left\langle w,\ x_i \right\rangle\right) \neq y_i\right]$$

Alternative formulation:

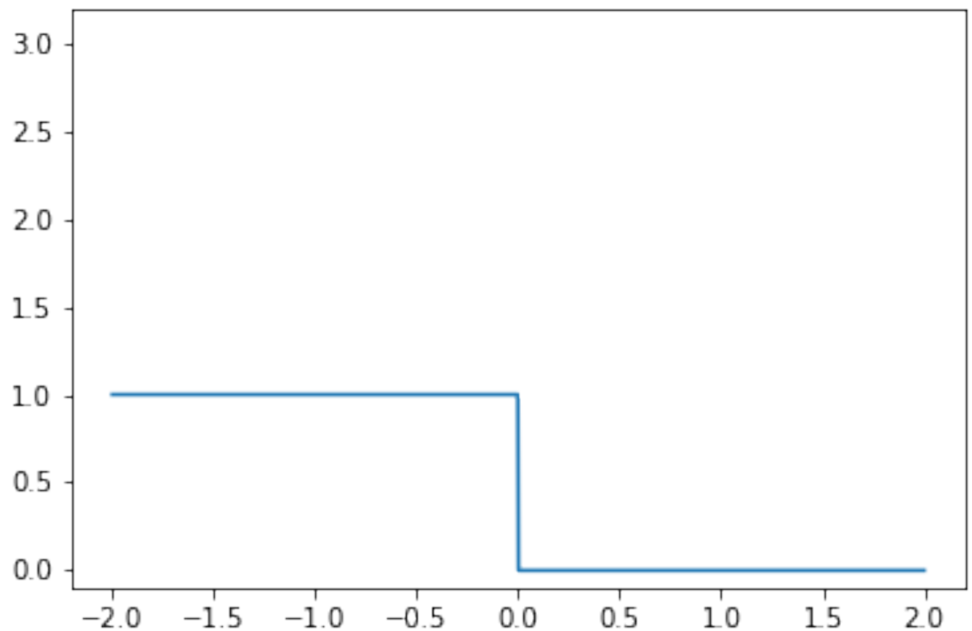
$$L(w,\ X) = \frac{1}{N} \sum_{i=1}^{N} \left[\underbrace{y_i \left\langle w,\ x_i \right\rangle}_{M_i} < 0\right]$$

Indicator – non-differentiable function

Margin

$$l(M) = [M < 0]$$

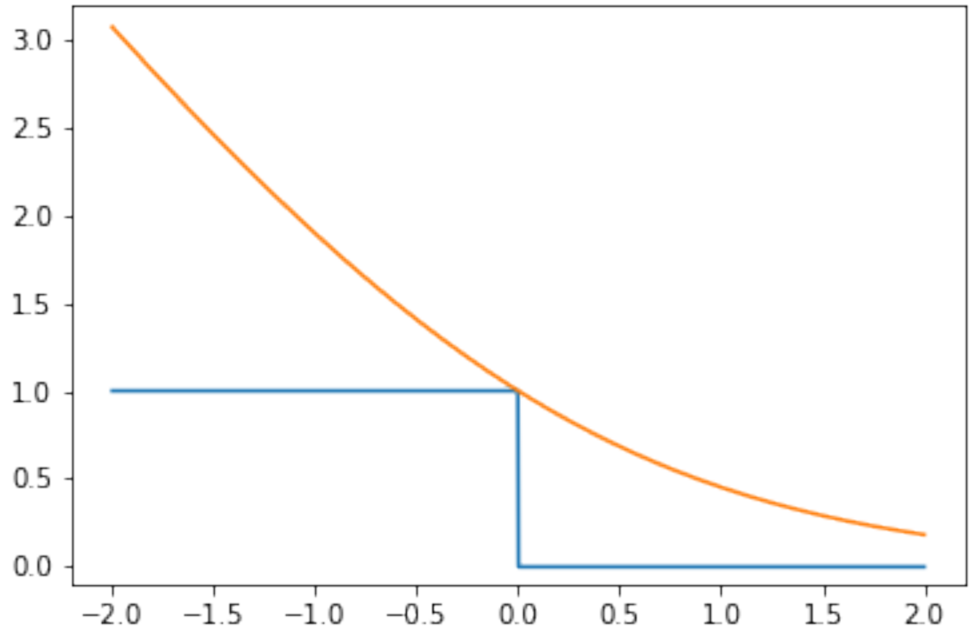
- Error rate (on 1 object) as a function of a margin



Upper bound

$$l(M) = [M < 0] \leq \tilde{l}\ (M)$$

Let us take an upper bound of the error rate



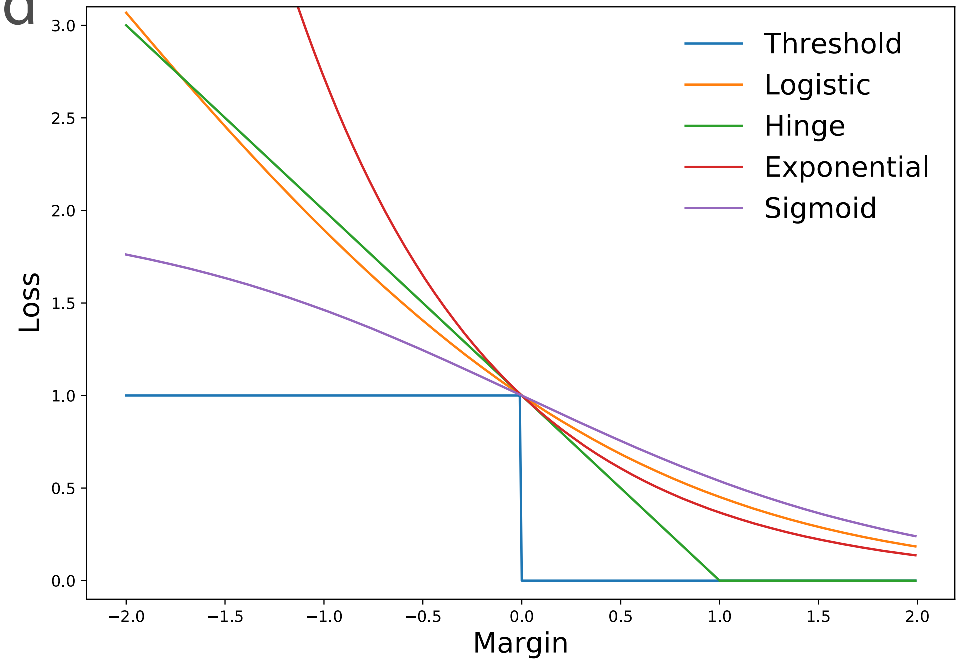
Upper Bound

$$0 \le \frac{1}{N} \sum_{i=1}^{N} [y_i \langle w, x_i \rangle < 0] \le \frac{1}{N} \sum_{i=1}^{N} \tilde{l}\left(y_i \langle w, x_i \rangle\right) \to \min_{w}$$

- We can now minimize the upper bound
- Hopefully, it will automatically reduce the error rate

Examples of Upper Bounds

- $\tilde{l}(M) = \log\left(1 + e^{-M}\right)$ – logistic
- $\tilde{l}(M) = \max(0, 1 - M)$ – hinge loss
- $\tilde{l}(M) = e^{-M}$ – exponential
- $\tilde{l}(M) = \dfrac{2}{1 + e^{M}}$ – sigmoid



Example: logistic regression

Assume, that we chose logistic loss function

$$\tilde{l}(M) = \log\left(1 + e^{-M}\right)$$

Example: logistic regression

Assume, that we chose logistic loss function

$$\tilde{l}(M) = \log\left(1 + e^{-M}\right)$$

We can now apply gradient descent to optimize the loss

$$\tilde{L}(w,\ X) = \frac{1}{N}\sum_{i=1}^{N} \log\left(1 + \exp\left(-y_i\langle w,\ x_i\rangle\right)\right) \to \min_w$$

Example: logistic regression

Assume, that we chose logistic loss function

$$\tilde{l}(M) = \log\left(1 + e^{-M}\right)$$

We can now apply gradient descent to optimize the loss

$$\tilde{L}(w,\ X) = \frac{1}{N}\sum_{i=1}^{N} \log\left(1 + \exp\left(-y_i\langle w,\ x_i\rangle\right)\right) \to \min_{w}$$

We can add regularization, just as we did with linear regression

$$\min_{w} \tilde{L}(w,\ X) + \lambda\|w\|^2$$

Summary

- It is not feasible to optimize the error rate
- We can upper bound the loss with some differentiable function and optimize this bound instead
- From this point, we can train our model as we did last week with linear regression: use gradient descent, add regularization

Quality Metrics in Classification

Loss function in classification

Loss function – error rate

$$L(a,\ X) = \frac{1}{N}\sum_{i=1}^{N}[a(x_i) \neq y_i]$$

Sometimes accuracy is measured:

$$L(a,\ X) = \frac{1}{N}\sum_{i=1}^{N}[a(x_i) = y_i]$$

Accuracy and Imbalanced Datasets

- Imbalanced Dataset – when one class has more observations than another one
- Examples:
  - Predicting that the user will click on the ad
  - Medical diagnostics

Imbalanced Datasets: examples

- Class +1: 50 observations
- Class -1: 950 observations
- Consider the model

$$a(x) = -1$$

Imbalanced Datasets: examples

- Class +1: 50 observations
- Class -1: 950 observations
- Consider the model

$$a(x) = -1$$

- Accuracy: 0.95
- What is wrong with this model?
  - The model does not add any value
  - Errors are not equivalent

Credit scoring

- Model 1: gives 100 loans
  - 80 pay-off
  - 20 defaults
- Model 2: gives 50 loans
  - 48 pay-off
  - 2 defaults
- Which one is better?

Confusion Matrix

	$y = 1$	$y = -1$
$a(x) = 1$	True Positive (TP)	False Positive (FP)
$a(x) = -1$	False Negative (FN)	True Negative (TN)

Confusion Matrix

Model $a_1(x)$

	$y = 1$	$y = -1$
$a(x) = 1$	80	20
$a(x) = -1$	20	80

Model $a_2(x)$

	$y = 1$	$y = -1$
$a(x) = 1$	48	2
$a(x) = -1$	52	98

Precision

- Can we trust a classifier, when it attributes an object to a positive class?

$$\text{precision}(a,\ X) = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Precision

Model $a_1(x)$

	y = 1	y = -1
a(x) = 1	**80**	**20**
a(x) = -1	20	80

$$\text{precision}\left(a_1, X\right) = 0.8$$

Model $a_2(x)$

	y = 1	y = -1
a(x) = 1	**48**	**2**
a(x) = -1	52	98

$$\text{precision}\left(a_2, X\right) = 0.96$$

Recall

- What proportion of a positive class the model was able to detect?

$$\text{recall}(a,\ X) = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Recall

Модель $a_1(x)$

	y = 1	y = -1
a(x) = 1	**80**	20
a(x) = -1	**20**	80

$$\text{recall}\left(a_1, X\right) = 0.8$$

Модель $a_2(x)$

	y = 1	y = -1
a(x) = 1	**48**	2
a(x) = -1	**52**	98

$$\text{recall}\left(a_2, X\right) = 0.48$$

Examples

- Credit scoring
  - No more that 5% of defaults
  - $\text{precision}(a, X) \geq 0.95$
  - Maximize recall
- Medical diagnostics
  - Find at least 90% of all the sick
  - $\text{recall}(a, X) \geq 0.9$
  - Maximize precision

Precision and Recall

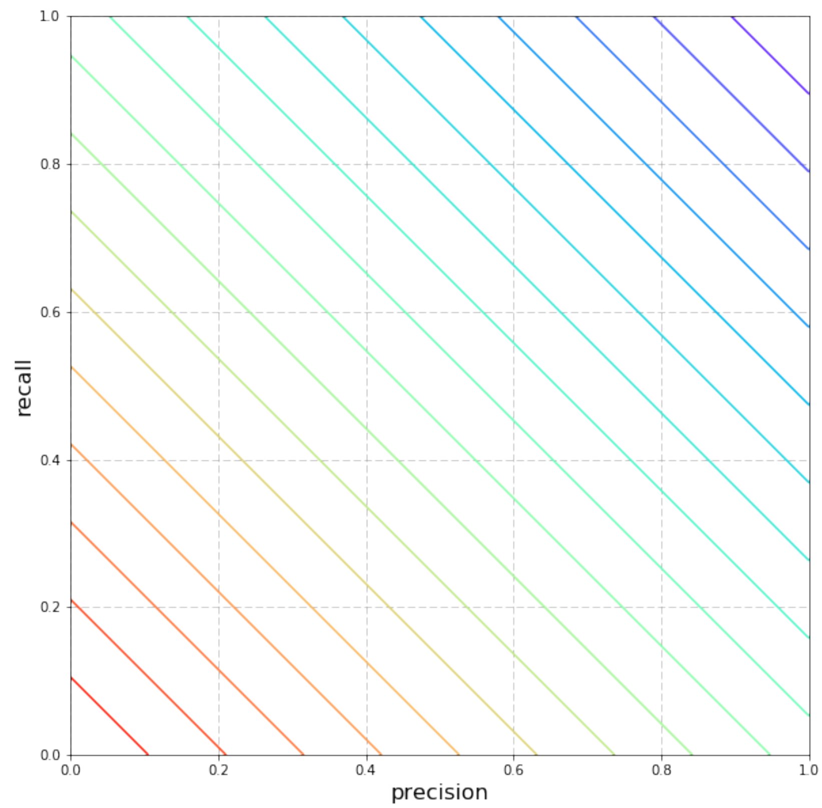
$$\text{precision}(a,\ X) = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{recall}(a,\ X) = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- What if we want to optimize both?

Average

$$A = \frac{1}{2}\left(\text{precision} + \text{recall}\right)$$

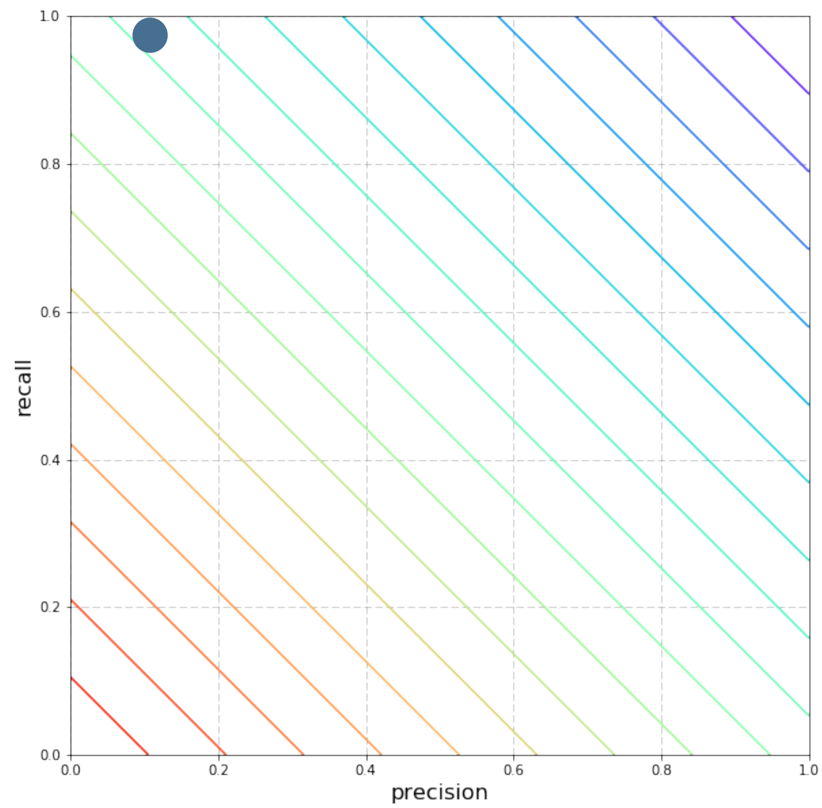


Average

$$A = \frac{1}{2}\left(\text{precision} + \text{recall}\right)$$

- $\text{precision} = 0.1$
- $\text{recall} = 1$
- $A = 0.55$

A bad algorithm

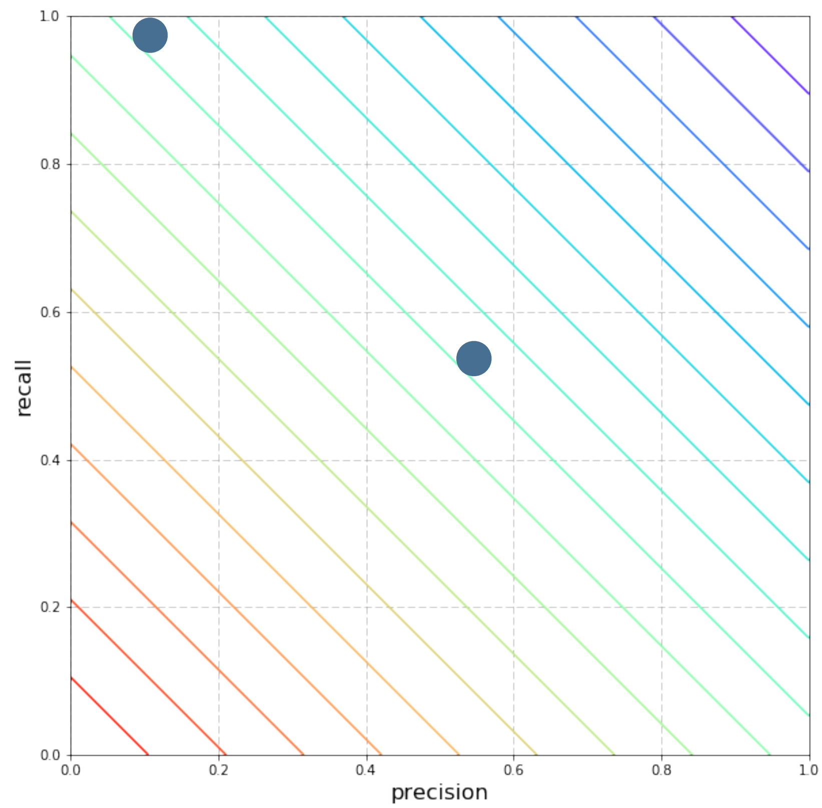


Average

$$A = \frac{1}{2}\left(\text{precision} + \text{recall}\right)$$

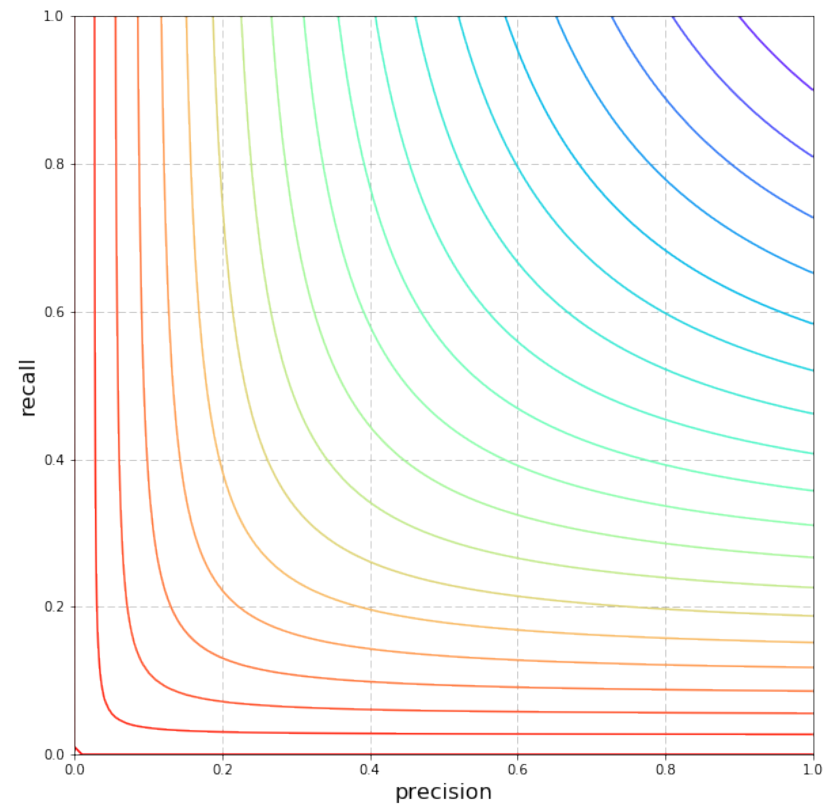
- $\text{precision} = 0.55$
- $\text{recall} = 0.55$
- $A = 0.55$

A better algorithm



F_1 score (harmonic mean)

$$F = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$



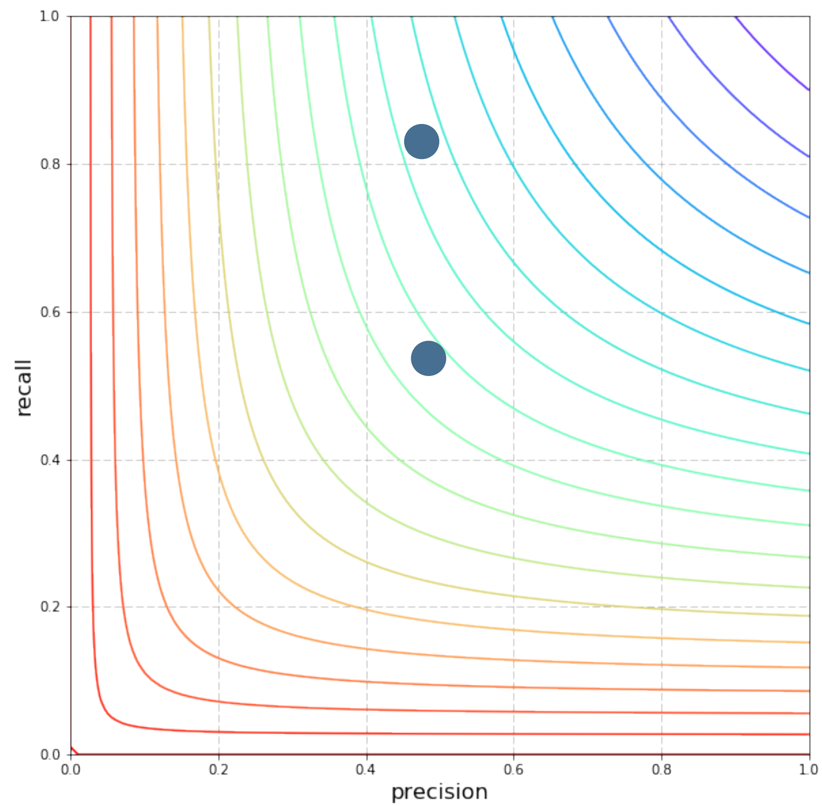
F_1 score (harmonic mean)

$$F = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

- $\text{precision} = 0.1, \text{recall} = 1$
- $F = 0.18$

- $\text{precision} = 0.55, \text{recall} = 0.55$
- $F = 0.55$

- $\text{precision} = 0.55, \text{recall} = 0.8$
- $F = 0.652$



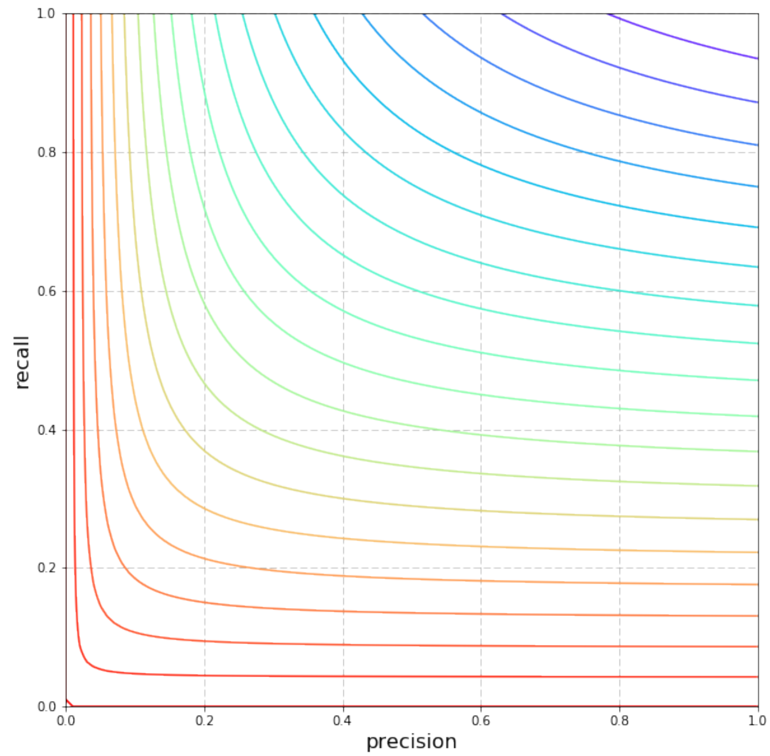
F_β score

$$F = \frac{(1 + \beta^2) * \text{precision} * \text{recall}}{\beta^2 * \text{precision} + \text{recall}}$$

F_β score

$$F = \frac{(1+\beta^2) * \text{precision} * \text{recall}}{\beta^2 * \text{precision} + \text{recall}}$$

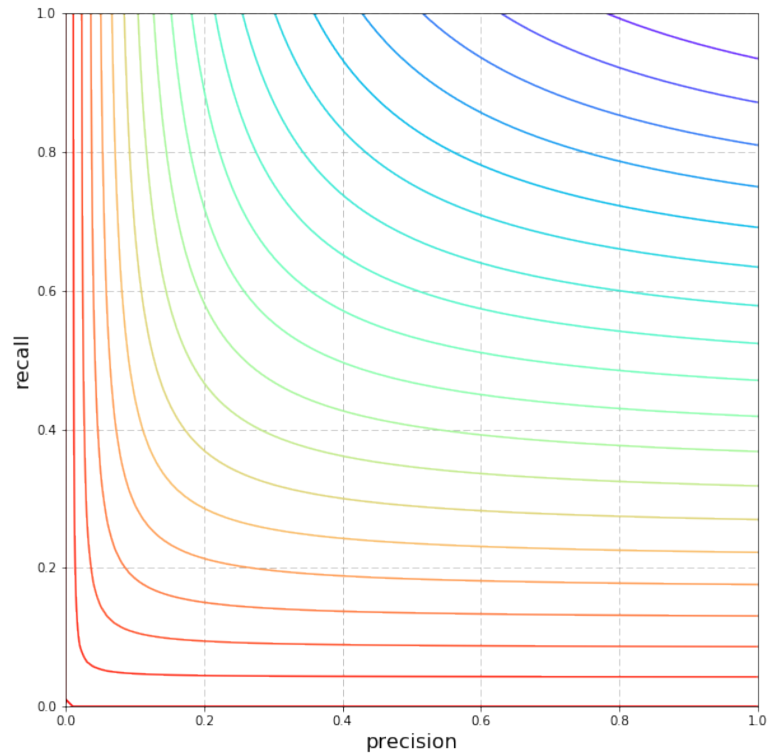
- $\beta = 0.5$ – precision is more important



F_β score

$$F = \frac{(1+\beta^2) * \text{precision} * \text{recall}}{\beta^2 * \text{precision} + \text{recall}}$$

- $\beta = 2$ – recall is more important



Summary

- Accuracy is a very convenient metrics, but sometimes it is not the best way to assess quality of a model

- To distinguish between different errors one can use precision and recall

- Moreover, we can combine them into one metric, e.g. use harmonic mean

Precision-Recall Curve

Linear Classifier and Threshold

- Classifier

$$a(x) = \operatorname{sign}\big(b(x) - t\big) = 2\big[b(x) > t\big] - 1$$

- Linear classifier:

$$a(x) = \operatorname{sign}\big(\langle w, x\rangle - t\big) = 2\big[\langle w, x\rangle > t\big] - 1$$

Linear Classifier and Threshold

- Classifier

$$a(x) = \operatorname{sign}\big(b(x) - t\big) = 2\big[b(x) > t\big] - 1$$

- Linear classifier:

$$a(x) = \operatorname{sign}\big(\langle w, x\rangle - t\big) = 2\big[\langle w, x\rangle > t\big] - 1$$

- $\langle w, x\rangle$ – assesses the possibility of the class +1
- How to choose t?
- How to evaluate $b(x)$?

Linear Classifier and Threshold

- Classifier

$$a(x) = \operatorname{sign}\big(b(x) - t\big) = 2\big[b(x) > t\big] - 1$$

- Linear classifier:

$$a(x) = \operatorname{sign}\big(\langle w, x\rangle - t\big) = 2\big[\langle w, x\rangle > t\big] - 1$$

- $\langle w, x\rangle$ – assesses the possibility of the class +1
- How to choose t? **Based on precision and recall**
- How to evaluate $b(x)$?

Threshold examples

$$a(x) = \text{sign}\big(b(x) - t\big)$$

	-1	-1	+1	+1	-1	-1	+1	+1	-1	+1
	0.01	0.09	0.12	0.15	0.29	0.4	0.48	0.6	0.83	0.9

Threshold examples

$$a(x) = \text{sign}\big(b(x) - t\big)$$

	-1	-1	+1	+1	-1	-1	+1	+1	-1	+1
	0.01	0.09	0.12	0.15	0.29	0.4	0.48	0.6	0.83	0.9
	-1	-1	-1	-1	-1	-1	+1	+1	+1	+1

$t = 0.45$

$$\text{precision} = \frac{3}{3+1} = 0.75$$

$$\text{recall} = \frac{3}{3+2} = 0.6$$

Threshold examples

$$a(x) = \text{sign}\big(b(x) - t\big)$$

	-1	-1	+1	+1	-1	-1	+1	+1	-1	+1
	0.01	0.09	0.12	0.15	0.29	0.4	0.48	0.6	0.83	0.9
	-1	-1	+1	+1	+1	+1	+1	+1	+1	+1

$t = 0.1$

$$\text{precision} = \frac{5}{5+3} = 0.625$$

$$\text{recall} = \frac{5}{5+0} = 1$$

Linear Classifier and Threshold

- Classifier

$$a(x) = \mathrm{sign}\big(b(x) - t\big) = 2\big[b(x) > t\big] - 1$$

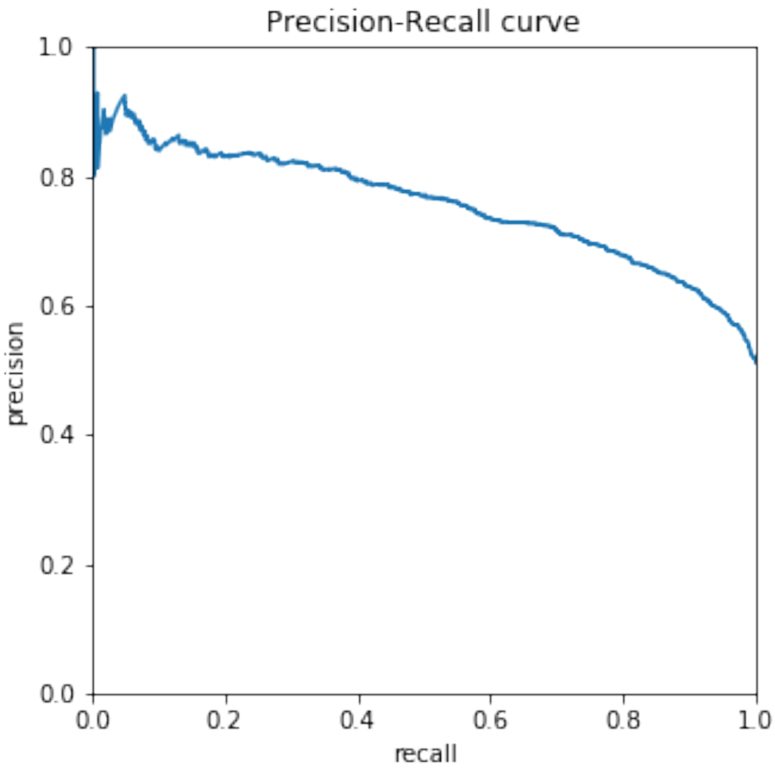
- Linear classifier:

$$a(x) = \mathrm{sign}\big(\langle w, x\rangle - t\big) = 2\big[\langle w, x\rangle > t\big] - 1$$

- $\langle w, x\rangle$ – assesses the possibility of the class +1
- How to choose t? Based on precision and recall
- How to evaluate $b(x)$?

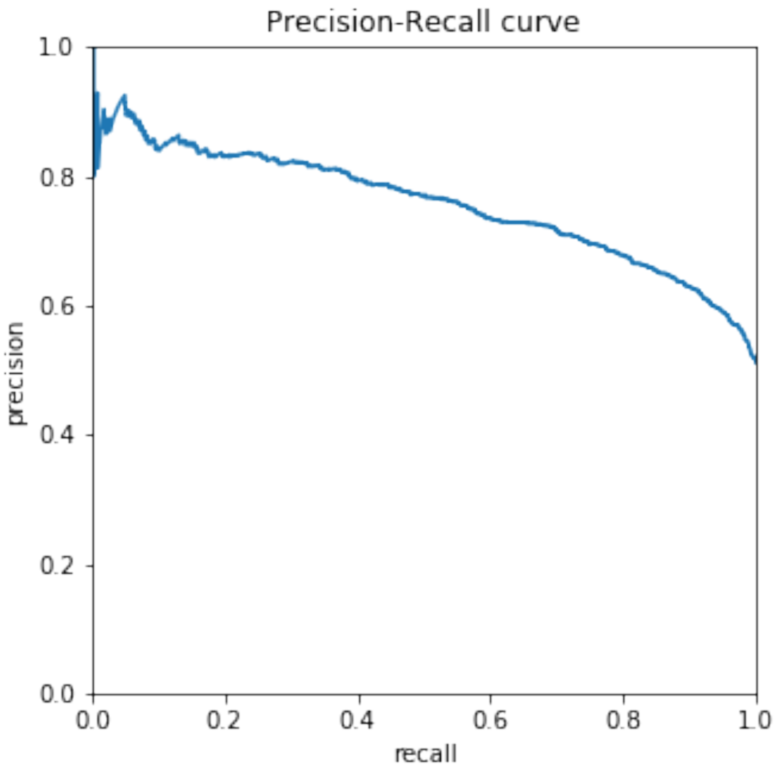
PR-curve

- X-axis – recall
- Y-axis – precision
- Each point – values of precision and recall for different thresholds



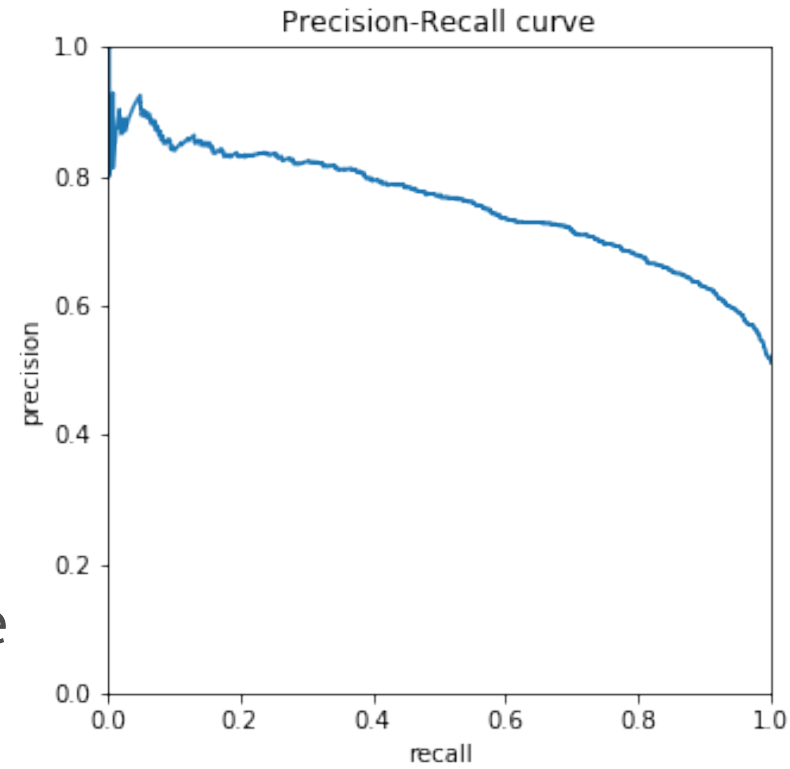
PR-curve

- Left point: (0, 0)
  - The largest threshold
  - No points in the positive class



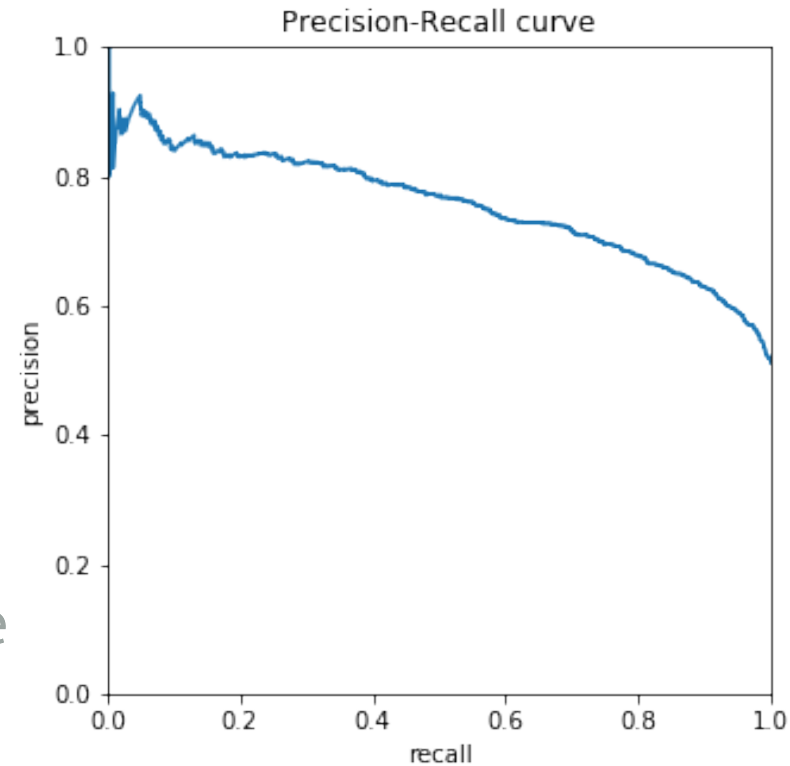
PR-curve

- Left point: (0, 0)
  - The largest threshold
  - No points in the positive class
- Right point: $(1, r)$
  - r – proportion of objects in positive class
  - The lowest threshold
  - All the points are predicted to be positive

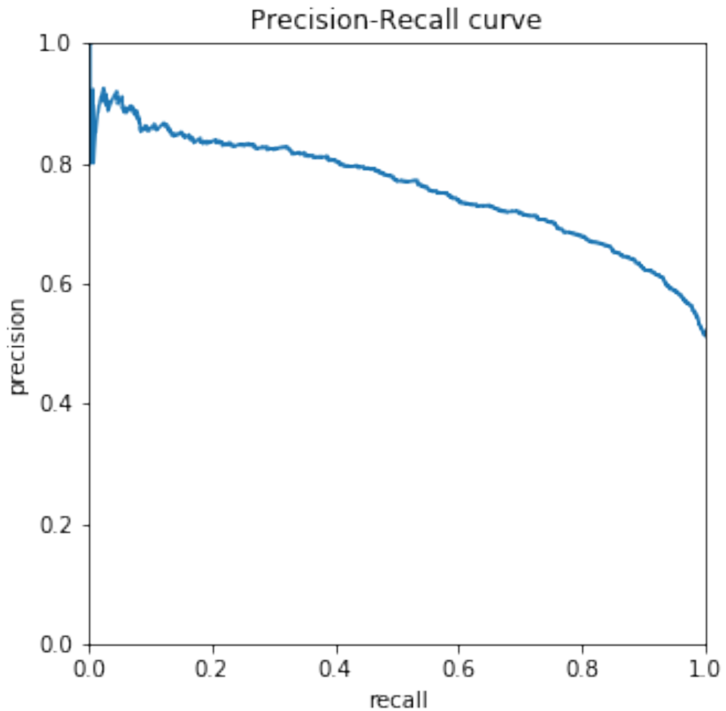


PR-curve

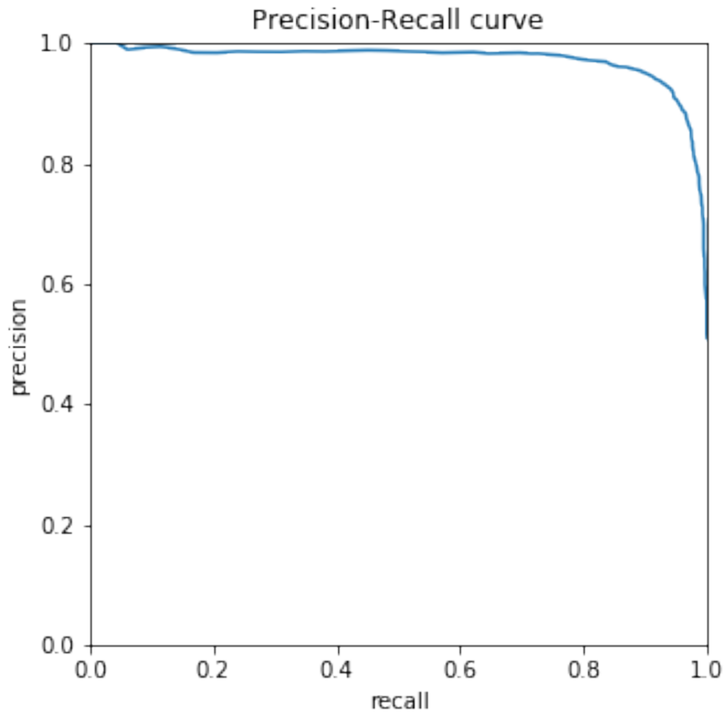
- Left point: (0, 0)
  - The largest threshold
  - No points in the positive class
- Right point: $(1, r)$
  - r – proportion of objects in positive class
  - The lowest threshold
  - All the points are predicted to be positive
- Ideal classifier – goes through the point (1, 1)
- AUC-PRC – area under PR-curve



AUC-PRC



Model 1:
AUC-PRC = 0.78



Model 2:
AUC-PRC = 0.97

Summary

- It is useful to evaluate how well the algorithm ranges the objects before choosing the threshold

- Area under PR-curve is one way to do that

Area Under ROC-curve

Area Under PR-curve

$$\text{precision} = \frac{TP}{TP + FP}; \qquad \text{recall} = \frac{TP}{TP + FN}$$

- Precision changes, depending on a class balance
- AUC-PRC of an ideal algorithm changes, depending of the class balance
- Easier to interpret in case of imbalanced dataset
- Better if we are interested in precision and recall

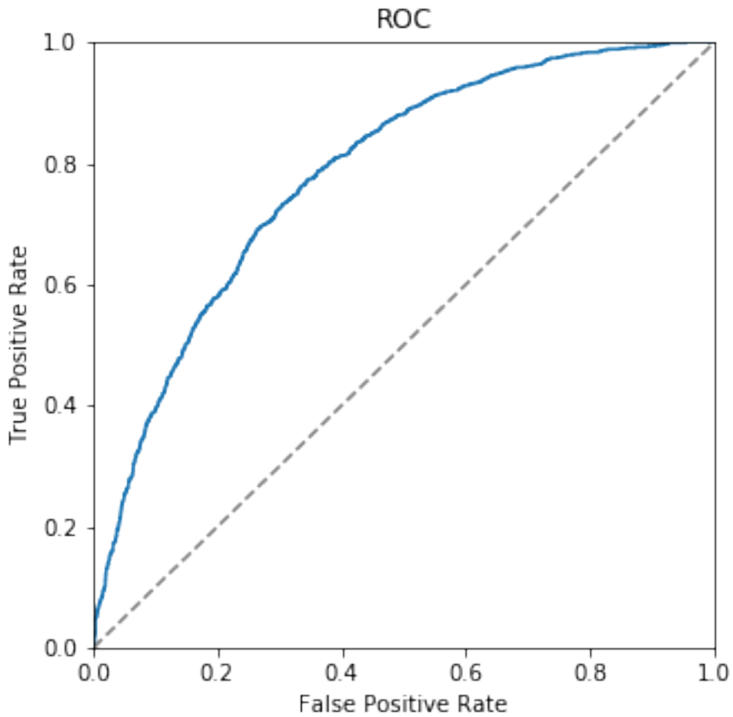
ROC-curve

- Receiver Operating Characteristic
- X-axis – False Positive Rate

$$FPR = \frac{FP}{FP + TN}$$

- Y-axis – True Positive Rate (Recall)

$$TPR = \frac{TP}{TP + FN}$$



ROC-curve

- Receiver Operating Characteristic
- X-axis – False Positive Rate

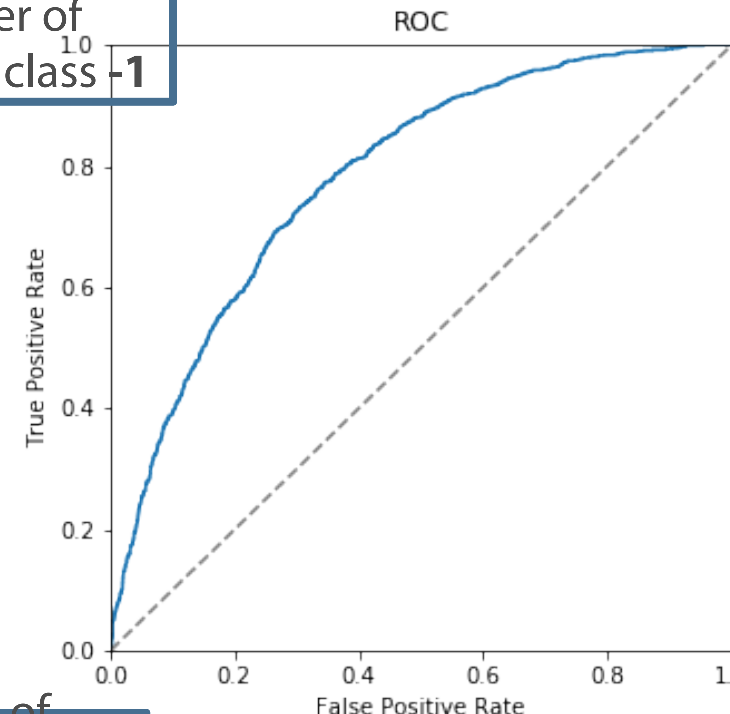
$$FPR = \frac{FP}{FP + TN}$$

Number of objects in class **-1**

- Y-axis – True Positive Rate (Recall)

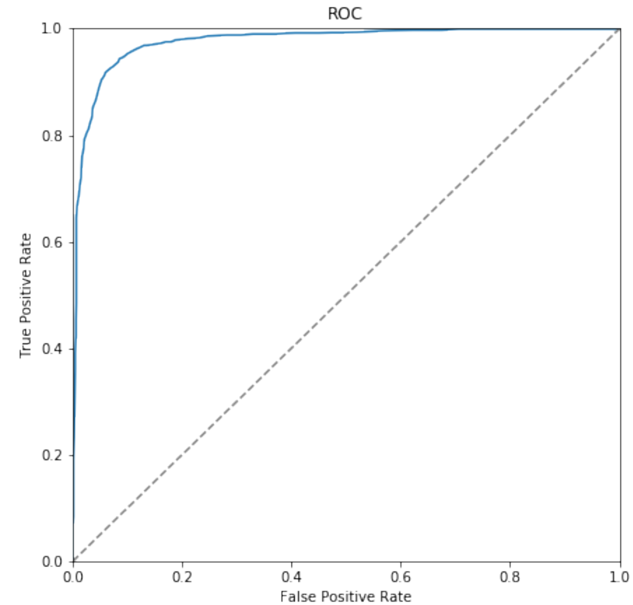
$$TPR = \frac{TP}{TP + FN}$$

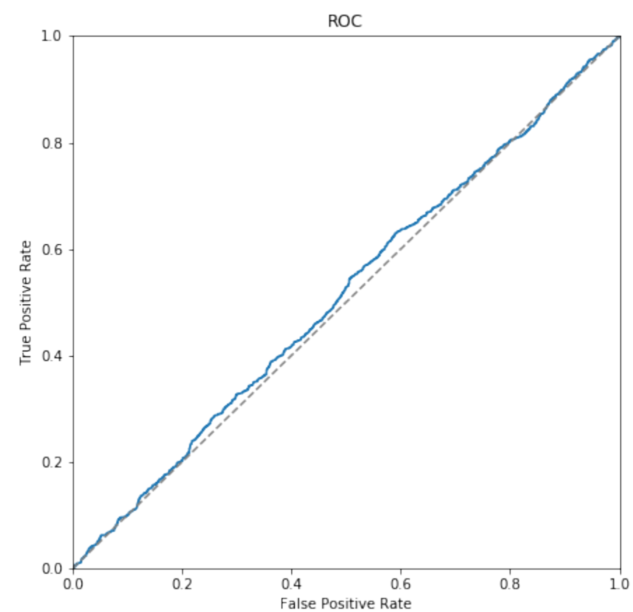
Number of objects in class +1



ROC-curve

- Left point: (0, 0)
- Right point: (1, 1)
- Idea classifier goes through (0, 1)
- AUC-ROC – area under ROC-curve





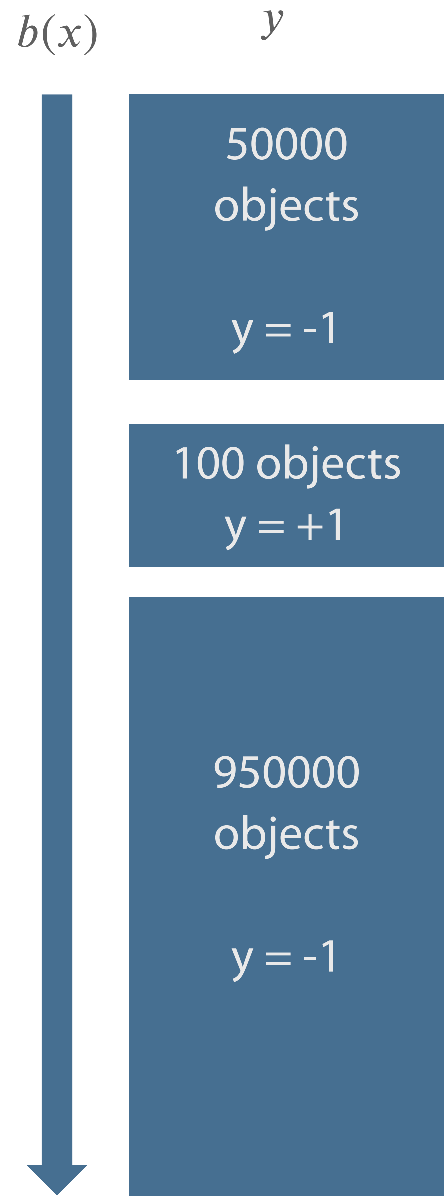
Area Under ROC-curve

$$FPR = \frac{FP}{FP + TN}; \qquad TPR = \frac{TP}{TP + FN}$$

- FPR and TPR are normalized to the class size
- AUC-ROC does not change if classes are imbalanced
- AUC-ROC of a ideal classifier is 1

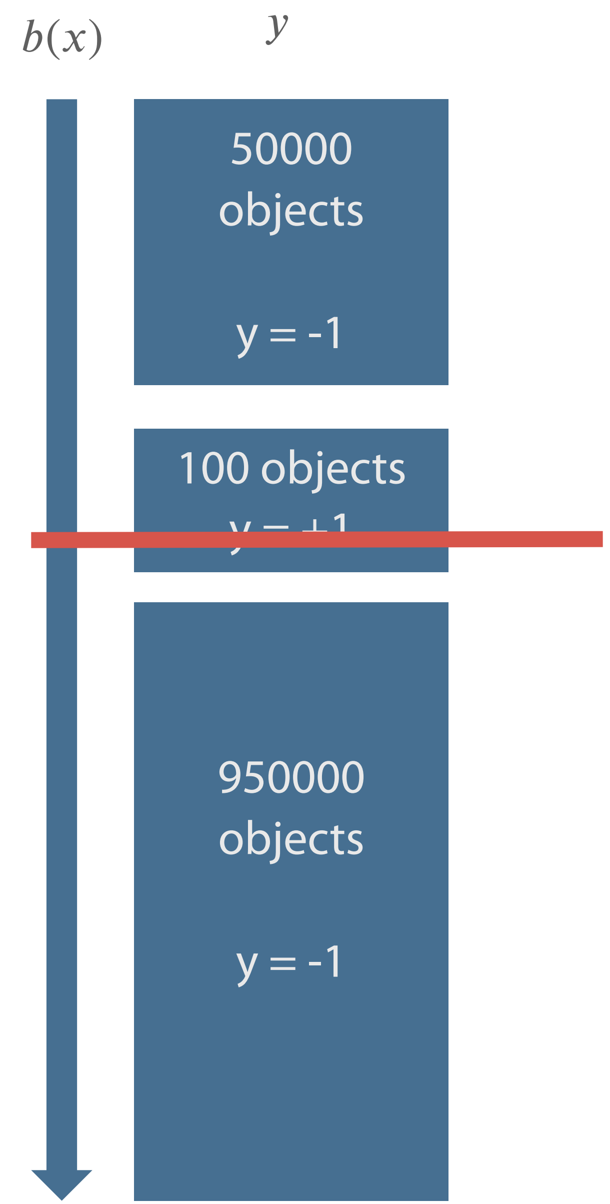
Example: AUC-ROC and AUC-PR

- AUC-ROC = 0.95
- AUC-PRC = 0.001



Example: AUC-ROC and AUC-PR

- Fix a threshold
- $a(x) = 1$ for 50095 objects
- FP = 50000, TP = 95
- TPR = 0.95, FPR = 0.05
- precision = 0.0019, recall = 0.95



Summary

- Area under ROC-curve is one of the most popular metrics used to estimate the ranging quality

- One have to be careful with AUC-ROC when classes are imbalanced

Logistic Regression

Logistic Regression

Binary classification task: $\mathbb{Y} = \{-1, +1\}$

Linear classifier:

$$a(x) = sign\big(b(x) - t\big) = sign(\langle w, x \rangle - t)$$

Logistic Regression

Binary classification task: $\mathbb{Y} = \{-1, +1\}$

Linear classifier:

$$a(x) = sign\big(b(x) - t\big) = sign(\langle w, x\rangle - t)$$

Error rate loss function:

$$\min_w \frac{1}{N} \sum_{i=1}^{N} \big[y_i \big\langle w, x_i \big\rangle < 0\big] = \min_w \frac{1}{N} \sum_{i=1}^{N} \big[M_i < 0\big]$$

Logistic Regression

Binary classification task: $\mathbb{Y} = \{-1, +1\}$

Linear classifier:

$$a(x) = sign\big(b(x) - t\big) = sign(\langle w, x \rangle - t)$$

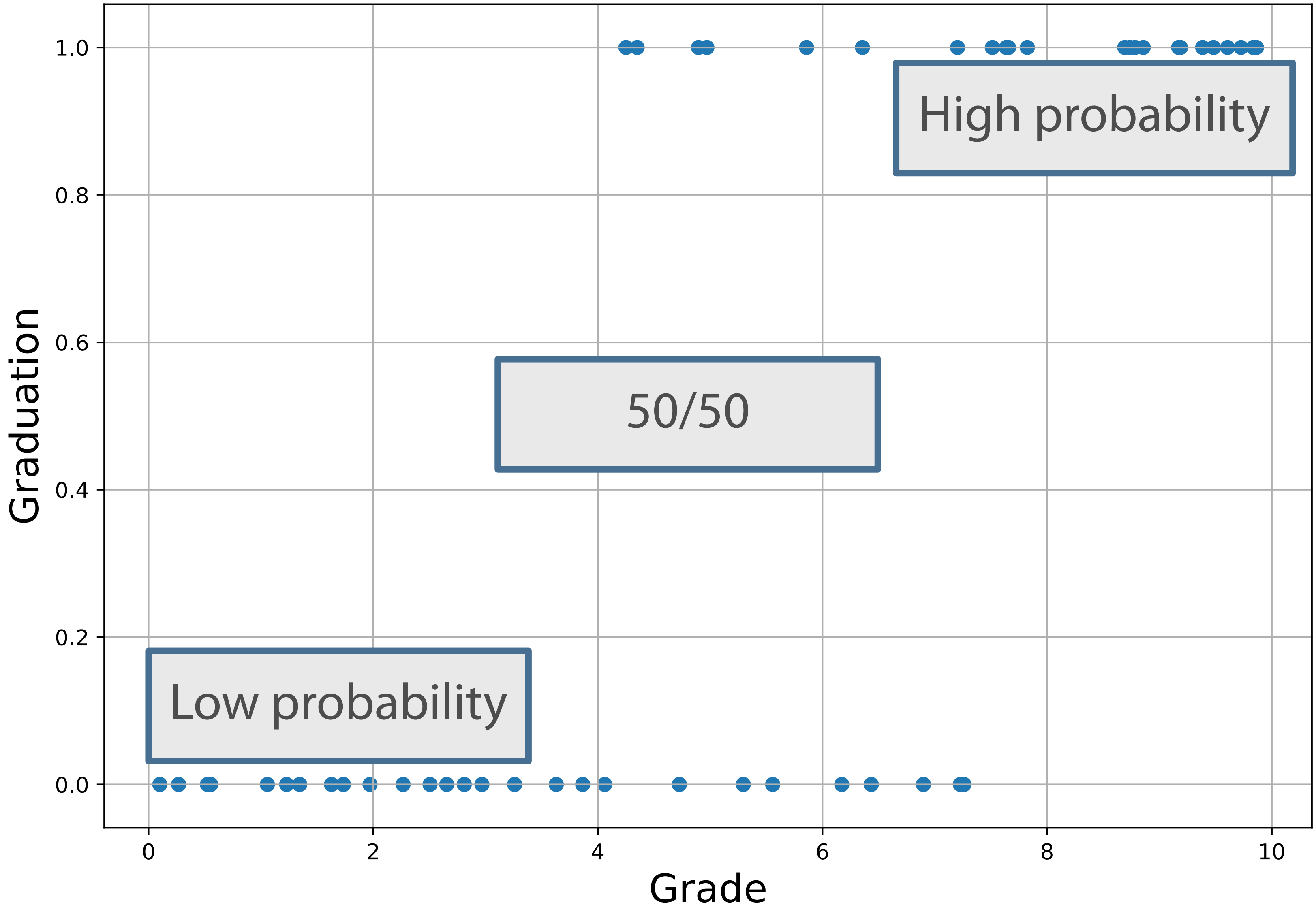
Error rate loss function:

$$\min_{w} \frac{1}{N} \sum_{i=1}^{N} \big[y_i \big\langle w, x_i \big\rangle < 0\big] = \min_{w} \frac{1}{N} \sum_{i=1}^{N} \big[M_i < 0\big]$$

We can optimize differentiable upper bound, e.g. logistic loss

$$\min_{w} \frac{1}{N} \sum_{i=1}^{N} \log\big(1 + \exp\big(-M_i\big)\big)$$

Predicting Probabilities

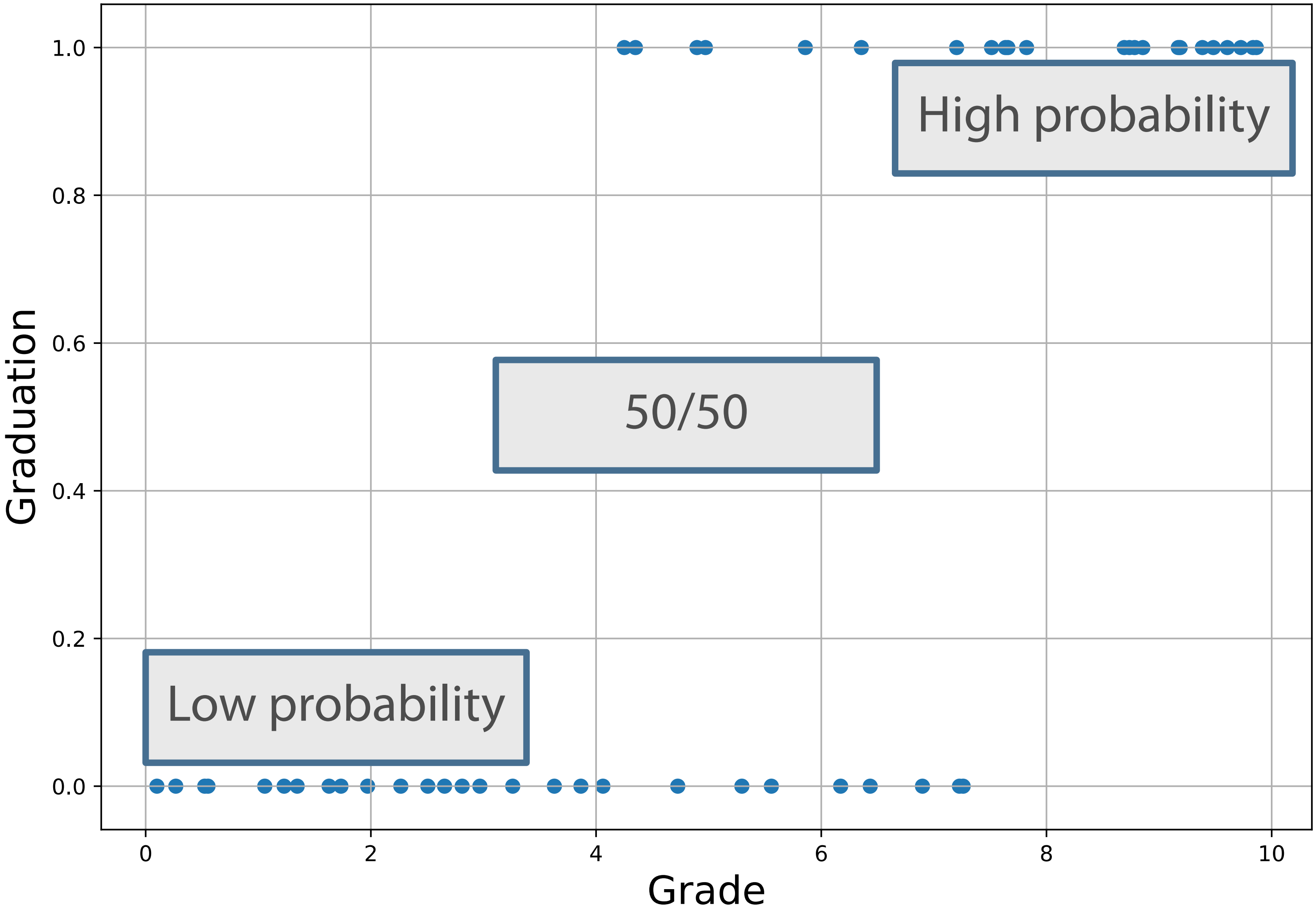


When Do We Need Probabilities?

- Credit Scoring
  - Give loans to client with probability of default less that 10%

- Internet Adds
  - $b(x)$ – probability that the person clicks
  - $c(x)$ – revenue from the ad
  - $c(x)b(x)$ – expected revenue, that we want to maximize

Predicting Probabilities



Logistic Regression

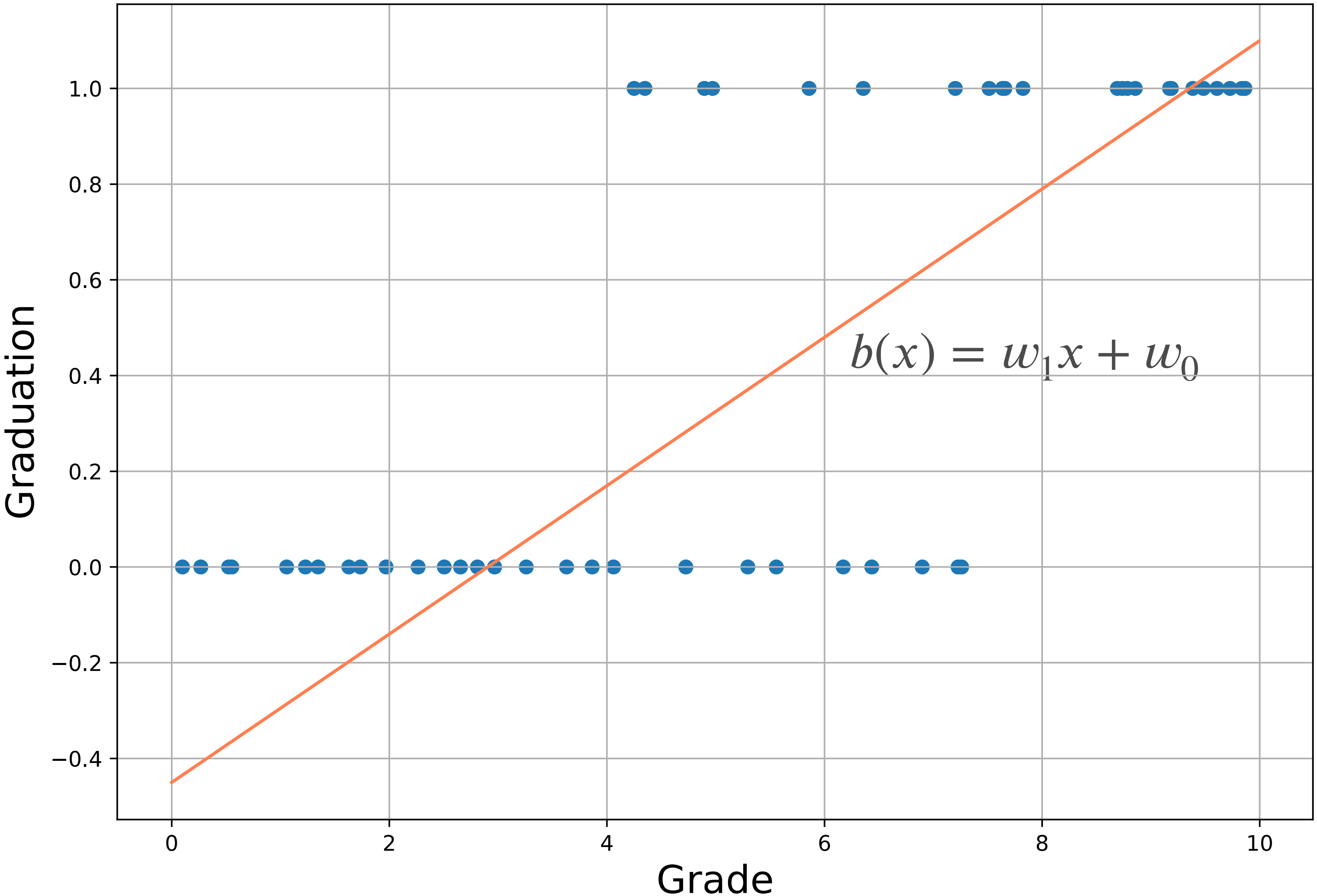
Binary classification task: $\mathbb{Y} = \{-1,\ +1\}$

Linear classifier:

$$a(x) = sign\big(b(x)\big) = sign(\langle w,\ x\rangle)$$

Can we use $\mathrm{b}(\mathbf{x}) = \langle w,\ x\rangle$ as a probability estimate?

Predicting Probabilities



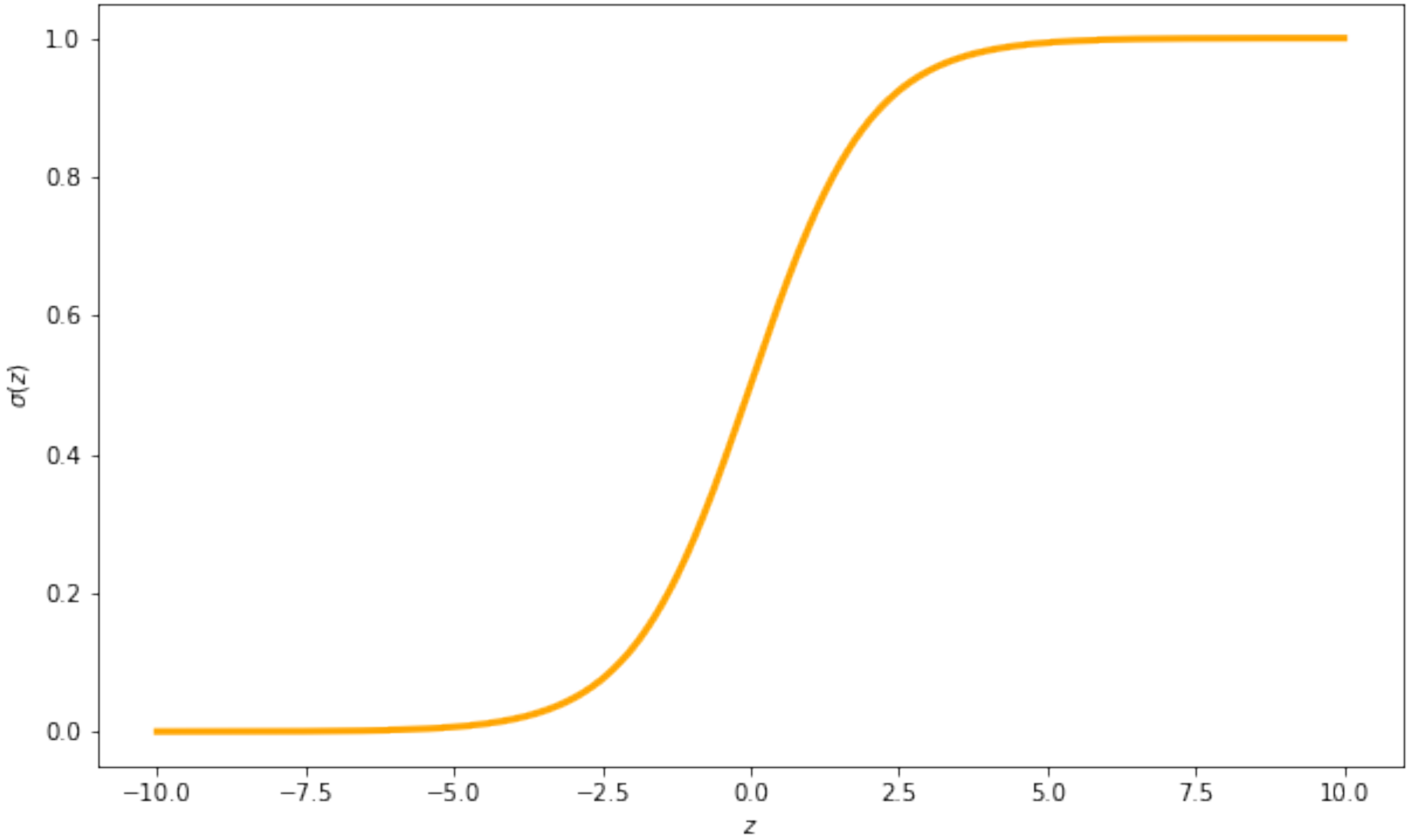
Linear Classifier

- Let us convert outputs of the model into [0, 1]

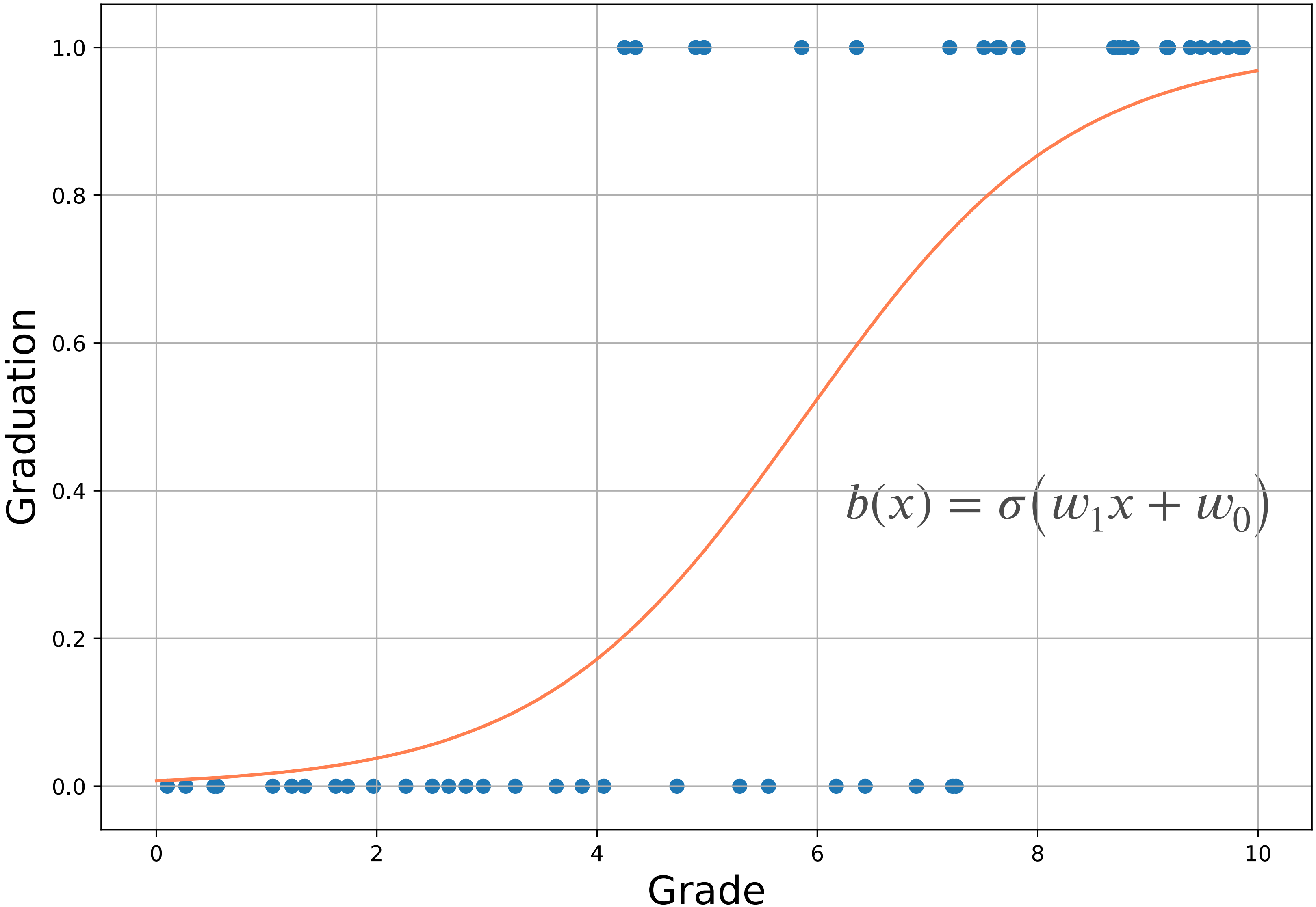
- E.g. we can use Sigmoid function:

$$\sigma\big(\langle w,\ x\rangle\big) = \frac{1}{1 + \exp\big(-\langle w,\ x\rangle\big)}$$

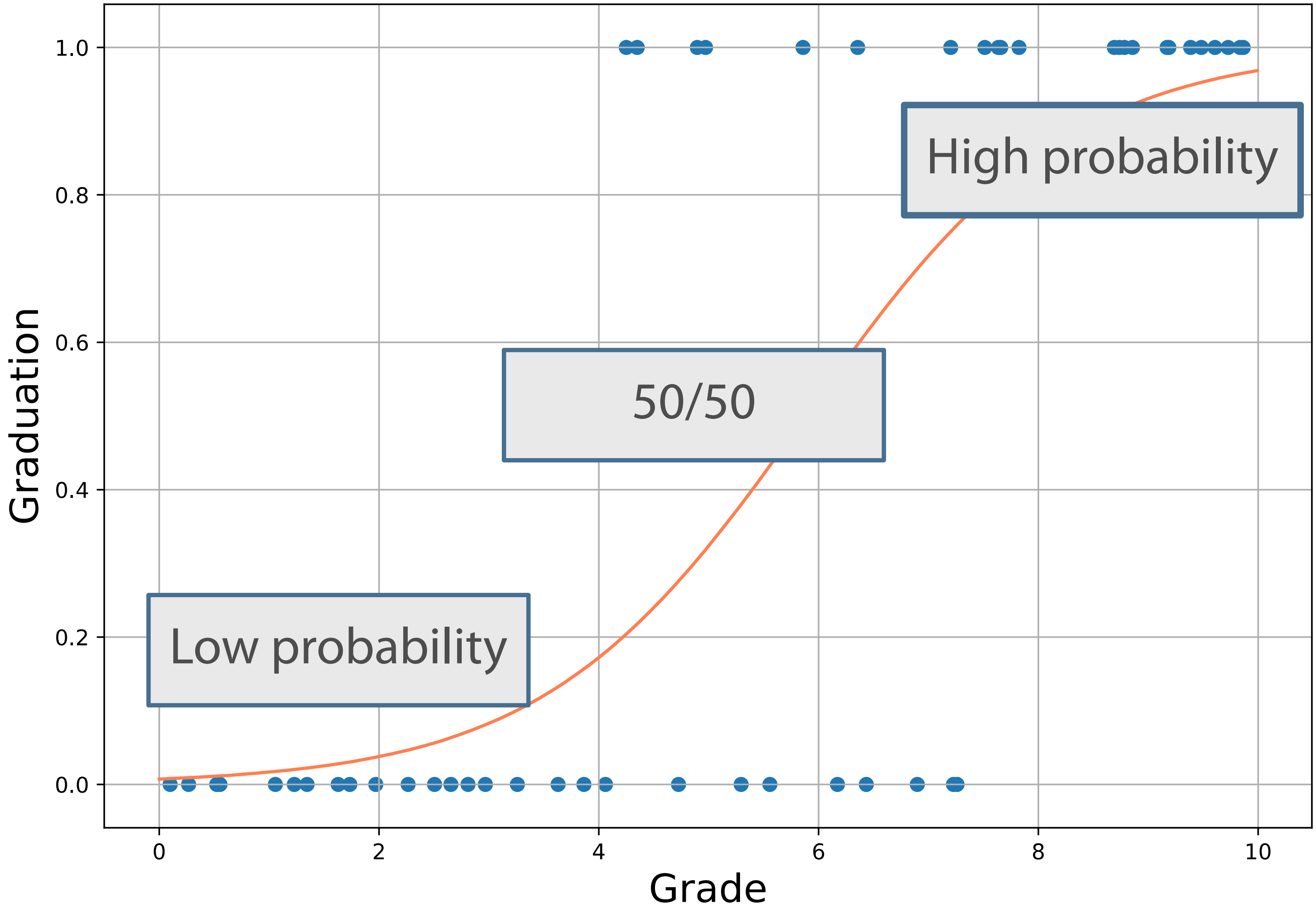
Sigmoid



Predicting Probabilities



Predicting Probabilities



Logistic Regression

Binary classification task: $\mathbb{Y} = \{-1, +1\}$

Predicted probabilities:

$$P\left(y_i = 1\right) = b(x_i)$$

Logistic Regression

Binary classification task: $\mathbb{Y} = \{-1,\ +1\}$

Predicted probabilities:

$$P\left(y_i = 1\right) = b(x_i)$$

Use sigmoid function to map outputs to the range from 0 to 1:

$$b(x) = \ \sigma\left(\langle w,\ x\rangle\right) = \frac{1}{1 + \exp\left(-\langle w,\ x\rangle\right)}$$

Logistic Regression

Binary classification task: $\mathbb{Y} = \{-1, +1\}$

Predicted probabilities:

$$P\left(y_i = 1\right) = b(x_i)$$

Use sigmoid function to map outputs to the range from 0 to 1:

$$b(x) = \sigma\left(\langle w, x\rangle\right) = \frac{1}{1 + \exp\left(-\langle w, x\rangle\right)}$$

We can now use maximum likelihood to train this model

Summary

- In some tasks it is important to predict class probabilities

- We can apply sigmoid function to the output of the model to get numbers between 0 and 1

- Finally, we want to train our model in such a way, that they would be interpreted as probabilities

Logistic Regression

Logistic Regression

Binary classification task: $\mathbb{Y} = \{-1, +1\}$

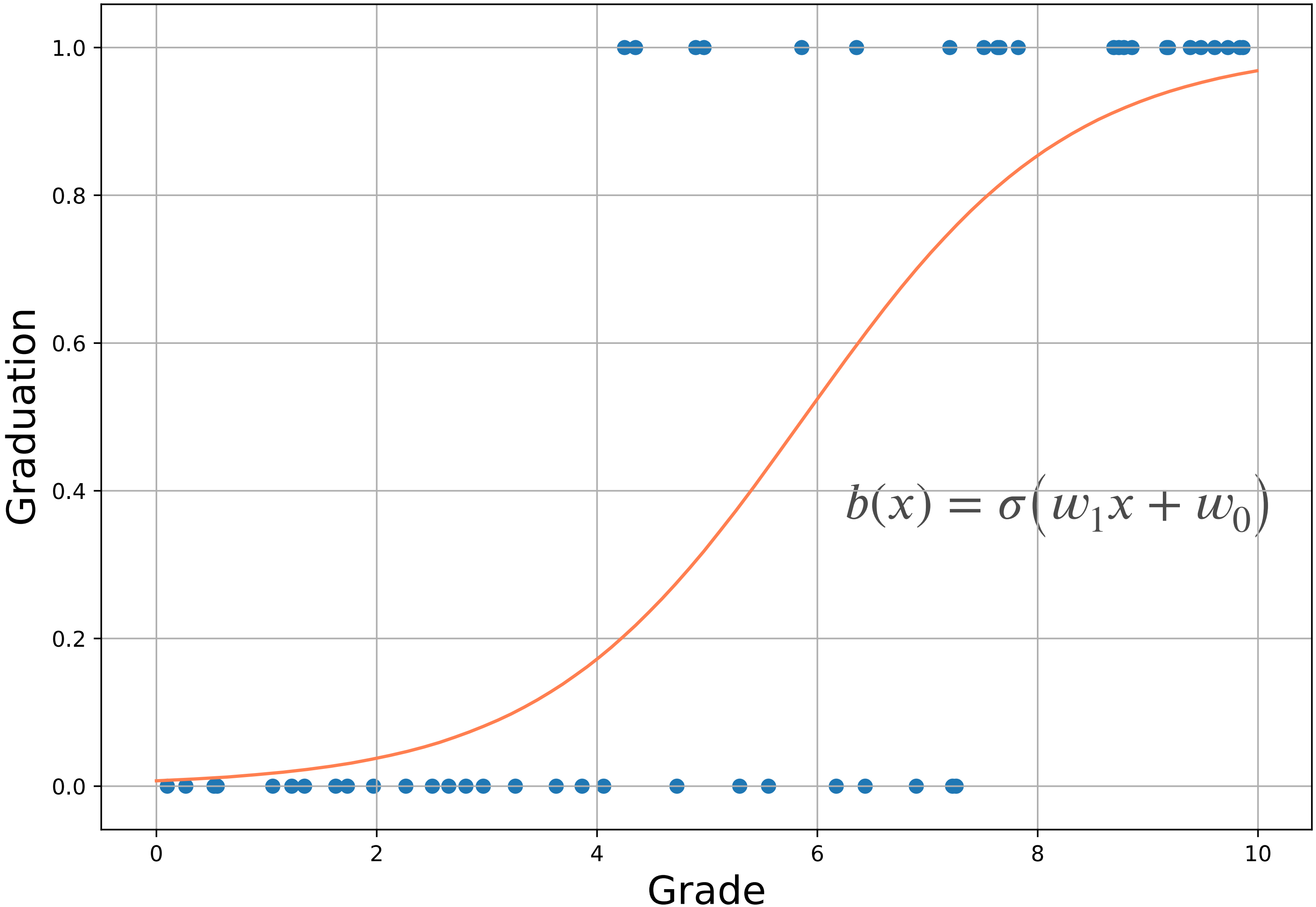
Predicted probabilities:

$$P\left(y_i = 1\right) = b(x_i)$$

Use sigmoid function to map outputs to the range from 0 to 1:

$$b(x) = \sigma\left(\langle w, x \rangle\right) = \frac{1}{1 + \exp\left(-\langle w, x \rangle\right)}$$

Predict Probabilities



Logistic Regression

Binary classification task: $\mathbb{Y} = \{-1, +1\}$

Predicted probabilities:

$$P\left(y_i = 1\right) = b(x_i)$$

Use sigmoid function to map outputs to the range from 0 to 1:

$$b(x) = \sigma\left(\langle w, x\rangle\right) = \frac{1}{1 + \exp\left(-\langle w, x\rangle\right)}$$

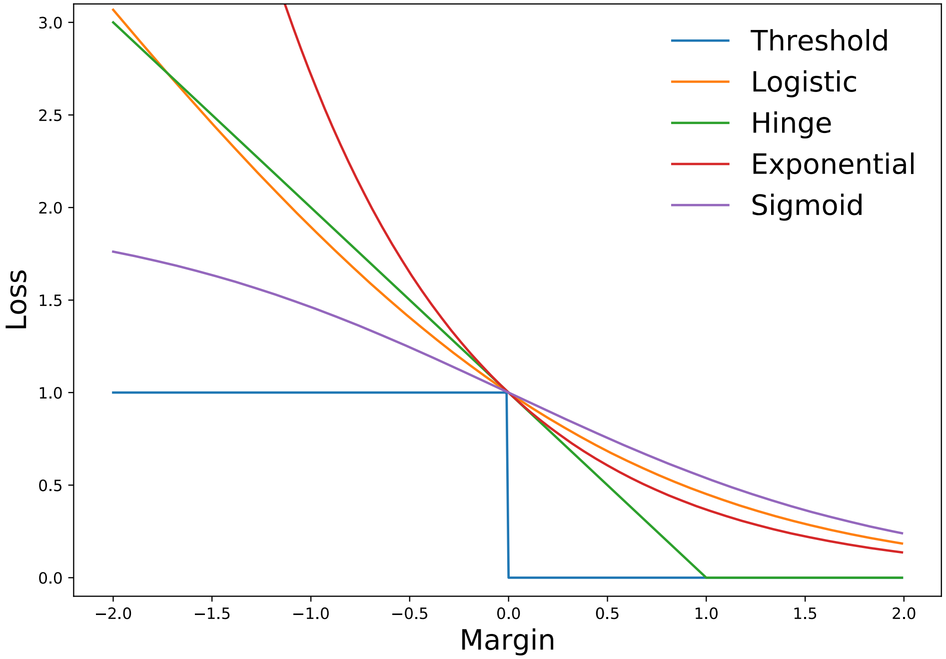
We can train with Log-Loss:

$$\min_w \sum_{i=1}^{N} \log(1 + \exp(-y_i\langle w, x_i\rangle))$$

Logistic Regression

Upper bound on the error rate

$$\tilde{l}(M) = \log\left(1 + e^{-M}\right)$$



Logistic Regression Loss:

$$\min_{w} \sum_{i=1}^{N} \log(1 + \exp(-y_i\langle w, x_i\rangle))$$

Predicting probabilities

- Does Logistic Regression give us correct probabilities?

We will say that the model $b(x)$ predicts probabilities correctly, if among objects with $b(x) = p$ proportion of positive is p.

Predicting probabilities

- Consider objects $x_1, \ldots, x_n$, where the $b(x)$ outputs the same probability around p:

$$\sum_{i=1}^{n} l\left(y_i,\ b\left(x_i\right)\right) = \sum_{i=1}^{n} l\left(y_i,\ p\right)$$

Predicting probabilities

- Consider objects $x_1, \ldots, x_n$, where the $b(x)$ outputs the same probability around p:

$$\sum_{i=1}^{n} l\left(y_i, b\left(x_i\right)\right) = \sum_{i=1}^{n} l\left(y_i, p\right)$$

- What is the optimal output for these objects?

$$p_* = \operatorname{argmin} \sum_{i=1}^{n} l\left(y_i, p\right)$$

- We expect that $p_* = \dfrac{1}{n} \sum_{i=1}^{n} \left[y_i = +1\right]$

Predicting probabilities: Log-Loss

- Consider objects $x_1, \ldots, x_n$, where the $b(x)$ outputs the same probability around p:

$$\sum_{i=1}^{n} l\left(y_i, b\left(x_i\right)\right) = \sum_{i=1}^{n} l\left(y_i, p\right)$$

- Which output logistic regression would have on these objects?

$$p_* = \operatorname{argmin} \sum_{i} \left\{-\left[y_i = +1\right]\log p - \left[y_i = -1\right]\log(1-p)\right\}$$

Log-loss

$$p_* = \operatorname{argmin} \sum_i \{-[y_i = +1]\log p - [y_i = -1]\log(1-p)\}$$

Calculate the derivative and find optimal probability:

$$\sum_i \left\{ -\frac{[y_i = +1]}{p} + \frac{[y_i = -1]}{1-p} \right\} = -\frac{n_+}{p} + \frac{n_-}{1-p} = 0$$

$$p_* = \frac{n_+}{n_+ + n_-} = \frac{1}{n} \sum_{i=1}^{n} [y_i = +1]$$

Predicting probabilities: Log-Loss

We assume that the model gives correct probabilities if for any set $y_1, \ldots, \ y_n \in \mathbb{Y}$

$$\operatorname{argmin} \sum_{i=1}^{n} l\left(y_i, \ p\right) = \frac{1}{n} \sum_{i=1}^{n} \left[y_i = +1\right]$$

- This is a condition on a loss function (we can check it for Log-Loss, MSE, MAE, etc.)
- It holds for Log-Loss
- Logistic Regression gives us correct probabilities

Summary

- We can formulate the condition that the model estimates the probabilities correctly
- Choose loss functions which satisfy this condition
- Log-loss is one example of such loss
- Another example is MSE, but MSE works poorly with classification tasks